

# Assessing Relevance With Extensionally Defined Principles and Cases

Bruce M. McLaren and Kevin D. Ashley

Graduate Program In Intelligent Systems  
University of Pittsburgh Learning Research and Development Center  
3939 O'Hara Street  
Pittsburgh, Pennsylvania 15260  
bmclaren+@pitt.edu, ashley+@pitt.edu

## Abstract

Expert decision-makers often explain decisions by citing general principles. In some domains, however, it is nearly impossible to define 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. To model this phenomenon computationally, we constructed SIROCCO, a system for retrieving principles and past cases. This paper presents empirical evidence that operationalization information can be leveraged to predict relevant principles and past cases more accurately than competing approaches that do not use such information.

## Introduction

General principles are useful for guiding and explaining decisions. For instance, in classic work designed to make MYCIN's rule-tracing explanations more comprehensible, Clancey recommended explaining a heuristic rule's role in terms of explicit principles of good diagnostic strategy. Such principles included general rules like "if there is evidence for two hypotheses that tend to be confused, try to rule out the second." (Clancey 1983, p. 226).

Since principles are rule-like, one might think that they can be applied deductively. In domains like professional ethics, law, and policy-making, however, often there are no ready sources of authoritative intermediate-level rules for deducing how the abstract principles apply to realistic scenarios. Also, multiple abstract principles may appear to apply equally well but recommend conflicting advice.

In the abstract, everyone may agree with general principles like, "Engineers shall ... recognize that their primary obligation is to protect the safety, health, and welfare of the public."<sup>1</sup> Or "Engineers shall act in professional matters for each employer or client as faithful agents or trustees."<sup>2</sup> Agreement is difficult, however, when one considers how to apply and reconcile such principles in

real-world situations (Jonsen and Toulmin, 1988). Since the general principles are open-textured, experts cannot write intermediate-level rules to cover all possible conditions to which the principles apply. Even their consequents are abstract (e.g., what exactly does it mean to be a "faithful agent"?) Also, an ethicist's rules for interpreting general principles may not be authoritative; ethicists have no special imprimatur to legislate right and wrong.

While efforts to define such general principles intensionally are thus bound to fail, we hypothesized that as ethicists explain their decisions by applying and resolving general principles, their explanations *extensionally* define those principles. In effect, they *operationalize* the general principles (Mostow 1983). We believed, moreover, that a computational model could take advantage of these operationalizations, if not to reason about novel ethical problems, then at least to predict which principles and past cases are relevant for analyzing novel problems.

We investigated a body of ethics case opinions issued from 1958-1998 by the National Society of Professional Engineers' Board of Ethical Review (NSPE BER) and now published on the Internet (<http://www.niece.org/cases/>). Reconstituted annually with a new set of five to seven professional engineers, the BER has written extensive explanations of its decisions in more than 400 cases. In justifying its decisions, the Board applied the Society's published Code of Ethics comprising 75 general principles including the two mentioned above.

We focused particularly on how the BER cited Ethics Code provisions and past cases in its justifications, and identified nine ways in which the BER operationalized the code provisions and past cases. SIROCCO (System for Intelligent Retrieval of Operationalized Cases and COdes) computationally models the operationalization techniques for the purpose of retrieving relevant code provisions and past cases. As described more fully in (McLaren 1999), key aspects of the model are: an ontology, including an Ethics Transcription Language (ETL) for representing the facts of engineering scenarios as narratives of chronologically ordered events, and extensions (EETL) for representing aspects of the Board's arguments for and against its conclusions, a web-based case acquisition tool, and a case base of 184 foundational cases. To input a target case, a human case enterer must encode the facts of the

<sup>1</sup> 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."

<sup>2</sup> NSPE Code I.4

case into ETL using the acquisition tool. For source cases in the case base, case enterers must also have encoded the Board's analysis into EETL.

Using the model, we tested two hypotheses:

- (1) whether a core subset of five operationalization techniques enables SIROCCO to make accurate predictions of the principles and past cases that are likely to be important in the analysis of new cases,
- (2) whether SIROCCO's temporal knowledge contributes to the accuracy of its predictions.

