Pros and Cons in Ethical Decisions: Introducing Role-based Ethics Harnessed by LLMs, Ontologies, and CBR for Automating Ethical Analysis with Evaluative AI

Anonymous submission

Abstract

Providing ethical guidance in practice is difficult because it often involves handling conflicting principles, individuals struggle to process complex information, and situations involve highly technical content that requires expert knowledge. In response to this challenge, we propose to focus on established professions, such as engineering or medicine, that have developed formal codes of ethics and documented case histories. These resources offer a structured basis for analyzing whether a professional action is ethical. However, even professional ethics committees face significant constraints. Qualified experts are scarce, serving on committees removes them from their primary responsibilities. In addition, technical experts may not have formal training in ethical analysis. While prior automation efforts have attempted to address these issues, they have generally failed to aggregate profession-specific methods of ethical reasoning in combination with structured knowledge and the newly emerging capabilities for natural language interaction provided by large language models. This paper describes ProEthica for Engineering, which integrates four components: ontologies that represent professional codes as RDF knowledge graphs, large language models for concept recognition and language interpretation, case-based reasoning that draws from prior ethical decisions, and evaluative AI to support structured comparison.

Role-Based Ethical Analysis: A Novel Application

This work describes an approach for building a system that supports the analysis of ethical decisions in professional practice. The focus is on professions with established role definitions and codified ethical standards, such as engineering and medicine. These domains have long used formal codes of ethics, committee deliberations, and precedent cases to determine whether professional conduct aligns with role obligations. The system under development operationalizes these methods to assist in consistent and well-grounded ethical evaluations.

Professional roles carry distinctive moral obligations and evaluative criteria that general moral theories do not capture with precision. These obligations derive from the functions and ends of the role, as well as the expectations of the professional community and the public. For example, the responsibilities of an engineer include safeguarding public safety in design and implementation, which shapes the ethical assessment of decisions in that domain (Oakley & Cocking, 2001).

Current practice often involves committees of experienced practitioners who interpret codes of ethics and apply precedent. Such committees face several constraints. Members may emphasize different principles in similar cases, which can produce inconsistent conclusions. Access to comprehensive databases of precedent cases is often limited, which increases the likelihood of overlooking relevant prior decisions (McLaren, 2003, 2006). Recruiting qualified experts to serve on committees can also reduce the availability of those experts for their primary professional work.

Computational ethics provides methods for representing and applying ethical reasoning in ways that can complement the work of professional committees. Segun (2021) characterizes computational ethics as the effort to represent ethical reasoning in a form that can be implemented in artificial systems, distinguishing it from broader conceptions of machine ethics. Within this view, role-based reasoning offers a means of structuring the analysis by anchoring evaluations in the norms, priorities, and constraints defined for a given professional role.

The approach described here integrates structured representations of professional codes, case-based reasoning methods, and tools for identifying relevant precedent. The objective is to support systematic and replicable ethical analysis that aligns with the established evaluative practices of the profession under consideration.

How Ethical Analysis is Done Today and Its Problems

Ethical decision-making in professional practice is often handled by committees of experienced practitioners. In fields such as engineering and healthcare, these committees review case documentation, interpret relevant codes of ethics, and determine whether actions align with professional standards and prior decisions (Beauchamp & Childress, 2019; Segun, 2021). This process is shaped by the concept of role-differentiated morality, which recognizes that the duties associated with a professional role can differ from those of ordinary moral life (Williams, 1981).

Professional committees frequently address situations where multiple prima facie duties apply but point toward different outcomes. Determining which duty should prevail requires expert judgment and sensitivity to the specifics of the case (Beauchamp & Childress, 2019). Case-based and formal reasoning methods have been developed to compute such judgments and resolve principle conflicts (Anderson & Anderson, 2006).

Committee work in professional ethical decision-making, while capable of producing reasoned conclusions, faces several inherent limitations. These include the significant time commitment required from qualified professionals, which can reduce their availability for primary roles, particularly in smaller organizations or specialized fields. Human deliberations can also lead to inconsistent outcomes when members emphasize different principles in similar cases. Ethical codes often present principles without clear methods for resolving conflicts, leaving decisions without explicit justification and potentially undermining social trust. Real-world cases add complexity through interrelated principles and high-dimensional inputs, further complicating consistent application.

These constraints create a need for systems that can organize past decisions, clarify conflicts between principles, and promote consistency. Computational ethics has emerged as one approach to address these needs. Segun (2021) defines computational ethics as the project of computing ethics, aimed at representing ethical reasoning in a computable form and focused on the practical steps required to codify ethics for implementation in artificial systems. This involves the development of formal structures to embed ethical principles into machines.

