

# Exploring the Dialectic Between Abstract Rules and Concrete Facts: Operationalizing Principles and Cases in Engineering Ethics

Bruce McLaren and Kevin Ashley

University of Pittsburgh  
Intelligent Systems Program  
3939 O'Hara Street  
Pittsburgh, PA 15260

bmclaren+@pitt.edu, ashley+@pitt.edu

**Abstract.** Abstract rules and specific fact situations interact in a highly complex fashion in engineering ethics, a weak analytic domain. In such domains, the construction of arguments or explanations does not rely on formal methods or proofs. Rather, experienced reasoners appear to address problems by applying ethical principles using a variety of techniques. In our study of a national engineering society's set of engineering ethics cases decided by an ethical review board, we have identified a number of *operationalization techniques* which help to fill the gap between abstract principles and specific case facts and which help to analyze new problems. Our goal is to develop a computational model that is capable of retrieving and applying operationalizations for the purpose of making accurate predictions of the facts, principles, and past cases that would be regarded as important in the analysis of new cases. In this paper, we present a preliminary design of such a model and outline an experiment to test it. We expect to make a contribution to interpretive case-based reasoning (CBR) by shedding light on the role of principles in decision making, by investigating the connection between abstract rules and concrete facts, and by testing the feasibility of using detailed, factual chronologies to represent cases.

*Submitted to and accepted for the European Workshop on Case-Based Reasoning, Dublin, Ireland, 1998*

## 1. Introduction

We are studying how abstract principles and specific fact situations interact with one another in the weak analytic domain of engineering ethics and, more particularly, in a case base of hundreds of ethics cases decided by a board of experienced ethical reasoners. We are in the process of designing, developing, and testing a computational model that uses relevance information from the board's explanations of how ethics principles (i.e., ethics code provisions) and past cases affected their decisions. Our computational model will be able to use the abstracted relevance information to frame analyses of new engineering ethics cases. The model is not intended to reach conclusions for new cases but to identify relevant information for the analysis. In a controlled experiment for a set of trial cases, we will compare the model's identification of relevant information with that of the board.

Our model is aimed at *operationalizing principles and cases* in engineering ethics. By "operationalizing" we mean making a principle or case useable by a computational method for purposes of relevance assessment and explanation. Our goal is to build a model of the information and processing required to operationalize principles and cases. Given a new engineering scenario which raises an ethical question (or questions), the computational model will predict the facts, principles, and past cases that the board of ethical review would regard as important in the analysis of the new problem. The model's predictions will be based on the board's code-based analyses of past cases. The model will also reveal the reasons for its predictions.

We hypothesize that: (1) past decided professional engineering cases operationalize ethical principles and cases by interpreting, arbitrating among and revising the principles and cases, by analogizing and distinguishing precedents, and by posing hypothetical variations of the problem facts. (2) A computational model of such operationalization can make accurate predictions of the facts, principles, and past cases that would be regarded as important in the analysis of new cases. Our model will also test the feasibility of using detailed, chronological facts as a basis for "interpretive" CBR (Kolodner, 1993).

The computational model's approach is based on an empirical study of how experienced professional engineers use principles and specific fact situations to resolve ethical issues and to justify their decisions. In particular, we have studied published opinions of the National Society of Professional Engineers' Board of Ethical Review (NSPE BER) and its code of ethics (NSPE, 1958-1996). Composed of five to seven professional engineers annually, the NSPE BER has written extensive explanations of how and why codes apply or do not apply to approximately 400 particular fact situations. The NSPE's code of ethics, consisting of 74 individual codes, provides engineers with guidance on issues such as public safety, misrepresentation in advertising, conflicts of interest, and confidentiality.

## 2. Abstract Rules and Concrete Facts

The interaction between principles, codes, and concrete fact situations in the NSPE BER engineering ethics domain is a highly dynamic and evolving process. The board

resolves cases by (1) applying principles, codes, and past cases, (2) storing the cases, and then (3) reapplying (and, perhaps, reinterpreting) the past cases and code provisions. The process is also nonmonotonic. New codes are introduced, old codes are reworded or deleted, and new interpretations of codes (and cases) are provided in the context of new cases. In essence, the principles and codes act like "guideposts," focusing engineers on important issues and dimensions of fact situations. Their evolving, abstract nature, the fact that they require interpretation, re-interpretation, and grounding in past cases, defies definition in terms of logical rules.

Little empirical work in AI focuses on the role abstract principles play in computational models of decision making. While reminiscent of hybrid case-based/rule-based architectures (e.g., Rissland and Skalak, 1991; Golding, 1991), in this iterative, dialectical process rules and cases interact in an essential way; neither could satisfactorily support decision-making without the other. Karl Branting sketched a computational model to "bridge the gap" between legal theories (similar in some respects to principles) and specific case facts using the concept of *ratio decidendi*, a theory to determine a precedent's controlling effect, and a warrant hierarchy relating abstract predicates to other predicates and, eventually, to specific facts of precedent cases (Branting, 1994). In (Rissland *et al.*, 1996a), legal theories are part of a network that BankXX searches to "harvest" information for one side or another in a legal dispute.

As compared to legal reasoning, our domain involves a less explicit model of argumentation. Engineering ethics problems are also not constrained to binary conclusions (e.g., plaintiff won or lost) but may involve multiple suggested actions and outcomes. The range of represented cases we intend to represent is greater than in any of the AI & Law programs. Essentially, the operationalization techniques we have observed and will implement are methods for identifying and marshaling information for problem analysis. Though some of the operationalization techniques are found in legal reasoning models, they have yet to be integrated in a single model of reasoning with specific cases and abstract principles.