This work extends interpretive CBR techniques (Kolodner 1993) from the legal domain (Ashley, 1990; Branting, 1991; Rissland *et al.*, 1996; Aleven, 1997) to a new domain. Arguments in practical ethics are more free-form in style and structure than legal arguments. Ethics cases do not have binary outcomes (e.g., plaintiff wins or loses) but may require "creative middle way" solutions (Harris *et al.*, 1999, p. 64-72). SIROCCO contributes a detailed, narrative case representation, including temporal relations among facts, and an extensional model of how general principles and cases accrue meanings through operationalization. It can retrieve cases over a wider range of factual scenarios than the AI&Law programs, but unlike those programs it is not able to make arguments.

## Overview of SIROCCO

SIROCCO accepts a target case expressed in ETL and produces suggestions about relevant code provisions and past cases. A sample target case in ETL is shown in Figure 1; it deals with an engineer who has discovered structural defects in an apartment building but has been told he must keep that information confidential. SIROCCO's output for that case is shown in Figure 2, at the top of which is a textual description of the case facts. Notice that SIROCCO identifies the two general principles introduced above (i.e., NSPE Codes II.1.A and I.4) as possibly relevant and even suggests a relevantly similar case 76-1-4, in which the BER concluded the former code provision overrode the latter.

## Ethics Transcription Language

Like SWALE (Leake, 1991), SIROCCO represents ethics cases as narratives, expressed in a limited language. As shown in Figure 1, ETL represents the actions and events of a scenario as an ordered list (i.e., a *Fact Chronology*) of individual sentences (i.e., *Facts*), each consisting of (1) *Actors and objects*, instances of general actors and objects which appear in the scenario, (2) a *Fact Primitive*, the action or event in which the actors and objects participated, and (3) a *Time Qualifier*, a temporal relation that specifies how a Fact relates to other Facts in time. At least one Fact in the Fact Chronology is designated as a *Questioned Fact*, this is an action or event corresponding to an ethical question raised in the scenario. If an NSPE BER case raises more than one Questioned Fact in the context of the same Fact Chronology, it generates one SIROCCO case for each Questioned Fact.

Extracts of the ETL grammar are shown in Figure 3. The Fact Primitive in each Fact Phrase is a verb phrase that indicates a specific action or event involving actors, objects, or similarly constituted Fact-Phrases.

Time Qualifiers are disjunctive compositions of Allen's temporal constraints (1983). Since the case enterer does not provide temporal relationships among *all* Facts in a chronology, SIROCCO uses TIMELOGIC (Koomen 1989), a time propagation system, to compute the relationships by forward-chaining over the Allen relations.

1. Apartment Building <may be hazardous to safety>....	Pre-existing fact
4. Residents <file a lawsuit or arbitration action against> Bldg. Owner <because> (Apt. Bldg. <may be hazardous to safety>)....	Occurs during 3
6. Owner's Attorney <hires the services of> Engineer A <for> (Engineer A <inspects> Apt. Bldg.)....	Occurs during 4, 5
8. Engineer A <discovers that> (Apt. Bldg. <fails standards and may be hazardous to safety.>)	Occurs during 7
9. Engineer A <knows> (Government Authority <should be informed about the hazard or potential hazard>).	Occurs during 8
10. Engineer A <informs> Owner's Attorney <that> (Apt. Bldg. <fails standards and may be hazardous to safety.>)	Immediately after conclusion of 8
11. Owner's Attorney <instructs> Engineer A <to> (Engineer A <withholds information from> Anyone Else <regarding> Apt. Bldg.).	After conclusion of 10
12. Engineer A <does not inform> Anyone Else <that> (Apt. Bldg. <fails standards and may be hazardous to safety.>) [ <i>Questioned fact</i> ]	After conclusion of 11

Figure 1: Excerpts of Sample Target Case (Case 90-5-1)

Cases stored in SIROCCO's case base are represented in the *Extended* ETL (EETL), which adds a template and standard components for representing the BER's analysis of a case. The template includes the BER's conclusion (i.e., ethical, unethical, or undecided), the protagonist whose action is questioned, the ethical review Board's general argument structure, and specific information about each code provision or past case cited in support of its conclusion or noted as conflicting. When entering Case 90-5-1, the case enterer filled in the information shown in Figure 4 regarding how Code II.1.a applies (*see* note 1.)

