An AI Investigation Of Citation's Epistemological Role

Kevin D. Ashley
University of Pittsburgh School of Law
3900 Forbes Avenue
Pittsburgh, PA 15260
412.648.1495
ashley@pitt.edu

Bruce M. McLaren
OpenWebs Corporation
2403 Sidney Street
Pittsburgh, PA, USA 15203
412.395.3900
bmclaren@openwebs.com

ABSTRACT

This paper describes how we used an AI model for retrieving ethics cases to investigate empirically the epistemological contributions of a decision-makers' citing cases and code provisions in justifying decisions. In practical ethics, like law, it is impossible to define abstract principles intensionally so that they may be applied deductively. After investigating hundreds of professional ethics case opinions, we hypothesized that the decision-makers' explanations *extensionally* defined principles over time, in effect, *operationalizing* them. We constructed SIROCCO, a system for retrieving principles and past ethics cases. We used this computational model to conduct an ablation experiment concerning a core set of operationalization techniques. This paper presents empirical evidence that the operationalization information supports predictions of the relevant principles and past cases more accurately than competing approaches that do not use such information.

1. INTRODUCTION

Citation is an important and ubiquitous feature of legal opinions. In both civil and common law jurisdictions, judges cite the statutes, treaties, or constitutional provisions that provide the legal rules for deciding a new case. In common law jurisdictions, judges also cite precedents that expound upon or apply the legal rules for deciding the case. Citations serve both normative and epistemological functions. Beside providing authoritative legal rules for decision, citations to precedents supplement the meanings of the open-textured terms in those legal rules. These meanings guide judges in applying the rules to novel fact situations.

We were interested in whether citation's epistemological role could be investigated empirically using an AI model. We developed the model after investigating a corpus of ethics opinions decided over a forty-year period by a professional engineering society's Board of Ethical Review (Board or BER). The National Society of Professional Engineers (NSPE) has published more than 400 ethics case opinions, issued from 1958-1998, on the Internet (http://www.niee.org/cases/), and continues to publish case opinions.

The Board members are not judges; the BER comprises five to seven professional engineers and is reconstituted annually. Nor are the cases legal cases. Instead, the BER opinions are advisory decisions based on submitted factual scenarios. Nevertheless, the Board's decisions share a number of important features with legal cases. First, in making their decisions, the Board applies and cites a set of normative rules. They are not legal rules but ethical ones drawn from the Society's extensive, published Code of Ethics, comprising 75 principles. Second, the BER opinions are written and published. Following a standard format, the Board recites the facts of the case and the question presented, and then explicitly cites applicable code provisions. In a discussion section, the Board considers how the cited code provisions apply to the facts of the case and justify the Board's stated conclusion. Third, the Board regularly cites its own past cases in deciding new ones. Presumably, the cases have no formal *stare decisis* effect, but the Board often states that an analogous past case provides strong guidance in deciding a current one or takes pains to distinguish the current case from a former one.

The ethics code provisions are decidedly open-textured. They embody mid-level, but still quite abstract, general principles such as: “Engineers shall … recognize that their primary obligation is to protect the safety, health, and welfare of the public.”[1] Or “Engineers shall act in professional matters for each employer or client as faithful agents or trustees.”[2] Or “Engineers shall not disclose confidential information concerning the business affairs or technical processes of any present or former client or employer without his consent.”[3] The Code does not define terms like “safety”, “health”, “welfare”, “public”, “faithful” or “confidential”. In addition, for any given fact situation, multiple abstract principles may appear to apply equally well but recommend conflicting advice.

Indeed, the high level of abstraction of the ethics code and its comprehensiveness make it hard for many engineers to use. We developed our computational model, SIROCCO (System for Intelligent Retrieval of Operationalized Cases and COdes) to

[1] NSPE Code II.1.A. The code provision continues: “If their professional judgment is overruled under circumstances where the safety, health, property or welfare of the public are endangered, they shall notify their employer or client and such other authority as may be appropriate.”

[2] NSPE Code I.4

[3] NSPE Code III.4

explore how to make it easier for engineers and students to retrieve code provisions and cases that are relevant to new problem scenarios.

In an extensive and systematic investigation of the Board's opinions (McLaren, 1999), we observed that as the Board decided cases, it made conceptual links between code provisions and selected case facts. More particularly, we hypothesized that, as the Board explained its decisions by citing, applying and resolving code provisions in concrete fact situations, and by citing past cases as justifications, its explanations *extensionally* defined those abstract principles, in effect, *operationalizing* them (Mostow 1983). This effort focused our attention on the extent to which SIROCCO could use these conceptual links. We hypothesized that our computational model could take advantage of these operationalizations to make better predictions of relevant code provisions and past cases in the analysis of new problems.

This paper describes an ablation study we conducted with SIROCCO to investigate that hypothesis. Section 2 presents a sample ethics case and illustrates our techniques for representing case facts and aspects of the Board's analysis. Section 3 elaborates the nine operationalization techniques we observed and shows how the case representation records the operationalizations. In Section 4, we describe SIROCCO's retrieval algorithm and illustrate how it makes use of operationalizations in an extended example. The ablation experiment is described in Section 5; results are presented in Section 6 and discussed in Section 7. In Section 8, we compare SIROCCO with other approaches, notably GREBE, and discuss some challenges in fielding SIROCCO as a practical tool.

2. REPRESENTING ETHICS CASES

One of the significant challenges of this work was to develop a way to represent ethics cases, one which could represent the BER's conceptual links among critical facts, code provisions, and past cases and one which engineers and students could use to represent problems.

The facts of a sample BER case, 76-4-1, are shown in Figure 1. A firm has hired an engineer to study the effect of its planned discharges on water quality, but does not like the results of the study, terminates the engineer's consultation, and directs him not to disclose the study to anyone. When the engineer discovers that the firm has presented contrary evidence at a regulatory hearing, he does not disclose the results of his study.

As described more fully in (McLaren 1999), we developed an ontology, comprising: an Ethics Transcription Language (ETL) for representing the facts of such engineering scenarios and extensions (EETL) for representing aspects of the Board's arguments for and against its conclusions.