The NSPE BER case base and the reasoning strategy of the board embody a casuistic approach to ethical reasoning which inspired our earlier work on TRUTH-TELLER (McLaren and Ashley, 1995; Ashley and McLaren, 1995). Casuistry is a form of ethical reasoning in which decisions are made by comparing a problem to paradigmatic cases (Jonsen and Toulmin, 1988; Strong, 1988). We designed and tested a program which compared pairs of cases selected at random from a set of ethical dilemmas involving telling the truth. In the present work we focus on a model of retrieving relevant cases and principles for analyzing a greater range of problems represented in more detail than in TRUTH-TELLER.

### **3. An Example Domain: Engineering Ethics**

Instead of measuring a principle's "weight" quantitatively or applying principles deductively, decision makers in engineering ethics appear to reason symbolically and qualitatively with principles. They identify which facts are relevant in light of applicable principles, resolve conceptual issues by defining terms of the principles and

their application to case facts, and engage in moral reasoning (e.g., use past cases for "line drawing" comparisons and employ creative middle-way solutions) (Harris *et al.*, 1995).

A system of explicit middle-level ethical principles tailored to engineering ethics is known as a *code of ethics*. Although an engineering code of ethics provides "rules" of ethical behavior for practicing professional engineers, typically the principles are stated in an abstract and sometimes quite complicated fashion. As a result, applying the codes to particular fact situations is not trivial. Consider, for example, the following code from (NSPE, 1958-1996):

"II.5.a. Engineers shall not falsify or permit misrepresentation of their ... academic or professional qualifications. They shall not misrepresent or exaggerate their degree of responsibility in or for the subject matter of prior assignments. Brochures or other presentations incident to the solicitation of employment shall not misrepresent pertinent facts concerning employers, employees, associates, joint ventures or past accomplishments with the intent and purpose of enhancing their qualifications and their work."

Each of the three sentences in the code deals with a different aspect of "misrepresentation of an engineer," and each sentence covers a wide range of possible circumstances. The precise circumstances that support application, however, are not specifically stated. Knowing whether this code applies to a particular fact situation requires that one must recognize the applicability of and interpret ill-defined terms in the code.

<p><b>Facts of Case 90-4:</b> Engineer X is employed by Firm Y, a medium-sized engineering consulting firm controlled by Engineer Z. Engineer X is one of a few engineers in Firm Y with expertise in hydrology, but the firm's work in the field of hydrology does not constitute a significant percentage of the firm's work. Engineer X, an associate with the firm, gives two weeks notice of her intent to move to another firm. Thereafter, Engineer Z, a principal in Firm Y, continues to distribute a brochure identifying Engineer X as an employee of the firm and list Engineer X on the firm's resume.</p> <p><b>Question:</b> Was it ethical for Engineer Z to continue to represent Engineer X as an employee of Firm Y under the circumstances described?</p>
---

Fig. 1. An NSPE BER case

Determining whether code II.5.a. applies to the particular facts of, for instance, the NSPE BER case depicted in Fig. 1, requires interpretation of open-textured concepts. The board must determine whether Engineer Z's advertising of Engineer X as an employee constitutes a "misrepresentation of the pertinent facts" and whether there was "intent and purpose" to enhance Firm Y's qualifications.

#### 4. Operationalization Techniques

The gap between the fairly abstract ethics codes and the cases' relatively detailed factual circumstances presents difficulties for both ethical reasoners and modelers of ethical reasoning. Our analysis of the NSPE BER case base has uncovered operationalization techniques that the board employs to annotate analyses of engineering ethics cases. Fig. 2 briefly describes the techniques we have cataloged during our analysis of the NSPE BER cases. With these techniques, the board justifies and explains the

applicability and significance of relevant code provisions and past case decisions to the analysis of the current problem. These operationalizations, we maintain, provide constraints on and a context for the use of codes and cases which can be applied to the analysis of new cases.

In resolving the issue in Fig. 1, for instance, the board "instantiated" two codes in stating the issues, "cited a distinguishing precedent" and "reused an operationalization," (i.e., an interpretation of terms from a past case analysis).