The ontology comprises the ETL grammar (Figure 3), valid attribute values of an EETL template, and two abstraction hierarchies (not shown): (1) an *Action/Event Hierarchy* which clusters and generalizes similar Fact Primitives, (2) a *Code Hierarchy* which clusters codes dealing with similar issues. The hierarchies help define inexact matching of cases and codes, respectively.

**\*\*\* 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 example of this type of code conflict and resolution....

Figure 2: Output for Target Problem (Case 90-5-1) (excerpts)

SIROCCO’s case base includes a subset of the NSPE BER cases analyzed during the empirical study: 184 foundational cases, covering 135 different Fact Chronologies and culled from the 475 cases decided by the BER between 1958 and 1992.

The foundational cases were used to design, implement, and refine the program. These cases completely cover a reasonable number of important ethics topics (i.e., the Selected Topics) and provide some (minimal) coverage of other topics (the Non-Selected Topics.) The Selected Topics include: public safety, confidential information, duty to employer, credit for engineering work, proprietary interests, and honesty in reports and public statements. Each topic is associated with one or more of the NSPE BER ethics code provisions, all of which are represented in SIROCCO. Of the foundational cases, 135 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 reasonably widely across the topics and tend to cite different sets of codes.

```

<Fact-Chronology> := <Fact> [ <Fact> ... ]
<Fact> := <Fact-#> <Fact-Phrase> [ (Questioned Fact <X>) ]
           <Time-Qualifier> [,<Time-Qualifier>, ... ]
<Fact-Phrase> := <Fact-Primitive> [<Fact-Modifier>]
                <Actor-Or-Object>
                [ <Actor-Or-Object> | (<Fact-Phrase>) ]
                [ <Actor-Or-Object> | (<Fact-Phrase>) ]
<Fact-Primitive> := An instance of a Fact-Primitive
<Actor-Or-Object> := An instance of an Actor or an Object
< Fact-Modifier> := partially | substantially | limited |
                  extensive
<Time-Qualifier> :=
  Pre-existing fact |
  After the start of <Fact-#> [,<Fact-#>, ... ] |
  Starts at the same time as <Fact-#> [,<Fact-#>, ... ] |
  After the conclusion of <Fact-#> [,<Fact-#>, ... ] |
  Immediately after the conclusion of <Fact-#>
                                     [,<Fact-#>, ... ] |
  Ends <Fact-#> [,<Fact-#>, ... ] |
  Occurs during <Fact-#> [,<Fact-#>, ... ] | ...
Key: | = Alternative; [ ] = Optional; <> = Grammar element

```

Figure 3: ETL Grammar Extracts

<b>Code</b>	II.1.a
<b>Code Status</b>	Violated
<b>How Cited</b>	Explicitly discussed
<b>Grouped With</b>	None
<b>Overrides</b>	II.1.c
<b>Why relevant</b>	Engineer’s judgment is overruled in a particular professional circumstance. [11] Overruling the Engineer’s judgment may lead to the endangerment of the safety, health, property or welfare of the public. [8, 9]
<b>Why violated, not violated,...</b>	In the given situation, Engineer does not hold paramount the safety, health, property, and welfare of the public. [12]

Figure 4: EETL Table for Code II.1.a in Case 90-5-1

Twelve independent case enterers transcribed the foundational cases into EETL using the case-acquisition web site (www.pitt.edu/~bmclaren/ethics). The web site contains instructions on how to transcribe ethics cases into EETL, a reference shelf of useful materials, including the full vocabulary of EETL, and an example set of 47 transcribed Fact Chronologies.

### Operationalizations

As noted, our investigation of the BER’s case explanations revealed nine operationalization techniques:

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.
4. Arbitrating between competing principles.
5. Grouping principles.

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.

Our experiment tested the contributions of a core subset, those techniques which contribute directly to suggesting relevant cases and codes (i.e., techniques 1, 5, 6, 8, and 9). The remaining techniques focus on providing explanations.