In designing ETL, we chose to represent ethics cases as narratives of chronologically ordered events, expressed in a limited language (See, e.g., Leake 1991). ETL represents the actions and events of an ethics scenario like that of case 76-4-1 as an ordered list of individual sentences (i.e., *Facts*) called a *Fact Chronology*. The Fact Chronology for case 76-4-1 is shown in Figure 2.

As enforced by an ETL grammar (McLaren 1999), each Fact comprises: (1) a fact number, (2) Fact-Phrase, and (3) a Time Qualifier. Each Fact-Phrase includes an instance of the kinds of actors and objects, which appear in these scenarios; here it is Engineer Doe. It also includes a *Fact Primitive,* a verb phrase that indicates a specific action or event involving the actors, objects, or similarly constituted Fact-Phrases. In Fact 11, for instance, Doe does not inform the regulatory authority that the discharge would cause a safety hazard.

The Time Qualifier is a temporal relation that specifies how a Fact relates to all of the other Facts. Here, for instance, the failure to inform in Fact 11 occurs after Doe's discovery of the hearing in Fact 10, which itself occurs after Fact 9 concludes, etc. Time Qualifiers are implemented as disjunctive compositions of Allen's temporal constraints (1983). SIROCCO uses TIMELOGIC (Koomen 1989), a time propagation system, to infer the temporal relationships among facts not directly specified.

At least one Fact in the Fact Chronology is designated as a *Questioned Fact*. Here Fact 11 is the action or event that corresponds to the ethical question raised in the scenario. SIROCCO generates a separate case for each Questioned Fact in the context of the same Fact Chronology.

To round out the description of our ontology, two abstraction hierarchies (not shown) help to define inexact matching of cases and codes, respectively: an *Action/Event Hierarchy* which clusters and generalizes similar Fact Primitives, and a *Code Hierarchy* which clusters codes dealing with similar issues.

Facts of case 76-4-1:

The XYZ Corporation is advised by a State Pollution Control Authority that it has 60 days to apply for a permit to discharge manufacturing wastes into a receiving body of water. XYZ is also advised of the minimum standard that must be met. In an effort to convince the authority that the receiving body of water will still meet established environmental standards after receiving the manufacturing wastes, the corporation employs Engineer Doe to perform consulting engineering services and submit a detailed report. After completion of his studies but before completion of any written report, Doe concludes that the discharge from the plant will lower the quality of the receiving body of water below established standards. He further concludes that corrective action will be very costly. Doe verbally advises the XYZ Corporation of his findings. Subsequently, the corporation terminates Doe's contract with full payment for services performed, and instructs him not to render a written report to the corporation. Thereafter, the control authority calls a public hearing, and the XYZ Corporation presents data at the hearing to support its view that the present discharge meets minimum standards. Doe learns of the hearing and XYZ's presentation but does not report his earlier contradictory findings to the authority.

Question:

Was it ethical for Doe not to report his findings to the authority upon learning of the hearing?

Figure 1: Facts of Case 76-4-1

1. XYZCo. **<hires the services of>** Eng'r Doe **<for>** (Eng'r Doe **<reviews and analyzes>** Discharge).	Pre-existing fact
2. Eng'r Doe **<reviews and analyzes>** Discharge.	After 1 starts
3. Eng'r Doe **<discovers that>** (Discharge **<fails standards and may be hazardous to safety>**).	After 2 starts
4. Eng'r Doe **<informs>** XYZCo. **<that>** (Discharge **<fails standards and may be hazardous to safety>**).	Immediately after 3 concludes
5. XYZCo. **<terminates the services of>** Eng'r Doe.	After 4 concludes; Ends 1
6. XYZCo. **<instructs>** Eng'r Doe **<to>** (Eng'r Doe **<does not write paper/article>** (Discharge **<fails standards and may be hazardous to safety>**)).	Occurs concurrently with 5
7. XYZCo. **<pays>** Eng'r Doe <for> (Eng'r Doe **<reviews and analyzes>** Discharge).	Occurs concurrently with 5
8. Control Authority **<calls a hearing regarding>** Discharge.	After 5 concludes
9. XYZCo. **<claims that>** (Discharge **<does not fail standards and is not hazardous to safety>**).	After 8 starts
10. Eng'r Doe **<discovers that>** (Control Authority **<calls a hearing regarding>** Discharge) & (XYZCo. **<claims that>** (Discharge **<does not fail standards and is not hazardous to safety>**)).	After 9 concludes
11. Eng'r Doe **<does not inform>** Control Authority **<that>** (Discharge **<fails standards and may be hazardous to safety>**). *[Questioned fact]*	After 10 starts

Figure 2: ETL Fact Chronology for Case 76-4-1

To input a target case, a human case enterer must transcribe the facts of the case into ETL using a web-based case acquisition tool that we developed (www.pitt.edu/~bmclaren/ethics). The web site contains instructions on how to transcribe ethics cases into ETL (and EETL), a reference shelf of useful materials, including the complete vocabulary employed, and an example set of 47 transcribed Fact Chronologies.

For source cases to be stored in the case base, case enterers also had to encode the Board's analysis into EETL. Figure 3 shows excerpts of the Board's analysis of case 76-4-1. The three tables include code provisions the Board cited in support of its conclusion (top), code provisions that conflict with its conclusion (middle), and past cases it deemed relevant (bottom).

Questioned Fact(s):	Fact 11
Questioned Actor or Actors:	Engineer Doe
The Board's Conclusion:	Unethical

A. The Board cited [this] evidence in support of their conclusion:

Code #/ Status/ Grouped With/ Overrides	Why Relevant?	Why Viol., Not Viol., Changed, or Not Appl.?
II.1.a/ Violated/ I.1, II.1.a/ III.1, I.4, III.4	^ Engineer's judgment is overruled in a particular professional circumstance. [6] Overruling the Engineer's judgment may lead to the endangerment of the safety, health, property or welfare of the public [3, 9] ^	^ Engineer does not notify the proper authority [11] ^
...	...	...