<u>Code Operationalizations</u>	
⇒	<b>Code Instantiation.</b> Typically, only a subset of the case facts imply that a particular code or codes applies to a case. In characterizing the issue, the board will either implicitly or explicitly connect the relevant facts to the relevant codes.
⇒	<b>Interpret Code.</b> In the context of a case, a code's conditions may be explicitly interpreted and applied in a way that more clearly defines the code. The specific types of code interpretation employed by the board are: <ul style="list-style-type: none"> <li>- <b>Interpret Code in Context.</b> The board may align a code with the specific circumstances of a case by rephrasing or rewording the code.</li> <li>- <b>Define Terms of Code.</b> Specific terms or the language of a code may be ambiguous or ill-defined. The board may define such terms in their analyses of particular cases.</li> <li>- <b>Make Code More Specific.</b> Many of the codes are very broadly defined. A typical tactic employed by the board is to specify the conditions of a code in the context of a case.</li> <li>- <b>Make Code More General.</b> Occasionally, the board will broaden the conditions of a code to cover the particular circumstances of a case.</li> </ul>
⇒	<b>Apply Hypothetical to a Code.</b> The board proposes hypotheticals to help establish the range of application of a principle. Hypotheticals are also used to fill out unknown, yet likely, facts of a story.
⇒	<b>Rewrite Code.</b> A controversial or ill-defined principle may explicitly change or be reinterpreted over time by the board. For instance, the wording of a principle may conflict with changing cultural norms. The board "rewrites" a principle in a case analyses by recommending that the code be formally changed.
⇒	<b>Define Code Superiority.</b> Sometimes two (or more) principles conflict with one another in the context of a particular case. The board identifies such an event and explains the underlying reasons for the "superiority" of one principle over others.
⇒	<b>Group Codes.</b> The board occasionally cites related codes as a group because they may all apply, at least abstractly, to the case in question and thus provide greater support of a particular conclusion.
<u>Case Operationalizations</u>	
⇒	<b>Cite a Precedent.</b> The board frequently cites previous cases that may be instrumental in deciding the case in question. Two types of "precedent" citations are employed: <ul style="list-style-type: none"> <li>- <b>Cite Analogous Precedent.</b> The board cites an analogous precedent to argue that its conclusion should be followed in the case in question. That is, the precedent case "rules" the present case.</li> <li>- <b>Cite Distinguishing Precedent.</b> The board sometimes cites a precedent to argue that, although it has similarities to the present case, its conclusion should not be followed due to relevant differences between the precedent and the present case.</li> </ul>
⇒	<b>Reuse an Operationalization.</b> A past case may be cited in order to reuse a previous operationalization. For instance, a past case (or cases) may define terms of a code that are relevant to a new case.
⇒	<b>Define or Elaborate a General Principle.</b> In some instances, a past case is cited because it discusses a relevant general principle, policy, or rule.
⇒	<b>Group Cases.</b> Cases are sometimes cited in tandem for a single illustrative purpose or to apply greater force to a conclusion. In effect, the set of cases acts a "single case" in the context of a specific case.

Fig. 2. Operationalization Techniques employed by the NSPE BER

## 5. How the Computational Model Will Operate

Our computational model will take the following approach. Given a case base of engineering ethics dilemmas that have been transcribed into a restricted, domain-specific language, the model will:

1. Accept a new fact situation that has been transcribed into the domain-specific language;

2. Match the new fact situation to existing cases using a two-step structural mapping that focuses on the most critical facts and on the chronology of facts; and
3. Frame analysis of the new fact situation by employing selection heuristics to (a) reuse the operationalization techniques from relevant past cases and (b) compare the new case to the relevant past cases.

While the design of the computational model is not finished, we have incorporated certain novel features to accommodate the representational demands and techniques revealed by our analysis of the NSPE BER domain. For instance, a particular type of structural relationship -- event chronology -- seems to play a critical role in similarity assessment.

### 5.1. Representing the Facts of a Case

Each ethics case in the NSPE BER case base will be represented in terms of a chronology of actions and events in the scenario and the relationships among the actors and various objects. The facts of the cases will be represented in a restricted, domain-specific language specifying the types of actors, actions, events and objects common to the domain (e.g., engineers, clients, designing, testing, advertising, engineering projects).

For example, at the top of Fig. 3, the fact description of Case 90-4 from Fig. 1 is transcribed as a *fact chronology* representing the story's discrete and relevant facts. Each fact in the chronology is composed of a *fact primitive* (the boldface text), one or more actors, objects, or other facts, and a *time qualifier* (the bold-italics text). A fact primitive is essentially a function ranging over actors, objects, and other facts. Fact primitives are organized in a taxonomy; thus specific primitives may match one another abstractly.

#### *Fact Chronology of Case 90-4:*

1. Engineer X <b>and</b> Engineer Z <b>and</b> Other Engineers <b>are employed by</b> Firm Y.	<i>Pre-existing fact</i>
2. Engineer X <b>and</b> Other Engineers <b>specializes in</b> Hydrology Engineering.	<i>Pre-existing fact</i>
3. Firm Y <b>does not specialize in</b> Hydrology Engineering.	<i>Pre-existing fact</i>
4. Engineer X <b>submits resignation to</b> Firm Y.	<i>After the start of 1, 2, 3</i>
5. Engineer Z <b>advertises or solicits engineering business using</b> (Engineer X <b>is employed by</b> Firm Y). ( <i>Questioned Fact</i> )	<i>Immediately after the conclusion of 4</i>

#### *Content Vector of Case 90-4:*

- (is employed by . 4)
- (specializes in . 2)
- (does not specialize in . 1)
- (submits resignation to . 1)
- (advertises or solicits engineering business using . 1)

**Fig. 3.** The Transcription and Content Vector of Case 90-4

Time qualifiers, based on the temporal constraints of (Allen, 1983) but defined at a more abstract level, specify the chronology of and the temporal relationship between the facts of a case. For instance, the "After the start of" qualifier in fact 4 of Fig. 3 specifies that Engineer X submits his resignation sometime after the first three facts have started. Thus, "After the start of" is based on a combination of the following group of Allen's constraints: (After, During, Finishes, Met by, or Overlapped by)

For each case, at least one fact is "questioned." Each questioned fact corresponds to a question found under the "Question" heading in the original text of the case. The board decides whether the questioned fact was ethical or not under the case circumstances. In Fig. 3, only one ethical question is raised in the scenario, as represented by line 5 of the fact chronology. Also shown in Fig. 3 is the *content vector* derived from the fact chronology. A content vector specifies the primitive facts that were used and how many times each appeared in the chronology.

We have developed a preliminary version of the language based on our study of a foundational set of 327 cases, and we are iteratively testing and developing it. Presently, there are 174 fact primitives and 66 actor and object types in the language. We have thus far modeled 47 of the NSPE BER cases using the preliminary language.