Code and case instantiations (i.e., techniques 1 and 6) are the primary operationalization techniques by which codes and cases become defined extensionally in a way that SIROCCO can reuse in analyzing new cases. Instantiating a code or past case means relating a questioned fact, certain critical facts, and the temporal sequence of those facts in the citing case to the cited code provision or case. Case enterers recorded information instantiating codes or past cases, grouping them or reusing previous operationalizations in the EETL tables such as that in Figure 4. The numbers in brackets [ ] in the last two rows of Figure 4 refer to those facts in the representation of Case 90-5-1 (Figure 1) that are critical to the code's application and explain why it was [not] violated. In this way, Code II.1.a is connected extensionally to a real case's relevant facts and chronology in a way that SIROCCO can reuse.

The NSPE BER's groupings of principles (i.e., code provisions) and cases in arguments (operationalization techniques 5 and 8) inform SIROCCO's selection heuristics as do the Board's reuse of past operationalizations (technique 9) (e.g., the case and code instantiations in a cited case can be reused.)

### SIROCCO's Two-Stage Retrieval Process

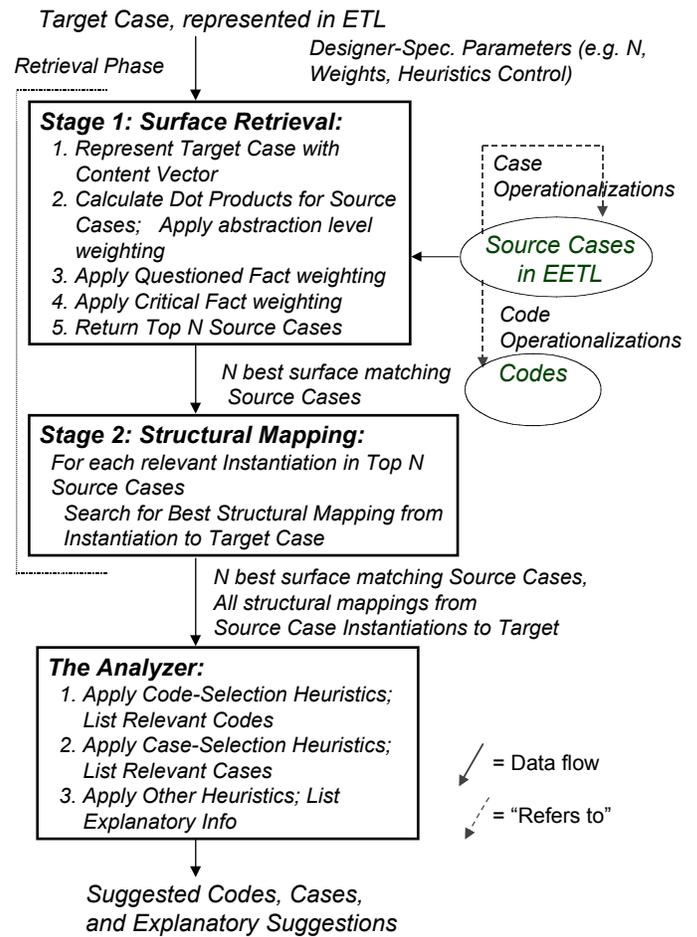
Following the general approach in designing analogical retrieval programs (Forbus et al. 1994; Thagard et al. 1990), SIROCCO's retrieval phase is implemented as a two-stage algorithm, as shown in Figure 5. Stage 1 rapidly matches the target case's Fact Primitives to those of all possible source cases. Stage 2 applies a more-expensive A\* search to map selected case structures between target and source cases.

In both stages, instantiations (i.e., operationalization techniques 1 and 6, above) help SIROCCO focus attention on the most critical facts. Stage 1's accuracy improves by giving more weight to the instantiations' Fact Primitives. In Stage 2 focusing on only the part of a source case's Fact Chronology in an instantiation makes the structural mapping routine more efficient and accurate.