B. The Board cited this evidence that conflicts with their conclusion:

Code #/ Status/ Grouped With/ Overrides	Why Relevant?	Why Viol., Not Viol., Changed, or Not Appl.?
III.1, I.4/ Not violated/ III.1.b/ None	^ Engineer has a client [1]^	^ Engineer acts as a faithful agent or trustee … [12] ^
...	...	...
III.4/ Not violated/ None/ None	^ Engineer obtains confidential information concerning the business affairs … of a former client [2, 3] ^	^ Engineer does not disclose the confidential information [11] ^

C. The Board cited the following background information that neither directly supports nor directly conflicts with their conclusion:

Case/ Citation Type/ Grouped with/ Q#	Why Relevant?	Why Distinguished or Analogous?
67-10/ Relevant, But Not Controlling/ None/ 1	^%Engineer is involved in a professional situation in which the public welfare is at stake %[3, 9] ^	^NA ^

Figure 3: EETL Tables Representing BER Analysis of Case 76-4-1 (excerpts)

SIROCCO's case base includes 184 cases from the NSPE database, covering 135 different Fact Chronologies and culled from the 475 cases decided by the BER between 1958 and 1992. Twelve independent case enterers transcribed these foundational cases into EETL using the case-acquisition web site. We used these cases to design, implement, and refine the program. They cover important ethics topics (i.e., the Selected Topics) including public safety, confidential information, duty to employer, credit for engineering work, proprietary interests, and honesty in reports and public statements. They also provide some coverage of other topics (the Non-Selected Topics.) 135 of the foundational cases cite at least one code related to at least one of the Selected Topics. The other 49 do not cite any of the Selected Topics codes. The cases are spread across the topics and tend to cite different sets of codes.

The case enterers reported that it took an average of 2 to 3 hours to transcribe a source case (i.e., representing both a Fact Chronology and the Board's analysis). Most of that time was spent representing the Board's analysis.

3. THE BER'S OPERATIONALIZATIONS

In examining the BER's opinions, we identified nine ways in which the Board conceptually linked critical facts, ethics code provisions, and past cases, effectively operationalizing the code provisions and past cases (See Figure 4). Using instantiations, for instance, (i.e., techniques 1 and 6) codes and cases are defined extensionally so that SIROCCO can reuse them in analyzing new cases. Instantiating a cited code or past case means explicitly relating it to a questioned fact, certain critical facts, and the temporal sequence of those facts in the citing case.

Information that the case enterers record in the EETL tables for a case, such as that in Figure 3 for case 76-4-1, provide the basis for instantiations and other operationalizations. The numbers in brackets [ ] in the "Why Relevant" and "Why Violated..." columns refer to those facts in the representation of case 76-4-1 (Figure 2) that are critical to the code's (or past case's) application and explain why the code was [not] violated (or why the case was distinguished). In this way, Code II.1.a, for instance, is connected extensionally to a real case's relevant facts and chronology in a way that SIROCCO can reuse. Specifically, as in the top table, code II.1.a is associated with Facts 3, 6, 9 and 11 in case 76-4-1 (and their accompanying temporal relations) and noted as having been violated in that case. Code III.4 is associated in the middle table with Facts 2, 3, and 11 and is noted as having not been violated.

1. Instantiating principles by linking them to clusters of questioned and critical facts.
2. Hypothesizing facts that affect how principles apply.
3. Revising a principle over time in light of cases.
4. Resolving conflicting principles in specific cases.
5. Grouping principles in specific cases.
6. Instantiating cases as precedents by linking them to clusters of questioned and critical facts, and by analogizing or distinguishing them.
7. Applying, defining or elaborating issues and principles from past cases.
8. Grouping cases.
9. Reusing specific applications of any of the above techniques from previous analyses.

Figure 4: Operationalization Techniques

Operationalization techniques 5 and 8 involve the BER's grouping together in its case analysis selected code provisions or past cases. This information is in the first column of the EETL tables in Figure 3 (i.e., "Grouped With").

Operationalization technique 9 (Reusing operationalizations) involves the fact that the Board may reuse any of the operationalizations it used in a cited past case. For instance, the case and code instantiations in a cited case can be reused.

We regard the above-described operationalizations (i.e., techniques 1, 5, 6, 8, and 9 in Figure 4) as a core subset because they contribute most directly to retrieving relevant cases and codes.

The remaining operationalization techniques (i.e., 2, 3, 4 , and 7) focus on elaborating explanations of the significance of retrieved information. For instance, technique 4 ("Resolving conflicting principles...") enables the model to point out that conflicting principles could be resolved in the same way as in a past case. It uses the information in the EETL tables that certain codes apply, some of which support and some of which conflict with the Board's conclusion for the case. For instance, it follows from the instantiation information noted above and recorded in Figure 3 that Code II.1.a overrode Code III.4 in case 76-4-1.

Technique 2 ("Hypothesizing facts") enables the model to point out factual changes that could affect how a code provision would apply. The case enterers record the Board's use of a hypothetical in discussing a code citation in the "Why Relevant?" or "Why Violated…?" columns of the EETL tables shown in Figure 3. They simply quote the Board's hypothesized facts. The quotation may be spliced into the program's explanation.

Technique 3 ("Revising a principle") reflects that the Board sometimes notes that it recommends changing the wording of a code provision in light of a case's facts or changing social mores. This information is recorded in the "Why ... changed" column of the EETL tables, Figure 3. Finally, the Board sometimes uses a possibly relevant but not highly analogous past case to define concepts or to introduce or elaborate upon a general issue. Case enterers quote this information in the EETL tables, Figure 3 (bottom); these quotes may be spliced into the program's explanations.

4. HOW SIROCCO WORKS

As shown in Figure 5, SIROCCO retrieves cases in two-stages (see, e.g., Branting 1991, 2000; Forbus et al. 1994; Thagard and Holyoak 1990). In order to make the search more efficient and accurate, each stage employs code and case instantiations (i.e., operationalization techniques 1 and 6, Figure 4) to focus SIROCCO on the most critical facts to match. In Stage 1, a target case's Fact Primitives are matched to those of all possible source cases. Extra weight is accorded, however, to matching source case Fact Primitives associated with instantiated codes and cases. In Stage 2, a computationally more-expensive A* search is applied to map selected case structures between the target case and the best N candidate source cases from Stage 1. Stage 2's structural mapping routine, however, attempts to match only the part of a source case's Fact Chronology associated with an instantiation.

