POLITECNICO DI TORINO

MASTER's Degree in DATA SCIENCE AND ENGINEERING

MASTER's Degree Thesis

Integrating DMN and LLM for Automated Student Feedback and Support

Supervisors

Prof. Paolo GARZA
Politecnico di Torino

Prof. Amin JALALI
Stockholm University

Candidate

Shaghayegh ABEDI

OCTOBER 2025

Abstract

Timely and constructive feedback is essential for supporting student learning, yet providing it at scale remains a challenge for educators. Large Language Models (LLMs) offer new opportunities to automate parts of this process, but their effectiveness depends heavily on how decision logic is formulated and maintained. When such logic is embedded directly in prompts, it can be difficult for instructors to update or adapt it over time. This thesis presents a framework that integrates Decision Model and Notation (DMN) with LLM prompting to make feedback generation more modular, transparent, and easy to refine. The approach decomposes complex evaluation rules into smaller, structured decision steps, which are then used to guide the LLM's reasoning. The framework was applied in a graduate-level course, using student assignments and DMN models representing feedback criteria as inputs. The resulting feedback was evaluated against a Chain-of-Thought (CoT) baseline and assessed for perceived usefulness through a Technology Acceptance Model (TAM)–based survey. Results show that the DMN-guided approach improves both accuracy and consistency compared to CoT prompting, while also receiving positive responses from students. These findings suggest that combining LLMs with structured decision modeling can enhance the quality, transparency, and adaptability of automated feedback in educational settings.

ACKNOWLEDGMENTS

I am deeply grateful to my supervisors, Prof. Paolo Garza of Politecnico di Torino and Prof. Amin Jalali of Stockholm University, for their invaluable guidance, encouragement, and thoughtful feedback throughout the course of this work. Their expertise has not only shaped the direction of this dissertation but has also greatly enriched my academic growth.

I would also like to express my sincere appreciation to my family for their continuous encouragement and care. Their faith in me has given me the confidence to persevere and the strength to overcome challenges. Their understanding and support have lightened the heaviest moments of this journey and made the path ahead seem possible even when it felt difficult.

My heartfelt thanks go to my partner, whose patience, love, and unwavering belief in me have been a constant source of motivation.

Lastly, I am thankful to my friends for their kindness, good humour, and steady moral support. Sharing this experience with them has brought joy and balance to an otherwise demanding process.

This dissertation is as much a reflection of their support as it is of my own effort, and I remain deeply indebted to each of them.

Shaghayegh Abedi

Table of Contents

List of Figures

List of Tables

Acronyms

DMN	Decision Model and Notation.
OMG	Object Management Group.
UML	Unified Modeling Language.
BPMN	Business Process Model and Notation.
DRD	Decision Requirements Diagram.
FEEL	Friendly Enough Expression Language.
LLM	Large Language Model.
LLMs	Large Language Models.
GPT	Generative Pre-trained Transformer.
GPT-4o	OpenAI GPT-4o model.
LLaMA	Large Language Model Meta AI.
RAG	Retrieval-Augmented Generation.
TAM	Technology Acceptance Model.
BPM	Business Process Management.
BRMS	Business Rule Management Systems.
AES	Automated Essay Scoring.
ITS	Intelligent Tutoring Systems.
CAI	Computer-Assisted Instruction.

NLP	Natural Language Processing.
SUS	System Usability Scale.
ROUGE	Recall-Oriented Understudy for Gisting Evaluation.
BERTScore	BERT-based Semantic Similarity Metric.
CoT	Chain-of-Thought.
API	Application Programming Interface.
HTTPS	Hypertext Transfer Protocol Secure.
JSON	JavaScript Object Notation.
PII	Personally Identifiable Information.
BPR	Business Process Redesign.
WoPeD	Workflow Petri Net Designer (Petri net modeling tool).
XOR	Exclusive OR.
LMS	Learning Management System.
PU	Perceived Usefulness.
PEU	Perceived Ease of Use.
TP	True Positive.
FP	False Positive.
TN	True Negative.
FN	False Negative.

Chapter 1

Introduction

1.1 Background and Motivation

In many domains where decision-making needs to be both consistent and explainable, the ability to clearly separate business rules from the processes that use them has proven invaluable. The Object Management Group (OMG) established the Decision Model and Notation (DMN), which was designed to address exactly this need [1]. By offering a structured and interpretable way to formalize decision logic, DMN allows organizations to document and automate operational rules without burying them in complex procedural code. Unlike traditional approaches, where rules are hidden inside software systems, DMN makes them visible and accessible to both domain experts and technical specialists. It achieves this through two complementary representations: graphical diagrams and tabular formats [2, 3]. Together, these tools connect the two informal policy descriptions and fully executable logic, ensuring that changes to decision-making criteria can be made with minimal disruption. A central element of DMN is the Decision Requirements Diagram (DRD) [1], which maps how different pieces of information and intermediate decisions interact. To illustrate, consider a basic loan approval scenario, as shown in Figure 1.1. Here, two input data elements—*Salary* and *Credit Score*—feed into a decision node labeled *Loan Approval Rule*. This rule determines whether an applicant is approved, sent for manual review, or declined. The result of this evaluation is passed on to another element, *Loan Approval Result*, which generates the appropriate message for the applicant.