We have tested the preliminary language by asking 4 graduate students to model 13 cases each. Ten of the 13 cases given to each student had not been modeled prior to the exercise, thus our existing set of primitives was tested against novel cases. The students were given a guide containing the primitives for transcribing cases. They were instructed to try to use the given primitives, but were also encouraged to add new primitives, as required. The preliminary results are very encouraging. The 40 transcriptions of new cases contained 79.2% pre-existing fact primitives and 90.6% pre-existing actors and objects, and many of the new primitives suggested by the participants will be useful additions to the language.

## 5.2. Representing the Analysis of a Case

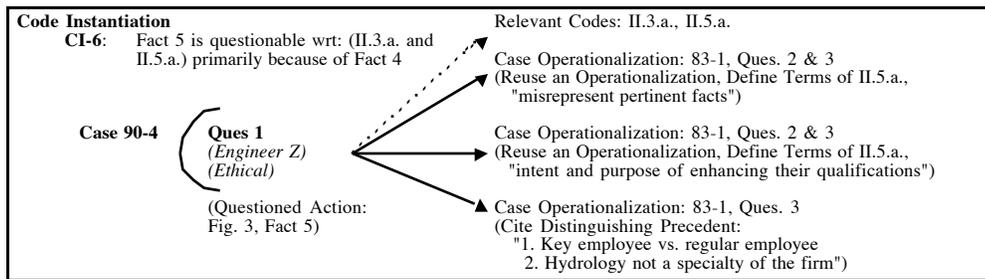
The key element of the analysis representation is a set of operationalizations employed by the board in analyzing the case. The operationalizations are, in effect, pieces of the review board's justification of a conclusion. They are represented in a constrained language and format indicating the type of operationalization (e.g., "Define Terms of Code," "Cite Analogous Precedent") and certain type-specific information such as related cases or codes, case facts that are relevant to the operationalization, and the outcome supported by the operationalization.

A *code instantiation* is a special type of operationalization that succinctly states an issue presented in the case. Each code instantiation relates one or more code provisions to a questioned fact, certain critical facts, and the facts' temporal relationships. A code instantiation is intended to capture the minimal collection of facts which the board has stated (or implied) invokes a code provision. The case transcribers identify code instantiations in decided cases based upon their understanding of the board's rationale in citing a code provision.

The board's analyses will be represented as: (1) the action questioned, (2) the actor(s) whose action it is, (3) the codes the board deemed relevant to the case and their instantiations, (4) the board's conclusion regarding the questioned action (e.g., ethical, unethical, ethical with a caveat), and (5) a list of the code and case operationalizations employed by the board during its discussion and analysis of the case.

An abbreviated version of the analysis representation of Case 90-4 is shown in Fig. 4. The actor whose action is questioned and the outcome assigned by the board are displayed beneath the question. The dashed arrow points to the codes that the board deemed relevant to the scenario including Code II.5.a., discussed in Section 3.

The code instantiation CI-6 shows why the board applied these two codes. CI-6 was defined because Z's advertising of X as an employee (fact 5, Fig. 3) occurred after X had submitted her resignation to Firm Y (fact 4, Fig. 3). This fact is questionable with respect to Code II.5.a. and Code II.3.a. ("Engineers shall be objective and truthful ..."). Note that code instantiations are not intended to be "rules" for concluding whether the questioned fact is ethical or unethical under that code provision. The instantiation only indicates that the code is relevant to the questioned fact. In Case 90-4, for instance, although the indicated codes "apply" to the fact situation, the board ultimately determines that Engineer Z's action was ethical.



**Fig. 4.** Abbreviated Analysis Representation of Case 90-4

The solid arrows in Fig. 4 point to operationalizations employed by the board in their analysis of the case. As discussed when we introduced Case 90-4, the board depended upon a previous case's interpretation of code II.5.a. and a comparison to that case in its analysis. In the interest of brevity and clarity, detail is omitted from the representation of the operationalizations. For instance, the full representation of "Cite Distinguishing Precedent" would include some mapping between the specific facts of 90-4 and 83-1 that lead to the distinguishing features.

### 5.3. Retrieving and Matching Cases

Since we intend to represent a wide range of engineering ethics scenarios as a chronology of actions and events, matching problems and cases will require structure mapping techniques (Holyoak and Thagard, 1995; Gentner, 1983; Branting, 1991), but extended to take chronological ordering among steps into account. We need to retrieve and compare cases using the inherent structure and complexity of our language.

A structural mapping is essentially a correspondence of relations, functions, and objects established between two cases and the evaluation of that correspondence. Unfortunately, such a mapping is typically expensive when used for retrieval. In our engineering ethics domain, correspondences would need to be established and evaluated between the actions, events, relationships, actors, objects, and chronology of an input case and the many cases in a large case base.

Following other structural mapping solutions, we intend to address the problem by developing a two-stage algorithm. The first stage will retrieve a preliminary set of matches, based on superficial criteria (i.e., mapping of non-structural elements) and

the second stage will apply a more complex structural mapping between the input case and those cases retrieved by the first stage. Provided the first stage retrieves the best candidates in most situations, this approach represents a good trade-off between accuracy and efficiency.

Although we have not yet designed an algorithm, we are considering something similar to MAC/FAC (Forbus *et al.*, 1994) but extended to account for chronological ordering. For illustrative purposes, the discussion and example in Section 6 employs elements of the MAC/FAC model. For instance, we use a content vector as a superficial summary of the fact chronology and as a means to support the first stage of retrieval. On the other hand, the particular characteristics of our problem and domain - in particular, the emphasis on event chronology -- argue for extending or modifying MAC/FAC's second stage.