In Stage 1, content vectors (Forbus et al. 1994) are employed to make a quick but rough comparison of the target case and all possible source cases. Figure 6 shows the content vector for case 76-4-1. Each vector summarizes the Fact Chronology of a single case. It specifies the Fact Primitives, and their corresponding abstractions in the Action/Event Hierarchy, and a count of how many times each appears. Figure 6 shows two content vectors for case 76-4-1. The top (Fact-Primitive) is the most specific; the bottom (Fact-Group) is one level higher up in the Action/Event Hierarchy.

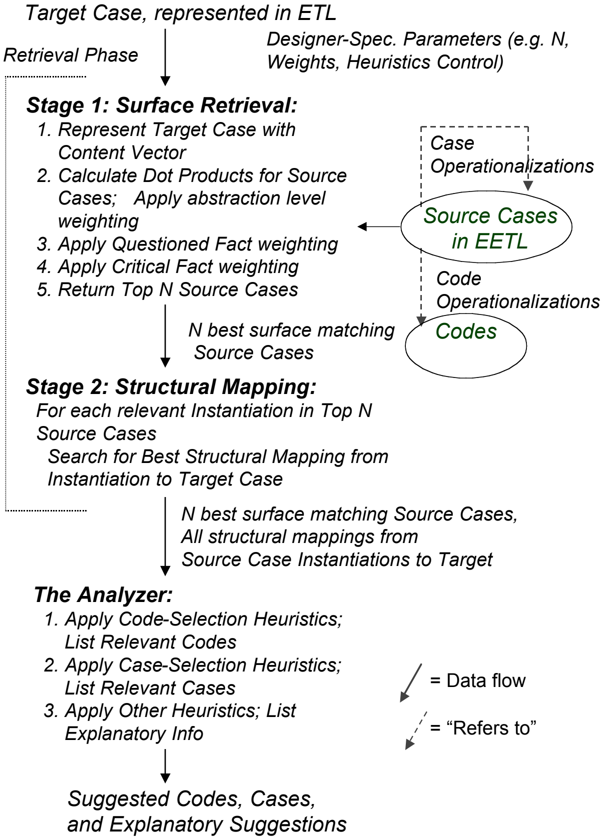
Target Case, represented in ETL
Retrieval Phase
Designer-Spec. Parameters (e.g. N, Weights, Heuristics Control)
Stage 1: Surface Retrieval:
1. Represent Target Case with Content Vector
2. Calculate Dot Products for Source Cases; Apply abstraction level weighting
3. Apply Questioned Fact weighting
4. Apply Critical Fact weighting
5. Return Top N Source Cases
Case Operationalizations
Source Cases in EETL
Code Operationalizations
Codes
N best surface matching Source Cases
Stage 2: Structural Mapping:
For each relevant Instantiation in Top N Source Cases
Search for Best Structural Mapping from Instantiation to Target Case
N best surface matching Source Cases, All structural mappings from Source Case Instantiations to Target
The Analyzer:
1. Apply Code-Selection Heuristics; List Relevant Codes
2. Apply Case-Selection Heuristics; List Relevant Cases
3. Apply Other Heuristics; List Explanatory Info
= Data flow
= "Refers to"
Suggested Codes, Cases, and Explanatory Suggestions

Figure 5: SIROCCO's Algorithm

Case 76-4-1

Fact-Primitive-CV:
(Hires-the-Services-Of 1)
(Reviews-and-Analyzes 1)
(Discovers-That 3)
(Informs-That 2) ***
(Terminates-the-Services-of 1)
(Instructs-to 1)
(Pays-For 1)
(Calls-a-Hearing-Regarding 1)
(Claims-That 1)

Fact-Group-CV:
(Work-as-an-Employed-or-Contract-Professional-Engineer 1)
(Perform-Engineering-Design-or-Analysis-Work 1)
(Perform-Engineering-Analysis-Review-or-Testing-Work 1)
(Know-or-Believe-Something 3)
(Disclose-Information 3) ***
(Terminate-Services-by-Client 1)
(Order-Subordinate-to-Perform-Task 1)
(Give/Receive-Remuneration 1)
(Purchase-or-Pay-for-Something 1)
(Initiate-Legal-or-Arbitration-Proceedings 1)

Figure 6: Content Vectors for Case 76-4-1

In Stage 1, the weighted dot products of content vectors are computed for the target case and all cases in the case base. Candidate cases are ranked by descending dot product scores. Different weights are assigned to matches at different levels of the Action/Event Hierarchy (e.g., Fact-Primitive matches may be weighted twice as highly as more abstract Fact-Group matches.) Higher weights are also assigned to matches of a source case's questioned facts and critical facts. The latter are those facts associated with the source case's code and past case instantiations. (See *** in Figure 6.)

In Stage 2, a heuristic A* search attempts to perform a structural mapping between the target case and each of Stage 1's N top-ranked candidates. SIROCCO's use of A* search for case structure mapping supplements that of (Branting 1991, 2000) by taking temporal relations into account, supporting abstract matches, and accommodating a wider range of factual scenarios. SIROCCO's search focuses on matching code and case instantiations associated with the source case (i.e., operationalization techniques 1 and 6). The program attempts to map each of the Facts of the source instantiation to a corresponding Fact in the target case. At the same time, it must maintain a one-to-one and consistent mapping between the Actors and Objects of the source and target.

The source's questioned Actor is mapped onto the target's in the initial node of the search space. New nodes are generated by selecting an unmapped Fact from the source instantiation and mapping it to each of the target's unmapped Facts. If the corresponding Fact Primitives match either exactly or abstractly, the new node is created. Thus, each subsequent node corresponds to: (1) a tentative mapping of a Fact from the source instantiation to a target case Fact, (2) all of the successful Fact mappings from ancestor nodes, (3) the set of Actors and Objects entailed by the Fact mappings, and (4) consistent temporal relations between the mapped Facts of the source and target. Temporal relations are consistent if the Allen relations of every pair of source Facts intersect with the Allen relations of the corresponding pair of target Facts. An "empty" node is generated at each ply to represent the possibility of no match between the current source Fact and any target Facts. In this way, a search path may be extended that contains a current failed match, but subsequent successful matches.