For Stage 1, the target case and all of the cases in the case base are represented as content vectors (Forbus et al. 1994). 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 90-5-1. The left (Fact-

Primitive) is the most specific; the right (Fact-Group) is one level higher in the Action/Event Hierarchy.

Figure 5: SIROCCO's Architecture



Stage 1 computes the weighted dot products of content vectors for the target case and all cases in the case base. It outputs a list of candidate cases ranked by descending dot product scores. Different weights have been assigned to matches at the four abstraction levels (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 critical and questioned facts.

Using a heuristic A\* search, Stage 2 attempts a structural mapping between the target case and each of the N top-ranking candidate source cases from Stage 1. (Branting 1991) first used A\* search for case structure mapping; SIROCCO also takes temporal relations into account, supports abstract matches, and accommodates a wider range of scenarios. The search focuses on matching the source case's instantiations (i.e., operationalization techniques 1 and 6). The goal is to map each of the Facts of the source instantiation to a corresponding Fact in the target case while maintaining a one-to-one and consistent mapping between the Actors and Objects of the source and the target. The initial node of the search space maps the source's questioned Actor to the target's. Each subsequent node represents: (1) a proposed mapping of a pair of Facts,

one from the source instantiation and one from the target case, (2) all of the Fact mappings that preceded this node (i.e., all of the successful Fact mappings from ancestor nodes), (3) a one-to-one, consistent set of Actors and Objects entailed by the Fact mappings, and (4) consistent temporal relations between 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.

Fact-Primitive-CV:	Fact-Group-CV:
(May-be-Hazardous-to-Safety 1)	(Deal-with-Potential-Dangers-or-Hazards 1)
(Owns 1)	(Own-Something 1)
(Resides-in 1)	(Specify-Location-of-Residence 1)
(Files-a-Lawsuit-or-Arbitration-Action-Against 1)	(Initiate-Legal-or-Arbitration-Proceedings 1)
(Is-Legally-Represented-by 1)	(Has-Legal-Representation 1)
(Hires-the-Services-of 1)	(Work-as-an-Employed-or-Contract-Professional-Engineer 1)
(Inspects 1)	(Perform-Engineering-Analysis-Review-or-Testing-Work 1)
(Discovers-That 1) (Knows 1)	(Know-or-Believe-Something 2)
(Informs-That 2) ***	(Disclose-Information 2) ***
(Instructs-to 1)	(Order-Subordinate-to-Perform-Task 1)

Figure 6: Content Vectors for Case 90-5-1

New nodes are generated from an existing node by selecting an unmapped Fact from the source case instantiation and mapping it to each of the target's unmapped Facts in which the corresponding Fact Primitives match either exactly or abstractly. 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.

Each node is evaluated in terms of the A\* cost function,  $f(n) = g(n) + h'(n)$ , which is calculated as follows. The mismatch cost  $g(n)$  is a summation of 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) are: 0.0 for an exact Fact Primitive match, 0.4 for a match at the Fact Group level, 0.6 for a match at the Sibling Group level, 0.9 for a Fact Root level match and 1.0 for a completely failed match.

The  $h'(n)$  function evaluates the most optimistic possible completion of the mapping from node  $n$ . It provides 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. The solution depth is always fixed to be the number of Facts in the instantiation (as noted, even if a Fact does not match,

an "empty" match node is created.) SIROCCO always returns the minimum  $f(n)$  found at the fixed solution depth.

The goal node is reached when the current depth equals the pre-defined solution depth 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 completing Stage 2, SIROCCO's Analyzer phase assesses the results of Stages 1 and 2 and lists suggested codes, cases and provides some additional explanatory information (as in Figure 2). A set of selection heuristics that embody Operationalization Techniques 1, 5, 6, 8 and 9 generate the lists of possibly relevant codes and cases. The heuristics favor codes that, for example, (1) occur more frequently in the top-ranked cases of Stage 1 or (2) have code instantiations with a low Stage 2 mismatch score and at least a minimum number of matches to the source's critical facts. Similar heuristics are used to select cases. Other heuristics embody the remaining Operationalization Techniques and generate explanatory information.