#### 5.4. Identifying and Displaying Critical Analysis Information

The final stage of the computational model is the *Analyzer*. The purpose of the Analyzer is to determine the portions of past analyses that may be relevant to and reusable in the analysis of the input case. The Analyzer presents the user with suggestions for analyzing the input case, based on its retrieval of relevant past cases. Essentially, each suggestion will correspond to an operationalization from a past case. The suggestions will be organized by categories of interest (e.g., possibly relevant codes, possible case citations, etc.). The computational model will not reach a conclusion (i.e., ethical, not ethical, take a particular action) to the ethical question posed in the case, nor will it develop an argument or arguments for one position or another. Rather, the output will provide the raw material from which one could develop an argument or reach a conclusion. An example of the type of output we expect to produce is depicted in Fig. 8 and will be discussed in detail in the example section of this paper.

- |   |
|---|
| <p><b>H1:</b> <i>Suggest operationalizations that are employed by more than one matching case.</i> If particular operationalizations are repeated within the set of best-matching cases, they are prime candidates to be suggested as relevant for the input case.</p> <p><b>H2:</b> <i>Suggest operationalizations associated with codes that are highly likely to be applicable.</i> A code that is "highly likely to be applicable" is defined as how often the code appears in matching cases and whether the code was highly rated during code retrieval.</p> <p><b>H3:</b> <i>Suggest analogies/distinctions with matching cases.</i> If the program identifies compelling analogies to or distinctions with matching cases, it should suggest them. Compelling analogies and distinctions will be identified during the structural analysis performed in the structural mapping stage.</p> <p><b>H4:</b> <i>Suggest case citations that are more "central" to the input case, as determined by citation statistics.</i> Decide whether a matching case (or a case cited by a matching case) should be cited in the new case by checking case citation statistics. Give preference to the case more often cited.</p> <p><b>H5:</b> <i>Suggest code citations that are referenced by the best matching past cases.</i> If a code is cited by all of the best matching cases, it is considered "highly likely." If it appears in any of the best matching cases, it is considered possible.</p> |
|---|

Fig. 5. Some Operationalization Selection Heuristics

The program will rely on *operationalization selection heuristics* for deciding which operationalizations are valid and should be suggested for an input case. Some of the heuristics that will be employed are shown in Fig. 5.

## 6. A Hypothetical Example

To illustrate how the computational model will operate, this section outlines a hypothetical run of the model using our example case, 90-4, as input. For the purpose of the example, we assume that Fig. 4's analysis representation of 90-4 is not available to the computational model. Conceptually, the model's objective is to generate output that is comparable to the analysis representation shown in Fig. 4 (with the exception of the board's conclusion).

### Fact Chronology of Case 83-1:

1. Engineer A is <b>employed</b> by Engineer B.	<i>Pre-existing fact</i>
2. Engineer B is <b>hired to provide services</b> for Clients.	<i>Pre-existing fact</i>
3. Engineer A is <b>informed of termination</b> by Engineer B.	<i>After the start of 1</i>
4. Engineer A <b>offers services</b> to Clients. ( <i>Questioned fact 1</i> )	<i>Occurs during 1, 2; After the conclusion of 3</i>
5. Engineer B <b>advertises or solicits engineering business using</b> (Engineer A is a <b>key engineering employee</b> of Engineer B). ( <i>Questioned fact 2</i> )	<i>Occurs during 1, 2; After the conclusion of 3</i>
6. Engineer A is <b>terminated</b> by Engineer B.	<i>Several months after the conclusion of 3; Ends 1</i>
7. Engineer B <b>advertises or solicits engineering business using</b> (Engineer A is a <b>key engineering employee</b> of Engineer B). ( <i>Questioned fact 3</i> )	<i>After the conclusion of 6</i>

### Content Vector of Case 83-1:

(is employed by . 1)  
(is hired to provide services for . 1)  
(is informed of termination by . 1)  
(offers services to . 1)  
(advertises or solicits engineering business using . 2)  
(is terminated by . 1)

### Code Instantiations of Case 83-1:

**CI-1:** Fact 4 is questionable wrt: (I.4., III.7., III.4.a.) primarily because of facts 2, 3  
**CI-2:** Fact 5 is questionable wrt: (II.5.a., III.3.a.) primarily because of fact 3  
**CI-3:** Fact 7 is questionable wrt: (II.5.a., III.3.a.) primarily because of fact 6

### Fact Chronology of Case 84-3:

1. Engineer A <b>found</b> the company "A-Engineering".	<i>Pre-existing fact</i>
2. "A-Engineering" <b>provides engineering services on</b> Projects X.	<i>Immediately after the conclusion of 1</i>
3. Engineer B <i>and</i> Engineer C <b>do not provide engineering services on</b> Projects X.	<i>Occurs concurrently with 2</i>
4. Engineer B <i>and</i> Engineer C <b>are employed by</b> "A-Engineering."	<i>10 years after the start of 1; After the conclusion of 3</i>
5. "A-Engineering" <b>provides engineering services on</b> Projects Y.	<i>After the start of 4</i>
6. Engineer B <i>and</i> Engineer C <b>provide engineering services on</b> Projects Y.	<i>Occurs as part of 5</i>
7. Engineer B <i>and</i> Engineer C <b>buys the company</b> "A-Engineering".	<i>10 years after the start of 4; After the conclusion of 5</i>
8. Engineer B <i>and</i> Engineer C <b>keep the company name</b> "A-Engineering" Name. ( <i>Questioned Fact 1</i> )	<i>Immediately after the conclusion of 7</i>
9. Engineer B <i>and</i> Engineer C <b>advertise or solicit engineering business using</b> Projects X. ( <i>Questioned Fact 2</i> )	<i>After the conclusion of 7</i>
10. Engineer A <b>retires from</b> "A-Engineering".	<i>5 years after the conclusion of 7; Occurs during 8, 9</i>