In order to illustrate how Stage 2 works, and in particular, how SIROCCO employs instantiations in past source cases to operationalize code provisions for later retrievals, let us consider how case 76-4-1 is employed as a source case in analyzing a target case 90-5-1. The facts of and SIROCCO's output for case 90-5-1 are shown in Figure 7. (See McLaren & Ashley 2000 for the ETL representation of case 90-5-1 as a target case. The example here focuses on the representation, matching and use of a *source* case with respect to that target.) An engineer hired as an expert witness by the building owner discovers serious safety hazards, but conceals them at the request of the owner's attorney. SIROCCO identifies case 76-4-1 and ethics code provision III.4 as relevant.

Figure 8 shows the search tree for mapping the code instantiation of ethics code III.4 ("Do not Disclose Confidential Information Without Consent") in case 76-4-1 to the facts of the target case, 90-5-1. As shown in Figure 3, the EETL tables for case 76-4-1, the instantiation of code III.4 has three components, Facts 2, 3, and 11. These Facts represent the engineer's actions in case 76-4-1 relating to his protection of his client's confidentiality. Recall that the engineer performed a review (i.e., step 2, "reviews-and-analyzes"), discovered a potential safety hazard (i.e., step 3, "discovers-that"), but did *not* report the hazard to the authorities (i.e., step 11, "does-not-inform-that"). Correspondingly, the solution depth of the search tree is fixed to 3, the number of components (i.e., Facts) in the instantiation to be mapped. Each level below the root node corresponds to a different component of the instantiation, and in particular, to the corresponding Fact Primitive.

***** SIROCCO is analyzing Case 90-5-1**

Facts: Tenants of an apartment building sue the owner to force him to repair many defects in the building that affect the quality of use. The owner's attorney hires Engineer A to inspect the building and give expert testimony in support of the owner. Engineer A discovers serious structural defects in the building, which he believes constitute an immediate threat to the safety of the tenants. The tenants' suit has not mentioned these safety-related defects. Upon reporting the findings to the attorney, Engineer A is told he must maintain this information as confidential as it is part of a lawsuit. Engineer A complies with the request of the attorney.

Question: Was it ethical for Engineer A to conceal his knowledge of the safety-related defects in view of the fact that it was an attorney who told him he was legally bound to maintain confidentiality?

***** SIROCCO['s]...suggestions for evaluating 90-5-1:**

*** ***Possibly Relevant Codes:***

I-4: Act as a Faithful Agent or Trustee

III-4: Do not Disclose Confidential Info. Without Consent

I-1: Safety, Health, and Welfare of Public is Paramount

II-1-A: Primary Obligation is to Protect Public (Notify Authority if Judgment is Overruled). …

II-1-C: Do not Reveal Confidential Info. Without Consent

III-2-B: Do not Complete or Sign Documents that are not Safe for Public …

*** ***Possibly Relevant Cases:***

76-4-1: Public Welfare - Knowledge of Information Damaging to Client's Interest

89-7-1: Duty To Report Safety Violations

84-5-1: Engineer's Recommendation For Full-Time, On-Site Project Representative

*** ***Additional Suggestions:***

o The codes II-1-A … and I-1 … may override codes III-4 I-4 … and III-1 … in this case. See case 76-4-1 for an

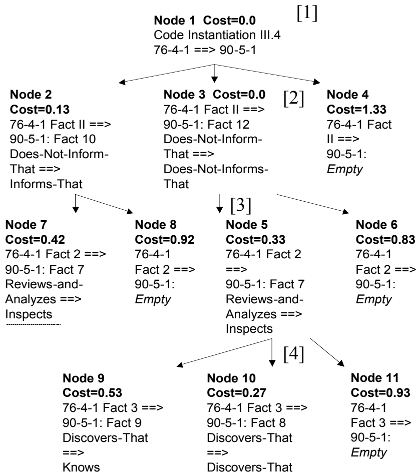

Figure 8: Search Tree for Code III.4 Instantiation

The search opens 11 nodes in finding the optimal path represented by nodes 1, 2, 3, and 4. Each node is evaluated in terms of the A* cost function, $f(n) = g(n) + h'(n)$. The mismatch cost g(n) is the degree of mismatch at each node up to and including n, divided by the current depth. The default mismatch costs (as match levels increase in abstraction in the Action/Event Hierarchy) range from 0.0 for an exact Fact Primitive match to 1.0 for a completely failed match.

The h'(n) function estimates the most optimistic completion of the mapping from node n. It is the mismatch cost that would be attained by achieving an exact match (i.e., adding 0) at each node from n until the goal node is reached. It is calculated by dividing the summed degree of mismatch up to node n by the solution depth. SIROCCO always returns the minimum f(n) found at the fixed solution depth.

The goal node (here Node 10, step [4]) is reached when the current depth equals the pre-defined solution depth (here 3) and either the current node has the lowest mismatch score of all open nodes, as defined by the A* cost function, or the list of nodes is empty. Upon reaching Node 10, the three components of the instantiation of code III.4 in case 76-4-1 have been mapped successfully to three Facts in the problem case 90-5-1.