Systems following this approach offer several benefits. They can encode professional role norms through formal representations such as deontic logic, enabling explicit rule-based governance. Tools like McLaren's SIROCCO use case-based reasoning to retrieve relevant principles and past cases, while GenEth abstracts decision principles from cases where ethicists have reached consensus and can trace these back to their origins for justification (Anderson, Anderson, & Armen, 2006; McLaren, 2003, 2006). Such systems also assist in identifying and resolving principle conflicts by applying methods such as Rawls's reflective equilibrium, revealing the sources of disagreement and supporting resolution. By deriving principles from expert consensus and testing them for consistency, these systems align with established evaluative practices and help ensure that similar cases receive similar treatment.

Integrating computational ethics tools into committee workflows can improve efficiency, enhance consistency, and increase the transparency and justification of decisions, even in complex or novel dilemmas.

Why Automate Ethical Analysis

Ethics committees require substantial time from qualified professionals. Serving on these bodies can limit the availability of experts for their primary responsibilities, especially in smaller organizations or specialized domains (Berkman et al., 2004). This constraint, combined with the difficulty of accessing all relevant precedents, contributes to inconsistency in ethical decisions (Wallach & Allen, 2009).

Computational ethics offers a method for supporting these bodies without replacing human judgment. Segun (2021) defines computational ethics as the representation of ethical reasoning in a form suitable for implementation in artificial systems. Such systems can store and organize professional role norms, retrieve relevant precedent, and assist in identifying and resolving conflicts between principles in a way that aligns with established evaluative practices (McLaren, 2003; McLaren, 2006).

Automated systems can take on the more routine aspects of ethical analysis, allowing specialists to focus on the most complex and unprecedented cases (Coin & Dubljević, 2022). They can systematically search large collections of professional codes, case histories, and related materials to ensure that applicable precedents are identified (Waser, 2014; Myers & Everett, 2025). Automated reasoning can also present the steps that lead to a recommendation, enabling inspection and audit of the reasoning process (Langley, 2019; Wallach & Allen, 2009).

Rather than making decisions independently, evaluative AI can prepare structured comparisons between the current case and relevant precedent, surface potential conflicts between principles, and clarify how different resolutions align with professional standards. This model avoids the problems associated with delegating moral authority to machines while providing the consistency, comprehensiveness, and transparency needed to improve the quality of committee-based ethical analysis (Segun, 2021; McLaren, 2006).

Many Attempts: Previous Approaches to Ethical Analysis

Work on computational support for ethical analysis follows several lines. Case-based reasoning approaches emphasize precedent. McLaren's systems are central examples. SIROCCO was developed for engineering ethics to retrieve prior cases with relevant features and to predict applicable codes for new situations, and Truth-Teller compares pairs of dilemmas to surface ethically salient similarities and differences. These tools were framed as advisory aids, with final judgments left to human experts (McLaren, 2003, 2006).

Principle-learning approaches combine cases with explicit ethical theory. MedEthEx derives decision principles from biomedical cases using inductive logic programming with prima facie duties, and GenEth generalizes from cases where ethicists agree to produce consistent, traceable principles and to expose points of disagreement for refinement (Anderson & Anderson, 2015; Takeshita et al., 2023; Tolmeijer et al., 2021). Formal logic work encodes obligations, permissions, and prohibitions to enable proof-based checking. Athena develops deontic logic–based agents, and LogiKEy uses theorem provers for reasoning about ethical arguments (Bringsjord et al., 2006; Benzmüller et al., 2020). Other strands include utilitarian optimization models such as Jeremy, and hybrid ILP systems for eldercare ethics such as EthEl (Arkin, 2008; Anderson & Anderson, 2006–2008).

Recent systems extend these ideas. HERA offers an agent library that applies principles such as the doctrine of double effect and Pareto optimality (Limarga et al., 2024). Constitutional AI trains models to follow specified constitutional principles when revising outputs (Xu, 2024). MOKA identifies moral events in text to support downstream reasoning, and Delphi trains on large-scale crowd judgments to approximate moral common sense (Zhang et al., 2023; Jiang et al., 2022). Work also appears on LLM-symbolic hybrids for moral narrative planning (Kampelopoulos et al., 2025).