### Content Vector of Case 84-3:

(found the company . 1)  
(provides engineering services on . 4)  
(does not provide engineering services on . 2)  
(is employed by . 2)  
(buys the company . 2)  
(keeps the company name . 2)  
(advertises or solicits engineering business using . 2)  
(retires from . 1)

### Code Instantiations of Case 84-3:

**CI-4:** Fact 8 is questionable wrt: (I.5., III.3.a.) primarily because of facts 7, 10  
**CI-5:** Fact 9 is questionable wrt: (I.5., III.3.a.) primarily because of facts 3, 7

**Fig. 6.** Two Cases in the NSPE BER Case Base

To set the stage for illustrating case retrieval, let us suppose that the two cases depicted in Fig. 6 -- both of which have been previously decided by the board -- are stored in the case base, along with many other decided cases. Both of these cases, as well as the input case, involve, at least partially, misrepresentation in advertising<sup>1</sup>.

The computational model begins execution by accepting the transcribed fact chronology depicted at the top of Fig. 3. In preparation for the first stage of retrieval, the computational model translates the input fact chronology into a content vector. The resulting content vector for Case 90-4 is depicted at the bottom of Fig. 3.

The first stage of retrieval is intended to identify the case (or cases) with the highest degree of superficial (i.e., non-structural) similarity to the input case. In our example this is achieved by computing the dot product of the content vector of input Case 90-4 against the content vector of all of the decided cases in the case base. The resulting dot products for the cases in Fig. 6 are: (83-1 = 6; 84-3 = 10). The case with the highest score and cases within x% of that case (x being a threshold to be defined empirically) are considered candidates to be passed to the second stage of retrieval.

Unfortunately, this calculation is prone to under-rating possible matching cases because it does not emphasize the focal point of a case: the code instantiations it contains. To correct for this, the computational model also computes the dot product of input Case 90-4 against every code instantiation in the case base. The resulting dot products for the code instantiations of Fig. 6 are: (CI-1 = 0; CI-2 = 1; CI-3 = 1; CI-4 = 0; CI-5 = 1).

The highest rated cases are passed to the structural analysis stage of retrieval, along with those that share a code instantiation. Assuming that the threshold is set to 20%, Case 84-3 and all cases with dot products  $\geq 8$  are collected. Although Case 83-1 has an insufficient rating, it is also passed to the second stage because at least one of its code instantiations (i.e., both CI-2 and CI-3) is non-zero with respect to the input case. A non-zero code instantiation is important; sharing a code is some evidence that a case is relevant. For instance, consider why the dot product of 90-4 and CI-2 is non-zero: the code instantiation matches the "advertising" fact of 90-4, the questioned fact of the case. Although the dot product of the input case together with cases in the case

---

<sup>1</sup> In Case 83-1 Engineer A works for Engineer B and is told that he is going to be terminated. After being told of his termination, but before the termination takes effect, A solicits work from clients of B. During the same period, B distributes a brochure listing A as one of his key employees and continues to distribute the brochure after A is terminated. The case raises 3 ethical questions; the board concludes: (1) A was unethical in soliciting B's clients while still employed, (2) B was ethical in advertising A's employment, as long as he divulges this information during negotiations with potential clients, and (3) B was unethical in advertising A's employment after A has left the firm.

Case 84-3 involves two engineers, B and C, who buy out a firm that was founded by Engineer A. A retires from the firm, but B and C decide to continue to use A's name as the firm name. In addition, B and C promote new business for the firm based on past, successful projects, some of which involved B and C but some of which did not involve B and C. This case raises 2 ethical questions; the board concludes: (1) B and C were ethical in keeping the old firm name and (2) B and C were ethical in referencing the successful projects, as long as they divulge the circumstances to potential clients during negotiations.

base ignores the importance of the ethical question and its relationship to other facts, utilizing code instantiations during retrieval captures that information.

The second stage of retrieval accepts the set of highly rated cases from the first stage and performs the more expensive structural mapping against the input case. Briefly, this involves mapping elements of the input case to elements of the candidate cases and evaluating the correspondences. Thus, for instance, fact primitives and the arguments to the fact primitives will be mapped from the input case to the candidate cases. More notable, and perhaps unique to our domain, fact *chronologies* will be mapped across cases. In effect, chronology is a higher order relation upon the facts. We have not devised a specific algorithm for performing the structural mapping, but as starting points we are considering, for example, the SME matcher of MAC/FAC (Forbus *et al.*, 1994), and the discussion in (Keane *et al.*, 1994).

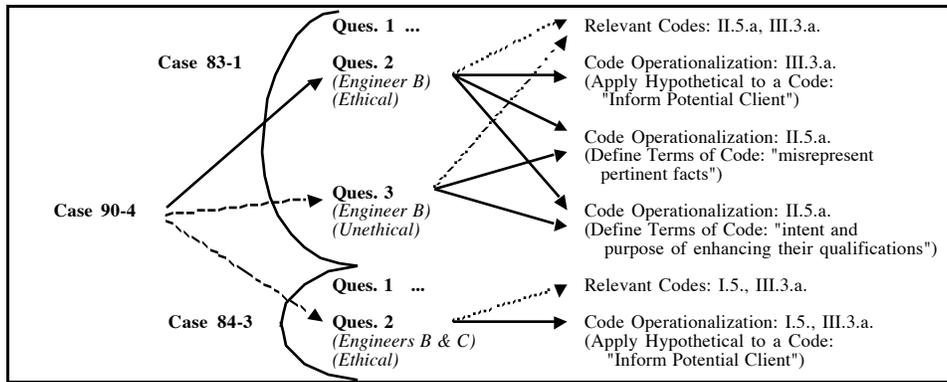


Fig. 7. Mapping of Case 90-4 to the Best Matching Cases after Structural Analysis