After Stage 2, SIROCCO's Analyzer phase outputs the explanation shown in Figure 7 for case 90-5-1 based on the results of Stages 1 and 2. Figure 9 shows some additional details in SIROCCO's output for that case. The program lists possibly relevant codes and cases and shows the data and heuristics that underlie its reasoning. Selection heuristics corresponding to Operationalization Techniques 1, 5, 6, 8 and 9 generate these lists and reasons. The heuristics favor codes that, for example, (1) occur more frequently in the top-ranked cases of Stage 1 (i.e., "best surface matching cases"), (2) match a high percentage of critical facts in cases citing the code, or (3) are grouped with other codes cited in those cases. Similar heuristics are used to select cases. SIROCCO also summarizes Stage 2's structural mapping between cases 76-4-1 and 90-5-1.

```
*** Possibly Relevant Codes
III-4: Do not Disclose Confidential Information Without
Consent
                    Heuristics Explanation:
          ----------------------------------------------
o Cited by 4 of the 6 best surface matching cases.
o 100.0% match to 4 critical facts in case 89-7-1.
o  86.7% match to 3 critical facts in case 76-4-1.
o 100.0% match and ques. fact match (Source and
          Target: FACT-PRIMITIVE) in case 89-7-1.
o 86.7% match and ques. fact match (Source and
          Target: FACT-PRIMITIVE) in case 76-4-1.
o  > 50.0% match in multiple cases: 89-7-1, 76-4-1.
o  Grouped with code I-4 in case 89-7-1; Good match
          to critical facts.
o  Grouped with code I-4 in case 76-4-1; Good match
          to ques. facts.
…
*** Possibly Relevant  Cases:
76-4-1: Public Welfare - Knowledge of Information
Damaging to Client's Interest
                    Heuristics Explanation:
          ----------------------------------------------
o 100.0% match to 3 critical facts in code I-4.
o 100.0% match and ques. fact match (Source and
          Target: FACT-PRIMITIVE) in code I-4.
o 86.7% match to 3 critical facts in code III-4.
o 86.7% match and ques. fact match (Source and
          Target: FACT-PRIMITIVE) in code III-4.
…
                 Structural mapping Explanation:
          ----------------------------------------------
To case 76-4-1   (Corresponding steps of case 90-5-1
indented; * = Questioned Fact)
ENGINEER-DOE DISCOVERS-THAT
((FAILS-STANDARDS-…))
          >>> ENGINEER-A DISCOVERS-THAT
          ((FAILS-STANDARDS-…))
ENGINEER-DOE REVIEWS-AND-ANALYZES DISCHARGE
          >>> ENGINEER-A INSPECTS
           APARTMENT-BUILDING
*ENGINEER-DOE DOES-NOT-INFORM-THAT
CONTROL-AUTHORITY …
          *>>> ENGINEER-A DOES-NOT-INFORM-
           THAT ANYONE-ELSE …
```

Figure 9: Details of SIROCCO Output for Case 90-5-1

As a final step, the Analyzer phase applies heuristics to make additional suggestions. These heuristics implement operationalization techniques 2, 3, 4, and 7. For each of the possibly relevant codes and source cases, SIROCCO attempts to find operationalized information in those cases which may bear upon the present circumstances. For instance, as shown at the bottom of Figure 7, SIRCOCCO suggests that the codes dealing with public safety (II.1.a. and I.1.) may override the code dealing with confidentiality (III.4.) in the circumstances of target case 90-5-1. It makes this suggestion because case 76-4-1 is an example of such a conflict, and all of the same codes are suggested by SIROCCO in the present case. To confirm this, see the first column and row of the top EETL table for case 76-4-1 in Figure 3. It shows that II.1.a overrides III.4. The Board actually did employ such a "resolving conflicting principles" operationalization in their analysis of case 90-5-1, where it resolved competing obligations to public safety and to preserving a client's confidentiality.

5. THE EXPERIMENT

In order to assess the contribution of the Board's operationalized code provisions and cases, we conducted some experiments, including an ablation experiment (Rissland *et al.* 1996). When tested, SIROCCO's casebase included 184 foundational cases. As test cases, we used a set of 58 trial cases that were decided later than any of the foundational cases (and thus could cite relevant foundational cases). We employed two independent case enterers to transcribe all of the trial cases into extended ETL. Their transcriptions were submitted unaltered to SIROCCO for processing. The 58 trial cases were chosen from a set of 77 cases decided by the BER after 1993: 44 trial cases were chosen randomly from 52 Selected Topics cases and 14 trial cases were chosen randomly from 25 Non-Selected Topics cases.

We compared SIROCCO's performance with that of five other methods:

> **RANDOM**: Codes/cases randomly selected for each case.
>
> **INFORMED-RANDOM**: Like RANDOM but accounts for frequency distribution of code/case citations in NSPE corpus.
>
> **NON-OP SIROCCO**: Ablated version of SIROCCO, with almost no functionality related to operationalizations.
>
> **MG (Managing Gigabytes):** Full-text retrieval method converts ethics case into term vector and compares to codes/cases vectors.
>
> **EXTENDED-MG**: Like MG, but codes selected according to frequency of citation in the top X selected cases.

Most relevantly for the present paper, the comparison with NON-OP SIROCCO focuses on the contribution of the Board's operationalizing code provisions and cases. In particular, it focuses on the core subset of the operationalizations in Figure 4, those which contribute directly to retrieving relevant cases and codes (i.e., techniques 1, 5, 6, 8, and 9.) In NON-OP SIROCCO, we turned these core-set operationalization techniques off. NON-OP SIROCCO did, however, prefer codes that appeared most frequently in the list of the *N* top-rated cases; this can be considered a weak kind of operationalization technique, but it is the only such information NON-OP SIROCCO used.

Each method, including SIROCCO, processed each of the trial cases one-by-one, and its retrieval results were compared to the BER's code and case citations for the same case. The *F-measure,* an IR metric that combines precision *P* and recall *R*, was used to calculate the overlap between a method's solution and the Board's slution: $F = (\beta^2 + 1)PR / (\beta^2 P + R)$ (Lewis *et al.* 1996). The value of β was set at 1.0 to assign equal weights to precision and recall. For each trial case, we calculated two F-Measure values. One indicated the extent of exact matches of codes and cases between the two solutions. The other indicated the extent of inexact matches. Using the Code Hierarchy, an inexact match was scored according to a citation overlap metric that measures the inverse of the length of the citation path between two cases. For instance, if a case directly cites another, the overlap is 1/1. If two cases share a citation to a third case, the overlap is 1/2 (McLaren & Ashley 1999).

6. RESULTS

The results are shown in Figure 10.

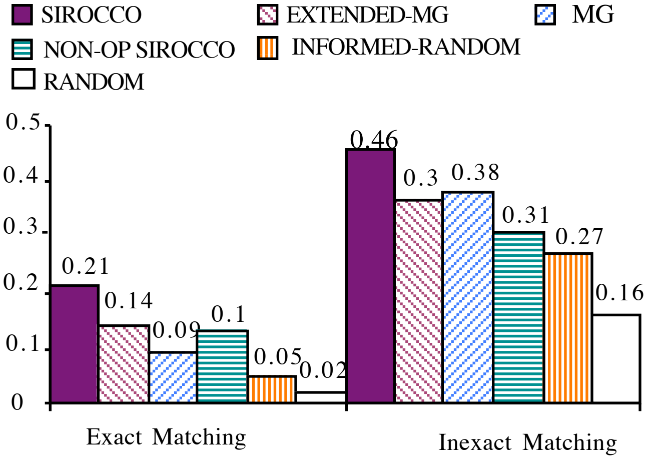

Figure 10: Mean F-Measures: All Methods Over All Trial Cases

The data generated by benchmarking each method against the BER’s citations using the F-Measure was highly non-Gaussian. Accordingly, we applied a *nonparametric bootstrap procedure* (Davison and Hinkley 1997) to compare SIROCCO with each other method. The probability that SIROCCO was more accurate than the other five methods was greater than 95% for all but EXTENDED-MG on the inexact matching. There the probability was 94.3%, just below the threshold of statistical significance.