These efforts face recurring obstacles. Many abstract principles remain too general for concrete design and do not specify conflict-resolution procedures, which leaves room for divergent interpretations and "tick-box" compliance (Morley et al., 2021; Mittelstadt, 2019). Practical deployment often suffers from missing operational methods, weak incentives, and organizational gaps between ethicists and engineers (de Laat, 2021; Prem, 2023; Schiff et al., 2021; Bleher & Braun, 2023). Real-time contexts increase computational demands and can force simplifying assumptions that undermine ethical analysis (Arkin, 2008; Stenseke, 2024). Together, these limits motivate an approach that unifies professional codes, precedent management, and principled evaluation within a format that supports human decision-makers.

What Is Needed for Effective Ethical Analysis

Professional ethics benefits from community consensus and institutional authority. Codes are products of that consensus and supply a stable basis for computation in a way that general moral theory does not (AI Ethics as Applied Ethics, 2022). Within defined professional contexts, stakeholders and obligations are specified in advance, which reduces problem scope and makes computational treatment more feasible (Tolmeijer et al., 2021).

Effective analysis should combine abstract rules with concrete cases. Professional practice already develops principles alongside precedents, so systems should support both representation types and test generalizations against particular cases to refine decision guidance (AI Ethics as Applied Ethics, 2022). For functional stance, evaluation rather than autonomy is preferred: analysis informs people who retain authority to decide, a view consistent with the taxonomy in Wallach and Allen (2009).

A modular architecture is required. Four capabilities are central: (1) ontologies that encode professional standards, (2) natural-language processing for concept extraction and case structuring, (3) case-based reasoning to apply the "treat like cases alike" norm, and (4) evaluative methods that expose reasoning steps without issuing determinations (Berreby et al., 2017). Scenario work can further contextualize duties, roles, and consequences, and support systematic assessment (York & Conley, 2020; Berreby et al., 2017; Bonnemains et al., 2018). ProEthica aligns with these requirements by combining RDF knowledge graphs, language models for concept recognition, precedent retrieval, and evaluative reporting that supports human judgment.

Building and Using ProEthica for Engineering Actions

Building and Using ProEthica for Engineering Actions

Use Case: Building a Novel Application (Steps to Build)

System architecture and domain creation. ProEthica supports domain experts in creating specialized ethical analysis within a general framework. A modular plugin design and a Model Context Protocol (MCP) server enable bidirectional interaction between language models and ontological knowledge bases, so extraction and updates proceed under formal constraints. Domain creation requires three components: (i) formal ontologies, (ii) professional guidelines, and (iii) case precedents.

World construction. An engineering ethics world is established to scope analysis to a defined professional community and practice context. The world declares roles, obligations, permissions, prohibitions, actions, states, standards, and evidence links needed for role-based reasoning. Roles and their obligations and constraints serve as primary targets during parsing of guidelines and cases, so duty clauses and role mentions in the NSPE Code attach directly to role instances and are grounded as RDF for traceability (Berreby, Bourgne, & Ganascia, 2017; Kong, Lam, & Cheng, 2020). Case decisions enter as structured items with facts, issues, rules, analysis, and conclusions, then receive annotations for role involvement, obligations at issue, and ethically relevant similarities and differences to support "treat like cases alike" comparison (Ashley & McLaren, 1995; McLaren, 2003).

Knowledge representation and integration. The NSPE Code is encoded as RDF to preserve structure and relations for automated checks. Concept extraction uses an intermediate ontology to normalize terms and map text to formal concepts. Concepts are categorized into principles, obligations, states, actions, and related elements. A knowledge-graph pipeline with expert validation and version control maintains currency and traceability (Berreby, Bourgne, & Ganascia, 2017; Bonnemains et al., 2018).

Case processing and constraints. Cases are processed with language models to extract atomic facts, build vector representations for retrieval, and model event timing. Ontological constraints encode professional duties as logical relations and world elements, supporting consistency checks and inference aligned with engineering practice (McLaren, 2003; Berreby et al., 2017; Bonnemains et al., 2018).

Scenario construction and formalization. Scenarios are constructed from existing cases because they contain the requisite ethical structure. Each scenario tailors duties to roles and contextual constraints (McLaren, 2006). Scenarios are formalized for systematic assessment and "what if" exploration (Berreby et al., 2017; Bonnemains et al., 2018). The NSPE domain build is documented in Appendix A, with a visual walkthrough in Appendix E.

Use Case: Using the Application for Engineering Ethics Analysis

User interaction. A practitioner enters a dilemma in natural language and specifies role context and situational constraints.