## Evaluation

A series of experiments were performed with SIROCCO to test the two hypotheses. The data included the 184 foundational cases and a set of 58 trial cases that were decided later than the foundational cases. All of the trial cases were transcribed into the extended ETL by two independent case enterers and, to ensure objectivity, were provided unaltered to SIROCCO for processing. The 58 trial cases were chosen from two pools of cases within a set of 77 cases decided by the NSPE BER between 1993 and 1998: 44 trial cases were chosen at random from 52 Selected Topics cases and 14 trial cases were chosen at random from 25 Non-Selected Topics cases.

To test the first hypothesis, SIROCCO was pitted against five competitor methods, shown in Figure 7. Each method, including SIROCCO, was given the entire set of trial cases to process one-by-one, and the retrieval results of each method were then compared to the BER's code and case citations for the same case. To calculate overlap between the method's solution and the Board's solution, we employed the *F-measure*, an information retrieval metric that combines precision  $P$  and recall  $R$ :  $F = (\beta^2 + 1)PR / (\beta^2 P + R)$  (Lewis *et al.*, 1996). The value of  $\beta$  was 1.0. Two F-Measure values were computed for each case, one representing exact matches of codes and cases between the method's solution and the Board's and one representing inexact matches. Inexact matches of codes were determined using the Code Hierarchy. Inexact matches of cases were determined using a citation overlap metric, inversely related to the shortest 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 and Ashley 1999).

**RANDOM:** Codes/cases randomly selected for each case.

**INFORMED-RANDOM:** Like RANDOM but accounts for frequency distribution of code/case citations in NSPE corpus.

**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.

**NON-OP SIROCCO:** Ablated version of SIROCCO, with almost no functionality related to operationalizations.

Figure 7: Five Methods Compared to SIROCCO

The results are shown in Figure 8. Since the data generated by benchmarking each method against the BER’s citations using the F-Measure was highly non-Gaussian, we applied a *nonparametric bootstrap procedure* (Davison and Hinkley 1997) to compare SIROCCO with the other methods. According to the procedure, the probability that SIROCCO was more accurate than the other five methods was greater than 95% in every instance except with respect to 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 EXTENDED-MG in retrieving *exact* codes and cases. It was also more accurate in retrieving *inexact* codes and cases, but the difference was not statistically significant.

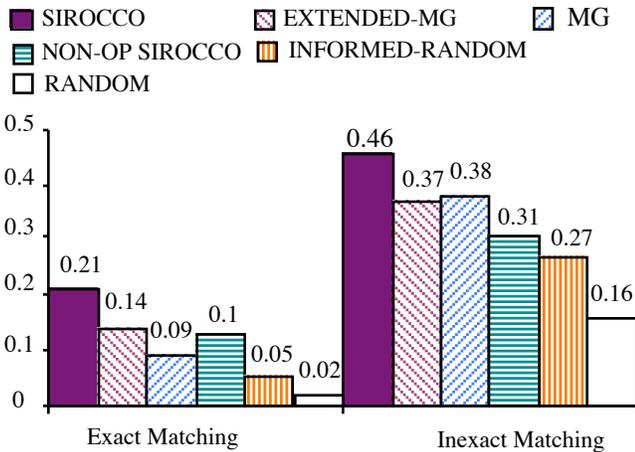


Figure 8: Mean F-Measures For All Methods Over All Trial Cases

The above experiment penalized SIROCCO for citing relevant codes and cases that the BER happened not to have cited. In a supplemental experiment, we asked two ethics graduate students to evaluate the extra code and case citations for the trial cases made by SIROCCO and SIROCCO’s closest competitor, 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.

After verifying that inter-rater reliability was satisfactory, we recalculated SIROCCO’s and EXTENDED-MG’s F-Measures for the 58 trial cases, counting the extra citations rated as “reasonable” by the evaluators as Board citations. For SIROCCO, the recalculated mean F-Measures were 0.36 for exact

matching and 0.58 for inexact matching. For EXTENDED-MG, the recalculated mean F-Measures were 0.25 for exact matching and 0.46 for inexact matching. The nonparametric bootstrap procedure now showed a significant difference between the accuracy of SIROCCO and EXTENDED-MG on both the exact and inexact match criteria. For both criteria, the confidence level of a difference (in favor of SIROCCO) was now at least 99%.