In other words, SIROCCO was significantly more accurate than NON-OP SIROCCO in retrieving codes and cases when measured by exact and inexact matching. It was significantly more accurate than EXTENDED-MG for *exact* matching. It was also more accurate in retrieving with *inexact* matching, but the difference was not statistically significant. In a supplemental experiment comparing SIROCCO and EXTENDED-MG, we employed two ethics graduate students to evaluate the extra code and case citations for the trial cases made by SIROCCO and EXTENDED-MG. For each additional code and case suggested by the two methods, the evaluators were asked to indicate whether the extra suggestion was reasonable or not. In this supplemental experiment, we found that SIROCCO was significantly more accurate than EXTENDED-MG for both the exact and inexact match criteria.

7. DISCUSSION

The experiments confirmed that SIROCCO's core operationalization techniques allow it to make more accurate predictions of the principles and past cases that are likely to be relevant in the analysis of new cases than it can without that information. The fact that SIROCCO outperformed NON-OP SIROCCO in the ablation experiment shows that the core operationalization techniques do make a significant difference. Both methods employ the same case representation and Stage 1 retrieval method. The critical difference is that NON-OP SIROCCO makes no use of the core set of operationalization techniques.

Notably, NON-OP SIROCCO makes no use of code and case instantiations (i.e., techniques 1 and 6), which intuitively are the primary operationalization techniques by which codes and cases become defined extensionally in a way that SIROCCO can reuse in analyzing new cases.[4]

The effect is substantial. In the exact matching, the operationalization information accounted for 52% of SIROCCO's F-Measure ((.21 - .1) / .21). In the inexact match, it accounted for 33% ((.46 - .31) / .46). The difference in effect size (52% v. 33%) may be due to the difference in exact and inexact matching. The former is a more difficult task since the program must pick the precise code or case cited in the Board's analysis. Inexact matching leaves open more opportunity for matching via "lucky guesses" since a number of codes might match to the same abstract code. Thus, the core operationalization techniques make a bigger contribution in the exact match test.

We interpret these results as evidence that the BER’s explanations of its decisions over time extensionally define applicability and relevance conditions for the code provisions and past cases. The conceptual links the Board draws between critical facts and code provisions and past cases, respectively, provide information valuable for retrieval in new cases.

While this result is intuitively plausible, perhaps even obvious, we do not believe that it has been demonstrated empirically. Here an AI model enables us to do exactly that.

Parenthetically, outperforming MG and EXTENDED-MG shows that SIROCCO is a more powerful retrieval method than the most likely competitor for this task, a full-text retrieval method (not unlike LEXIS or WESTLAW). The fact that SIROCCO outperformed EXTENDED-MG which, in turn, outperformed MG, is also significant. Since EXTENDED-MG makes weak use of operationalization information, its improvement over MG also supports our hypothesis.

8. CONCLUSIONS

This work extends case-based AI&Law programming techniques (Ashley 1990; Branting 1991, 2000; Rissland *et al*. 1996; Aleven 1997) into a related but distinct new domain. Arguments in practical ethics are more free-form in style and structure than legal arguments. Although like legal cases the NSPE BER cases have binary outcomes (i.e., ethical or nonethical, similar to plaintiff wins or loses), generally ethics cases may require “creative middle way” solutions (i.e., solutions that reconcile conflicting principles) (Harris *et al*. 1999, p. 64-72).

While SIROCCO's use of code instantiations and its two-stage retrieval algorithm with structure-mapping were inspired by GREBE's explanation-based exemplars and criterial facts and its use of A* search (Branting 1991, 2000), SIROCCO attempts to improve upon GREBE in a number of ways.

[4] In principle, additional ablation experiments could determine the relative importance of each operationalization technique in the core set; we did not undertake such experiments.

First, SIROCCO represents cases in a more general way as temporally-ordered, narrative descriptions of events. GREBE's case representation focused more on representing the events in relation to the court's explanations. SIROCCO's ETL provides a total of 190 actions and events compared to 70 to 90 in GREBE, and it covers a wider range of scenarios in its domain. SIROCCO's representation includes formally-defined temporal relations among facts, and a well-defined algorithm for matching temporal relations. GREBE's approach to representing temporal ordering is comparatively ad hoc; its structure mapping does not include matching temporal relations.

Second, GREBE critically depends on maintaining consistency of representation across cases without providing any practical means to achieve it. GREBE's structural mappings fail unless cases are consistently represented. SIROCCO's case acquisition web site, with its examples, guidelines, ontological constraints and limited language for representing cases, provides some practical means for achieving consistency. Its generalized matching techniques and inexact matching also reduce the need for perfect consistency. Indeed, *twelve* case enterers represented SIROCCO's 242 cases. None of them had been involved in developing the program.

Third, SIROCCO's extensional model of how abstract principles and past cases accrue meanings through operationalizations, is more general than GREBE's. Other than Code Instantiations, GREBE does not appear to have equivalents to SIROCCO's operationalization techniques.

On the other hand, GREBE's arguments are deeper than SIROCCO's case analyses. GREBE's representation of a court's precedent-setting explanations in terms of causal and evidential relations enables it to analyze target problems in far more detail than SIROCCO is capable. The evidential relations of GREBE's EBEs are more finely grained than SIROCCO's Instantiation links from Facts (i.e., the individual steps of a Fact Chronology) to relevant codes and past cases. Also, GREBE combines rule-based reasoning and case-based reasoning; SIROCCO relies exclusively on cases for retrieving appropriate codes and cases.

Those familiar with our previous work in AI&Law may wonder why factors play no apparent role in SIROCCO's representation. Factors do appear to be useful in modeling practical ethical reasoning. (See, e.g., Strong 1988). They would be helpful, for instance, in constructing a more detailed computational model of the Board's posing of hypothetical fact variations (operationalization technique 2 in Figure 4) and in modeling line-drawing (Harris, et al. 1999). This is a line of research worth pursuing.