System processing. **LLM** extraction identifies *Facts, Issues, Rules, Analysis, Conclusions* (FIRAC). **Ontological** validation checks the scenario against NSPE obligations and constraints. Scenario analysis clarifies action–situation relationships and consequences (Hobbs & Moore; York & Conley, 2020). Actual causality assessment supports role-based accountability (Sarmiento et al., 2023). **CBR** retrieval draws on NSPE Board of Ethical Review cases, with multi-metric ranking to prioritize relevant precedents. Reasoning follows professional frameworks rather than abstract theory, with bidirectional validation to keep both interpretation and recommendations aligned with codes and precedent (McLaren, 2003; Berreby et al., 2017).

Outputs and decision support. Transparency documentation records the full reasoning trail. Confidence scoring indicates alignment with standards and precedents. An expert validation workflow flags scenarios that require consultation. Integration with professional workflow systems supports documentation and compliance, and decision records preserve accountability. **Evaluative AI** presents analysis so a practitioner can make an informed decision (Hobbs & Moore; York & Conley, 2020; Berreby et al., 2017).

Example and validation. NSPE BER Case 22-7 (AI use in engineering practice) serves as a demonstration. Scenarios are generated from case facts to test capability. Analysis is compared with the Board's determination under a leave-one-out protocol, and the complete pipeline is documented in Appendix D (McLaren, 2006; Berreby et al., 2017; Bonnemains et al., 2018).

Use Case: Using the Application for Engineering Ethics Analysis

Practitioner input. A practitioner enters a dilemma in natural language and identifies the operative role context and constraints. The scenario is then instantiated in the world with roles, actions, states, and timing.

Structured extraction and validation. Language processing extracts candidate facts and issues, then aligns them with role obligations and constraints via the ontology for consistency checks (Berreby et al., 2017; McLaren, 2003). The workflow records each alignment step for inspection, which supports audit and adoption in professional settings (Wallach & Allen, 2009).

Precedent retrieval and comparison. The case-based component retrieves relevant precedents keyed to role, obligation, and salient factual patterns. Comparative evaluation highlights ethically relevant similarities and differences and surfaces cited principles and reasons that mattered in earlier decisions (Ashley & McLaren, 1995; McLaren, 2003).

Scenario-based evaluation. Alternative actions are evaluated within the instantiated scenario. The analysis traces how each action interacts with role duties, constraints, and likely outcomes, drawing on scenario methods to explore consequences and expose conflicts (Hobbs & Moore, 2005; York & Conley, 2020; Berreby et al., 2017).

Evaluative presentation, not delegation. Output emphasizes a clear reasoning trail: roles and obligations implicated, code provisions and cases consulted, points of conflict, and candidate resolutions supported by precedent. The approach supports human judgment rather than replacing it, consistent with evaluative uses of AI in ethics (Wallach & Allen, 2009).

Example Walk-Through: NSPE BER "Use of Artificial Intelligence in Engineering Practice"

1. **Case selection and scenario formation.** The NSPE Board of Ethical Review case "Use of Artificial Intelligence in Engineering Practice" serves as the seed example. A scenario is formed from the public case record. The source case is withheld from the precedent pool during retrieval to avoid leakage. Only cases cited within the source case may remain available, with disclosure during analysis.
2. **Deconstruction into timeline, events, and decisions.** Facts are ordered into an event timeline. Actions and decision points are enumerated with available options. Scenario-based decision analysis frames alternatives and responsibilities before evaluation, drawing on established scenario methods (Hobbs, 2005; York & Conley, 2020; Berreby, Bourgne, & Ganascia, 2017).
3. **Role and duty extraction.** Roles are identified as first-class elements (for example, engineer of responsible charge, reviewer, supervisor, client, tool supplier). Obligations from the NSPE Code are attached to roles during parsing (for example, Code I.1 on public safety and III.2.b on signing or sealing plans that are not safe), following representations that link principles to concrete facts for operational use (McLaren, 2003; Ashley & McLaren, 1995).
4. **Precedent retrieval and structured comparison.** Precedents are retrieved and ranked by similarity once the source case is excluded. Comparison emphasizes ethically relevant similarities and differences, with reasons grounded in duties and practical considerations, consistent with case-based methods in practical ethics (Ashley & McLaren, 1995; McLaren, 2003).
5. **Option evaluation against codes and conflicts.** Each option is checked for alignment with attached duties. Where multiple duties apply, conflicts are examined using extensional links between code provisions and concrete facts to reduce the gap between abstract rules and practice (McLaren, 2003). Code hierarchy and related provisions guide interpretation during clashes.
6. **Outputs and review.** The analysis report includes: (a) a scenario-indexed rationale that ties facts to duties and precedents; (b) citations to code passages and past decisions used; (c) an alignment summary with the pattern of

reasoning by the Board for the chosen case, using a leave-one-out protocol for fairness. Scenario use also supports education and audit by exposing how conclusions follow from role duties within a structured case narrative (Ashley & McLaren, 1995; McLaren, 2003; York & Conley, 2020).