We compared SIROCCO’s predictive capability for cases within and outside the Selected Topics. Outside its primary area of expertise, SIROCCO’s predictive capability degraded slightly. The differences between the mean F-measures for Selected Topics trial cases and Non-Selected Topics trial cases were:  $0.23 - 0.15 = 0.08$  on exact matching, and  $0.47 - 0.44 = 0.03$  on inexact matching.

To test the second hypothesis (i.e., that SIROCCO’s temporal knowledge makes a difference in the accuracy of its predictions) the trial cases were processed by an ablated version of the program, NON-TEMP SIROCCO, that did not employ temporal knowledge. NON-TEMP SIROCCO provides the full functionality of SIROCCO with the exception that it doesn’t check the consistency of temporal relations across matched cases. As with the initial experiments, the results of NON-TEMP SIROCCO were compared against the suggestions made by the ethical review Board and the F-Measure calculated for each individual sample and as a mean value over all samples. These results were then compared to the output of the standard version of SIROCCO, which did apply temporal knowledge. The differences between SIROCCO with and without its temporal knowledge were essentially negligible.

## Discussion and Conclusion

The experiments confirmed that the core operationalization techniques allow SIROCCO to make accurate predictions of the principles and past cases that are likely to be relevant in the analysis of new cases. This is evidence that the BER’s explanations extensionally define applicability conditions for the code provisions and past cases.

The fact that SIROCCO significantly outperformed NON-OP SIROCCO is the strongest evidence that SIROCCO’s operationalization techniques do, in fact, make a difference. SIROCCO and NON-OP SIROCCO share the same case representation and Stage 1 retrieval method. The critical difference is their use of the operationalization techniques. NON-OP SIROCCO makes only weak use of an operationalization technique. Like EXTENDED-MG, it selects codes that appear most frequently in the list of  $N$  top-rated cases. By contrast, SIROCCO makes extensive use of the core operationalization techniques in performing its retrieval task.

Outperforming MG and EXTENDED-MG shows that SIROCCO is a powerful retrieval method. These full-text retrieval methods are strongly competitive alternatives for performing SIROCCO’s task. The fact that SIROCCO outperformed EXTENDED-MG which, in turn, outperformed MG is also significant. Since EXTENDED-

MG makes use of an operationalization technique, its improvement over MG also supports our first hypothesis.

We were unable to show that SIROCCO's temporal knowledge contributed to its accuracy. This result surprised us. Intuitively, temporal orderings of events are important in ethical analysis. For instance, one can be expected to report a dangerous situation only after one has learned of it, not before. Then it occurred to us that the latter case would not appear in the NSPE BER cases because it so obviously does not involve a moral duty. In such a case base it might be a rare event that pairs of cases exist such that a difference in temporal ordering leads to different ethical interpretations.

This work suggests how to design intelligent aids for retrieving principles and cases in fields like professional ethics where intelligent access to the right standards and examples 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. Of course, 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. We plan, however, to incorporate SIROCCO into a tutoring environment for practical ethics. Expending some effort in representing problems has pedagogical value (e.g., it induces students to consider more carefully the facts of a case.) SIROCCO's explanations of its outputs are also pedagogically useful. Full-text retrieval schemes cannot generate such explanations. Finally, other work explores how to use CBR knowledge representations to guide a program in learning how to assign indices to textual cases automatically (Brüninghaus and Ashley 1999). SIROCCO's case representation may support an automated approach to processing textual cases in ethics.

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

## References

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

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

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

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

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 (ICCB-99)* Lecture Notes in AI No. 1650, 59-73. Springer: Berlin.

Clancey, W.J. 1983 The Epistemology of a Rule-based Expert System: A Framework for Explanation. *Artificial Intelligence*, 20(3):215-251.

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

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

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

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

Kolodner, J. 1993. *Case-Based Reasoning*. San Mateo: Morgan Kaufmann.

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

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

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.

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

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 (ICCB-99)*. Lecture Notes in AI No. 1650. 248-262. Springer: Berlin.

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

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.

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