This work suggests how to design an intelligent aid for retrieving code provisions and relevant past cases in fields like professional ethics. A professional engineering society's code of ethics is so comprehensive and so open-textured that engineers and students may find it of limited utility in practical decision-making. Deciding and publishing cases applying the codes is a laudable effort to flesh-out the meanings and conditions for applying the abstract principles, but at the cost of even more material to search and read. Conceivably, intelligent access to the right standards and examples given a problem situation may lead to better decision-making. SIROCCO performs significantly better than a full-text retrieval method, providing clear evidence of the value of its case representation.

More significantly, SIROCCO captures the contribution the Board's decisions make in extensionally defining the abstract codes and utilizes it to improve retrieval. It could help engineers and students bridge the gap between cases and abstract principles. SIROCCO's connections are based on the accumulated links the Board says it is making as it cites codes and cases. SIROCCO does not involve the case enterers or program designers in intensionally defining intermediate-level rules for deducing how the abstract principles apply to realistic scenarios. Such an effort would be impossibly difficult and would result, in any event, in a body of non-authoritative rules (Jonsen & Toulmin 1988).

The experiment described in this paper is strong evidence of the epistemological contribution the Board's citations provide and of the appropriateness of SIROCCO's representation for capturing its operationalizations. Parenthetically, we believe that the Board's decisions also flesh out temporal event-ordering conditions under which a code provision may reasonably apply. In a subsequent experiment, we plan to investigate this contribution empirically.

Of course, there is a cost. It takes much more time and effort to represent a problem situation as a Fact Chronology for SIROCCO than to encode input to a full-text retrieval system. Source cases take much longer to transcribe than target cases, but even so, a more convenient mode of case entry is required. Our hope is that ETL, with its use of a *limited* language focusing on the important *verbs* in a domain, may support a textual CBR approach to representing cases automatically. (See, e.g., Brüninghaus and Ashley 1999). That will require considerably more research, however.

In the meantime, we plan to incorporate SIROCCO into a tutoring environment for practical ethics. It would appear to be pedagogically valuable for students to expend some effort in representing problem scenarios as narratives of temporally-ordered events (e.g., it induces students to consider more carefully the facts of a case.) SIROCCO's explanations of its outputs could also be pedagogically useful. Full-text retrieval schemes alone cannot generate such explanations.

9. ACKNOWLEDGMENTS

This work was supported by an NSF-LIS grant No. 9720341.

10. REFERENCES

[1] Aleven, V.A. 1997. *Teaching Case-Based Argumentation Through a Model and Examples*. Ph.D. Diss., U. Pittsburgh.

[2] Allen, J.F. 1983. Maintaining Knowledge about Temporal Intervals. *Comm. ACM* 26(11), 832-843.

[3] Ashley, K.D. 1990. *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge: MIT Press.

[4] Branting, L.K. 1991. Building Explanations from Rules and Structured Cases. *Int'l J.Man-Machine Studies*, 34 (6): 797-837.

[5] Branting, L.K. 2000. *Reasoning with Rules and Precedents*. Kluwer: Dordrecht, The Netherlands.

[6] Brüninghaus, S. and K.D. Ashley. 1999. Bootstrapping Case Base Development with Annotated Case Summaries (Althoff, K.-D., *et al*. ed.) *CBR Research and Development* (ICCBR-99) Lecture Notes in AI No. 1650, 59-73. Springer: Berlin.

[7] Davison, A.C. and D.V. Hinkley. 1997. *Bootstrap Methods and Their Application*. Cambridge U. Press.

[8] Forbus, K.D., D. Gentner, and K. Law. 1994. MAC/FAC: A Model of Similarity-based Retrieval. *Cognitive Science* 19, 141-205.

[9] Harris, C.E., M.S. Pritchard, and M.J. Rabins. 1999. *Engineering Ethics: Concepts and Cases*. 2d ed. Belmont, CA: Wadsworth.

[10] Jonsen, A.R. and S. Toulmin. 1988. *The Abuse of Casuistry: A History of Moral Reasoning*. Berkeley: U. California Press.

[11] Koomen, J.A. 1989. *The TIMELOGIC Temporal Reasoning System*. Tech. Rep. 231, Computer Science Dept., U. Rochester, NY.

[12] Leake, D.B. 1991. An Indexing Vocabulary for Case-Based Explanation. In *Proc. AAAI-91*. 10-15.

[13] Lewis, D.D., R.E. Schapire, *et al*. 1996. Training Algorithms for Linear Text Classifiers. In *Proc. 19th Ann. Int'l ACM-SIGIR Conf. on Res. and Dev. in Information Retrieval*. Zurich.

[14] McLaren, B.M. 1999. *Assessing the Relevance of Cases and Principles Using Operationalization Techniques*. Ph.D. Diss., U. Pittsburgh.

[15] McLaren, B.M. and K.D. Ashley 2000. Assessing Relevance With Extensionally Defined Principles and Cases. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence* (AAAI-00) pp. 316-322. AAAI Press/ The MIT Press: Menlo Park, CA.

[16] McLaren, B.M. and K.D. Ashley 1999. Case Representation, Acquisition, and Retrieval in SIROCCO (Althoff, K.-D., *et al*. ed.) *CBR Research and Development* (ICCBR-99). Lecture Notes in AI No. 1650. 248-262. Springer: Berlin.

[17] Mostow, J. 1983. Machine transformation of advice into a heuristic search procedure. *Machine Learning*, vol. 1.

[18] Rissland, E.L., D.B. Skalak, and M.T. Friedman. 1996. BankXX: Supporting Legal Arguments through Heuristic Retrieval. *Artificial Intelligence and Law* 4:1-71.

[19] Strong, C. 1988. Justification in Ethics. In Baruch A. Brody, ed., *Moral Theory and Moral Judgements in Medical Ethics*. pp. 193-211. Kluwer: Dordrecht, The Netherlands.

[20] Thagard, P., K.J. Holyoak, *et al*. 1990. *Analog Retrieval by Constraint Satisfaction*. Tech. Rep. CSL-Report 41, Princeton U.