Rationale for structure. Scenario analysis surfaces options and responsibilities before applying codes (Hobbs, 2005; Berreby et al., 2017). Case-based comparison supports the "treat like cases alike" norm with explicit reasons for and against actions (Ashley & McLaren, 1995). To reduce the gap between abstract rules and practice, extensional links between code clauses and concrete facts are employed, while final moral judgment remains with human decision-makers in an evaluative role (McLaren, 2003; Wallach & Allen, 2009).

Addressing Challenges – How Each AI Method Addresses Engineering Challenges

Ontologies – Formal Knowledge Representation

Professional knowledge structuring. Professional ethics analysis relies on a formal organization of codes, precedents, roles, and context. Domain ontologies supply the structure needed for role-based duties and permissions and for comparison against precedent and codes (Ashley & McLaren, 1995; McLaren, 2003). Cross-domain analysis is supported through semantic mapping that aligns related notions such as public safety, patient welfare, and access to justice. The engineering world draws on ISO terminology, and medical deployments may reference UMLS for compatibility.

Concept hierarchy and organization. An intermediate ontology links natural language documents to formal concepts (Berreby, Bourgne, & Ganascia, 2017). Professional knowledge maps to $W = (R, P, O, S, Rs, A, E, C, T)$: Roles, Principles, Obligations, States, Relations, Actions, Evidence, Context, and Time. Description logics support classification of principles, obligations, rights, and standards.

Expert validation and QA. Language-model extraction is human-mediated. Specialists review candidate terms and can reassign or reject associations to maintain professional accuracy.

Technical implementation and integration. Semantic RDF extraction yields knowledge graphs with URI links back to source passages (Appendix B.1). Multi-level inheritance runs from foundational ontologies through intermediate layers to domain vocabularies. Version control records evolution with new guidance and decisions. Dynamic linking provides current definitions during analysis. Ontological constraints limit inconsistent outputs and address training-data issues by enforcing validation against authoritative knowledge.

LLMs – Natural Language Processing for Case Analysis and Concept Extraction

Natural language understanding for professional contexts. Language technologies support ontology retrieval, parsing, association identification, and structured record creation. Extraction identifies stakeholders, actions, consequences, obligations, and context, with ontological checks to maintain conformance. Confidence scoring aids triage for expert review.

Specialized agent architecture. A plugin architecture supports domain-specific extraction strategies. Configurable adapters perform format checks and pattern recognition. Task-focused agents coordinate work across guideline analysis, precedent processing, and concept extraction (Berreby et al., 2017; Cointe et al., 2016).

Temporal and causal reasoning. Temporal models adopt Common Core TimeOntology patterns. Model Context Protocol (MCP) enables tool integrations (Appendix B.2). OWL description logic queries support discovery over relations such as *isA*, *contains*, *partOf*, and *hasQuality*.

Agent-based analysis and integration. Event Calculus representations model causality and decision steps. Expert validation maintains oversight of associations and reasoning. The complete architecture demonstrates MCP integration between agents and ontological data (Appendix B.5).

CBR (Case-Based Reasoning) – Learning from Similar Ethical Cases

Precedent-based professional reasoning. Professional practice follows the norm to treat like cases alike. The NSPE Board of Ethical Review corpus provides authoritative precedents. Professional methods generalize from judgments in particular cases and then test those generalizations to refine principles; the dual use of abstract rules and concrete examples reduces the gap between theory and practice (AI Ethics as Applied Ethics, 2022).

Case processing and similarity assessment. The pipeline confines precedent access to authoritative repositories. Cases link through ontological relations and temporal progression. Section-level deconstruction enables matching between specific questions in new scenarios and established precedents. A multi-metric similarity method combines vector search with pattern analysis; scoring weighs embeddings, language-model analysis, heuristics, and ontology agreement (Appendix B.6).