Code instantiations also play a role in focusing the structural mapping. Case 83-1, Question 2 (Q2) provides the best structural mapping to 90-4 because facts 4 and 5 of Case 90-4 -- and their chronology -- correspond, respectively, to facts 3 and 5 of 83-1, Q2. The code instantiation CI-2 focuses the program on this mapping. Notice, however, that the mapping is not exact; for instance, "submits resignation" (from 90-4) is the same general type as "is informed of termination by" (from 83-1) (i.e., both are "inform of departure from employment" facts), but is different from the point of view of the initiator. Fig. 7 provides a pictorial view of the hypothetical results of the structure mapping. The thick, solid arrow indicates the best mapping from the input case to candidate cases; the thick, dashed arrows indicate less accurate, yet still relevant, mappings.

Consider how the structural mapper might reject cases that are similar at a superficial level but not at a deeper level. Suppose, for instance, there is a Case X in the case base involving an Engineer Y who advertises past successful projects to gain new business. Such a case would include the fact primitive (or a closely related primitive) "advertises or solicits engineering business using" in its fact chronology and thus has at least superficial similarity to Case 90-4. Suppose, however, that the ethical question raised in Case X deals not with the propriety of the advertising itself but rather with how Y mishandles relations with a potential client who responds to

the advertisement. Case X might be passed to the second stage because of its "advertising" similarity to 90-4, but the code instantiations for Case X, which would not emphasize the advertising fact, would lead to rejection of the case during structural analysis.

**Consider the following in analyzing Case 90-4:**

**Possibly Relevant Codes:**

- Code III.3.a. is highly likely to be applicable to Case 90-4.  
(Reason: III.3.a. is referenced by all of the top-rated cases.)
- Codes I.5. and II.5.a. are possibly applicable to Case 90-4.  
(Reason: I.5. and III.3.a. are referenced by at least one of the top-rated cases.)

**Possible Hypotheticals:**

- A hypothetical fact might make Engineer Z's action ethical. If Engineer Z reveals Engineer X's resignation during negotiations with clients, then Engineer Z may not be unethical in using Engineer X's name while soliciting work.  
(Reason: Case 83-1, question 2 and Case 84-3, question 2 employ a similar hypothetical)

**Possible Code Interpretations:**

- Try interpreting & applying the term "misrepresent pertinent facts" from Code II.5.a.  
(Reason: Case 83-1, question 2 and Case 83-1, question 3 employed a similar interpretation)
- Try interpreting & applying the term "intent and purpose of enhancing their qualifications" from Code II.5.a.  
(Reason: Case 83-1, question 2 and Case 83-1, question 3 employed a similar interpretation)

**Possible Case Citations:**

- Compare Case 90-4 to Case 83-1, question 2.  
Possible Distinctions:
  1. The object of "**advertises or solicits engineering business using**" is a different type of fact. (Reason: Mismatch in Structural Analysis)  
Case 90-4: (Engineer X **is employed by** Firm Y)  
Case 83-1: (Engineer A **is a key engineering employee of** Engineer B)
  2. The cases share the abstract mapping "inform of departure from employment" but have different specific mappings. (Reason: Mismatch in Structural Analysis)  
Case 90-4: (Engineer X **submits resignation to** Firm Y)  
Case 83-1: (Engineer A **is informed of termination by** Engineer B)

**Fig. 8.** Analysis Suggestions made by the Computational Model for Case 90-4

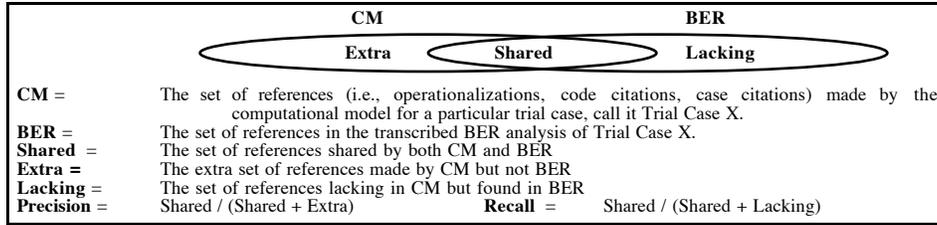
The best matches from the structural mapping are next passed to the Analyzer stage of the computational model. The Analyzer employs the operationalization selection heuristics depicted in Fig. 5 to produce output such as that depicted in Fig. 8. For instance, based on Heuristic H5, it suggests that code III.3.a. is "highly likely to be applicable" because that code is referenced by all of the top-rated cases. Based on Heuristic H1, the Analyzer suggests a hypothetical. The analyses of 83-1, Q2 and 84-3, Q2 employ a similar hypothetical and that hypothetical appears to fit the circumstances of 90-4. The code interpretations of II.5.a. are suggested, again by Heuristic H5, because two of the top-rated cases employ them. Finally, the Analyzer follows Heuristic H3 and draws distinctions between 90-4 and the top-rated case (it would probably do so with other top-rated cases, as well). To achieve this, it examines the mappings produced by the structural mapper.

## 7. A Proposed Experiment to Test the Computational Model

We plan a two-part experiment to test our hypothesis that the computational model will accurately predict facts, principles, and cases that would be regarded as important in the analysis of new cases. As mentioned, the model will be developed using a *foundational set* of 327 NSPE BER cases, decided between 1958 and 1992, which will comprise the program's case base for the experiment. The model will be tested

with a *trial set* of approximately 50 NSPE BER cases, decided between 1993 and 1997. All of the trial cases (and a large majority of the foundational cases) will be transcribed into the engineering ethics language by independent case enterers.

The main quantitative measures for the experiment will be the information retrieval metrics *precision* and *recall*. The application of these metrics, as well as other relevant quantities, is depicted in Fig. 9.



**Fig. 9.** Quantitative Measurements for the Proposed Experiment

We will employ two tactics to test the claim that the computational model makes "accurate" predictions. First, following (Rissland *et al.*, 1996b), we will test how well the model covers the references actually made by the board for the trial cases. All of the trial cases will be run through the computational model, the model's outputs will be compared with the transcribed analyses of the board, and precision and recall will be computed. Of course, there are no standard benchmarks for comparing precision and recall in this domain, and we know of no comparable systems or data sets. We will document the model's trade-off between recall and precision, and hope to show that "Lacking" generally is small. We will also be in a position to conduct ablation studies to isolate which operationalization information is most important.

Second, we will test whether "Extra" (i.e., the references made by the model, but not by the board) are "reasonable" even though the board failed to make them. For several reasons, we anticipate that the model will retrieve, on average, more information than the board would for the same cases, and therefore, that precision scores may be low: (1) The board's analyses often are somewhat sparse. (2) The model will also have the advantage of a complete memory of and consistent access to the past cases, something the board has in theory but not in practice.

To assess the reasonableness of "Extra," experienced ethical reasoners will be asked to grade the references suggested by the model but not by the board. The grading criterion will be whether the extra suggestion is reasonable and relevant to the particular problem.

## 8. Conclusion

Abstract principles and specific fact situations interact in a complex fashion in engineering ethics. In a study of a particular set of engineering ethics cases, we have identified a number of operationalization techniques which help bridge the gap between principles and facts. Our goal is to develop a computational model capable of using operationalizations to accurately predict the facts, principles, and past cases relevant

for analyzing new cases. In this paper, we have presented a preliminary design of such a model and have outlined an experiment to test it. We expect to make a contribution to interpretive CBR by shedding light on the role of principles in decision making, by investigating the connection between abstract rules and concrete facts, and by testing the feasibility of using detailed, factual chronologies to represent cases.

## References

- Allen, J. (1983). Maintaining Knowledge about Temporal Intervals, In the *Communications of the ACM*, 26(11): 832-843.
- Ashley, K. D. and McLaren, B. M. (1995). Reasoning with Reasons in Case-Based Comparisons. In the *Proceedings From the First International Conference on Case-Based Reasoning*, Portugal.
- Branting, L. K. (1994). A Computational Model of Ratio Decidendi. In *Artificial Intelligence and Law 2*: 1-31. Kluwer Academic Publishers. Printed in the Netherlands.
- Branting, L. K. (1991). Building Explanations from Rules and Structured Cases. In *International Journal of Man-Machine Studies*, 34 (6): 797-837.
- Forbus, K. D., Gentner, D. and Law, K. (1994). MAC/FAC: A Model of Similarity-based Retrieval. In *Cognitive Science* 19, 141-205.
- Gentner, D. (1983). Structure-mapping: A Theoretical Framework for Analogy. In *Cognitive Science* 7, 155-170.
- Golding, A. R. (1991). *Pronouncing Names By a Combination of Rule-Based and Case-Based Reasoning*. Ph.D. Dissertation, Stanford University.
- Harris, C. E., Pritchard, M. S., and Rabins, M. J. (1995). *Engineering Ethics: Concepts and Cases*. Wadsworth Publishing Company, Belmont, CA.
- Holyoak, K. J. and Thagard, P. (1995). *Mental Leaps: Analogy in Creative Thought*. The MIT Press, Cambridge, Massachusetts.
- Jonsen A. R. and Toulmin S. (1988). *The Abuse of Casuistry: A History of Moral Reasoning*. University of CA Press, Berkeley.
- Keane, M., Ledgeway, T. and Duff, S. (1994). Constraints on Analogical Mapping: A Comparison of Three Models. In *Cognitive Science* 18, 387-438.
- Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers, Inc., San Mateo, CA.
- McLaren, B. M. and Ashley, K. D. (1995). Case-Based Comparative Evaluation in TRUTH-TELLER. In the *Proceedings From the Seventeenth Annual Conference of the Cognitive Science Society*.
- NSPE (1958-1996). *Opinions of the Board of Ethical Review*, Volumes I through VII and the *NSPE Ethics Reference Guide*. Published by the National Society of Professional Engineers, Alexandria, Virginia.
- Rissland, E. L., Skalak, D. B., and Friedman, M. T. (1996a). BankXX: Supporting Legal Arguments through Heuristic Retrieval. In *Artificial Intelligence and Law 4*: 1-71. Kluwer Academic Publishers. Printed in the Netherlands.
- Rissland, E. L., Skalak, D. B., and Friedman, M. T. (1996b). *Evaluating a Legal Argument Program: The BankXX Experiments*. Technical Report, CMPSCI TR95-30, Department of Computer Science, University of Massachusetts, Amherst, MA.
- Rissland, E. L. and Skalak, D. B. (1991). CABARET: Statutory Interpretation in a Hybrid Architecture. In *International Journal of Man-Machine Studies*, 34 (6): 839-887.
- Strong, C. (1988). Justification in Ethics. In Baruch A. Brody, editor, *Moral Theory and*

*Moral Judgments in Medical Ethics.* Kluwer Academic Publishers, Dordrecht.