Advanced case analysis techniques. Retrieval-augmented methods incorporate relevant precedents into analysis (Zhang et al., 2023; Limarga et al., 2024; Awad et al., 2018). Local similarity focuses on factual patterns and duties; global similarity addresses overall context and reasoning style. Case adaptation applies insights to novel scenarios, and multiple-case synthesis supports scenarios with several decision points.

Learning and validation mechanisms. Continuous learning draws on outcome analysis and expert feedback. A historical database maintains complete Board decisions with structured indexing. Scenario work supports learning from demonstrations and human feedback about ethical utility functions (Abel et al., 2016). Prior systems inform this design: MedEthEx abstracts principles from biomedical cases (Cervantes et al., 2020; Anderson et al., 2005) and GenEth infers principles within dilemma types (Anderson & Anderson, 2018). Scenario interfaces surface relevant precedents for side-by-side comparison (Anderson et al., 2006; Berreby et al., 2017).

Validation methodology. Source cases used to generate scenarios are excluded during retrieval. Precedents cited within a source remain available with disclosure. Leave-one-out protocols support systematic evaluation, with alignment measured against Board determinations.

Evaluative AI

Human-centered ethical analysis. Evaluative AI supports human judgment rather than replacing it. Analysis supplies data, codes, and precedents for informed decision-making. The focus stays on examination of options and implications, consistent with functional levels of machine ethics that prioritize analysis over autonomous moral choice (Wallach & Allen, 2009).

Validation and quality assurance. Bidirectional validation checks both scenario interpretation and recommendations against professional codes. Confidence scoring reports alignment with standards and similarity to precedent. A structured rationale traces movement from facts through obligations to recommendations. Linkages among guidelines, cases, and scenarios remain available for audit. Uncertainty indicators flag areas with heavier reliance on predictive components.

Professional integration and compliance. Procedural regularity enables publicly defensible resolutions adaptable across contexts (Morley et al., 2021). Scenario work tests and refines principle application in role settings (Bonnemains et al., 2018; Waser, 2014). This approach bridges theoretical ethics and professional application by providing systematic methods for applying abstract principles to concrete situations (AI Ethics as Applied Ethics, 2022).

Benefits, Limitations and Future Deployment

Benefits

This approach replaces ad hoc reasoning that relies on abstract moral theory with systematic analysis grounded in established professional codes, documented procedures, and precedent. Conclusions can evolve through continued reference to prior determinations, similar to case law development in legal contexts. Automated precedent retrieval and similarity matching reduce the effort needed for comprehensive review. The workflow can be used as a structured learning tool for case analysis, so students and practitioners study how principles apply to concrete scenarios, examine alternative reasoning paths, and review the relation between codes and decisions. Formal representation supports consistent application of code provisions across analyses, while still accommodating context-sensitive professional judgment. Empirical validation uses NSPE Board of Ethical Review cases as scenario sources and measures alignment with Board conclusions across factual analysis, rule application, precedent use, and reasoning structure. Accuracy of precedent citation is checked against Board references, and quantitative metrics assess correspondence between reasoning pathways and Board analyses. The alignment scoring methodology follows a FACTSCORE-style multi-component weighting with reported error rates below 2% (Min et al., 2023).

Limitations

The present implementation focuses on NSPE engineering ethics rather than broader domains. The application provides analysis rather than autonomous determination, so final judgment rests with qualified professionals. Computational complexity can constrain use in urgent, real-time contexts. Codes and precedents change over time, which requires ongoing ontology maintenance and database curation. Cultural and jurisdictional variation is only partly addressed by state-specific practice within the United States; international use would require further adaptation. Reliance on historical precedents may also perpetuate limitations or biases that appear in earlier determinations.

Future Deployment

Initial use will occur in controlled settings with students studying computational ethics and with a small board of lawyers and other professionals convened for testing and review. Broader integrations remain under consideration. Possible directions include curricular use beyond pilot courses, extensions to medical, legal, and financial domains through a generalized plugin architecture with domain adapters, cross-domain semantic mapping for comparative analysis, and modes for time-sensitive decision support. International adaptation with region-specific ontologies and case bases is being explored conceptually. Additional research directions under consideration include expansion of the ethical text corpus, AI-assisted creative scenario generation for emerging challenges, expert-panel simulation for evaluation of generated scenarios, and comparative studies of agreement between ProEthica analyses and panel determinations. Longer-term exploration may consider support for evolving professional roles, alignment with certification or continuing education contexts, and contributions to standards development through systematic analysis of trends in precedent and reasoning.

Appendix

Acknowledgments

[REDACTED]

References