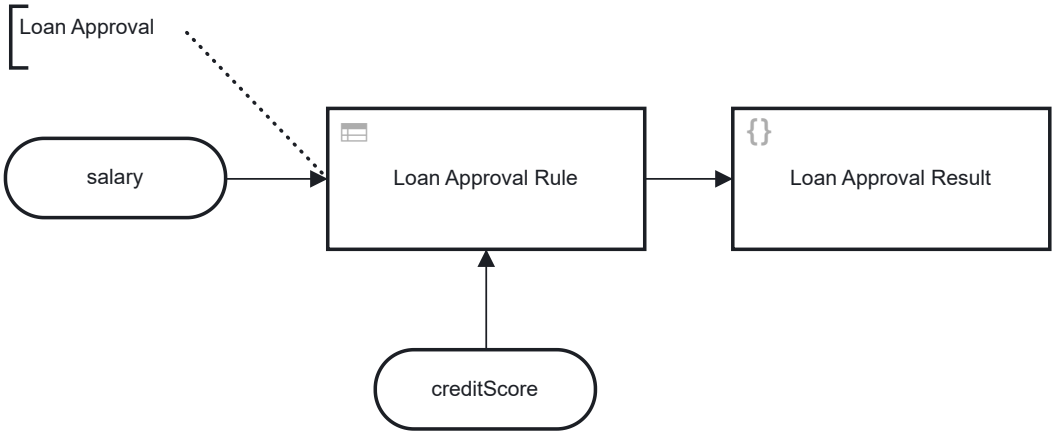

Figure 1.1: A sample decision requirements diagram.

Within the *Loan Approval Rule*, the underlying decision logic is often defined using a decision table,Figure 1.2.

Loan Approval Rule | **Hit policy:** Unique

	When salary string	And creditScore string	Then Is Eligible for Loan? string
1	> 50000	>= 700	approve
2	> 50000	[650..700[	manual_review
3	-	-	decline

Figure 1.2: A decision table example

In this table, each row indicates a distinct rule, while the columns specify input conditions and the corresponding outcome. For example, one rule might state that applicants earning more than 50,000 with a credit score above 700 are approved, whereas a slightly lower credit score (between 650 and 700) triggers a manual review. Any case that fails to meet these criteria defaults to a decline. Decision tables are evaluated sequentially from top to bottom, with wildcard entries ("–") providing a fallback when no earlier rule applies. Once the decision has been made, a literal expressionFigure 1.3—written in the Friendly Enough Expression Language (FEEL)—can be used to map the result to a human-readable message. This enables the system to return clear and context-specific communication, such as congratulating approved applicants or offering declined applicants a polite explanation.

Loan Approval Result

```
approave_message = "We're pleased to inform you that your loan application has been
approved. Congratulations!".
manual_review_message = "We'd like to inform you that your application has been
processed and requires a manual review. We will update you as soon as the review is
complete.".
decline_message = "We regret to inform you that your loan application has not been
approved due to either insufficient income or credit score. If you have any questions
or would like further clarification, please feel free to contact us."
```

Figure 1.3: A case of literal expression.

While this example is deliberately simple, it reflects why DMN is valued in practice: it produces decision logic that is explicit, easy to maintain, and adaptable to change [1]. Although DMN is well established in industries like finance, insurance, and regulatory compliance, its role in orchestrating AI-driven systems remains underexplored. In this thesis, DMN is applied in a novel way—as a guiding framework for Large Language Models (LLMs) in educational feedback generation—combining the precision of formal decision modeling with the flexibility of modern AI tools.

1.2 Problem Statement

The central challenge this thesis addresses is the lack of transparency, adaptability, and maintainability in current LLM-based feedback generation systems. While large language models such as GPT, Claude, and LLaMA have shown significant potential in producing context-aware, domain-specific feedback, their effectiveness heavily depends on the quality of the prompts used to guide them [4]. In most existing approaches, including chain-of-thought prompting, all decision logic is embedded directly within long, unstructured prompt text [5, 6]. Although this can produce good results under controlled conditions, it introduces several practical issues when deployed in real educational settings:

1. **Opacity** – The reasoning process driving the LLM's evaluations is hidden inside text instructions, making it difficult for educators to verify correctness, identify errors, or audit the logic.

2. **High Maintenance Costs** – Any change to evaluation criteria often requires rewriting the entire prompt from scratch, increasing the risk of inconsistencies.

3. **Limited Reusability** – Prompts designed for one course or assignment type are rarely transferable without extensive modification.

4. **Pedagogical Misalignment** – Without an explicit, structured representation of the decision process, there is no assurance that generated feedback consistently aligns with the intended learning objectives.

These limitations hinder scalability and long-term sustainability in real classrooms. For instance, in a graduate-level course with more than a hundred students, each

assignment may need to be graded against a detailed evaluation criterion with many conditional rules. When rules are hidden inside long, free-text prompts, even a small change like shifting a grading threshold can turn into a major task. This not only slows down the feedback process but also makes mistakes more likely.

Moreover, prompt engineering is often a trial-and-error activity that demands technical expertise, which many educators may lack. As the decision logic changes over time, the prompts need to be revised as well, creating extra work and complexity. These challenges extend beyond accuracy—they affect transparency, collaboration, and the ability to maintain consistent standards over time.

To address these issues, this thesis proposes integrating Decision Model and Notation (DMN)—a standardized graphical notation for defining decision logic—into the LLM prompting process [7, 1]. DMN enables decision pathways to be broken into smaller, structured, and interpretable components that can be easily updated without rewriting the entire prompt. This modularization not only improves maintainability but also makes the decision process transparent to educators, ensuring pedagogical alignment. In the proposed framework, DMN rules guide the LLM's reasoning, while retrieval-augmented generation (RAG) can optionally supply contextual knowledge to enhance accuracy.

The approach was tested in a graduate-level course, where student assignments and DMN-based feedback criteria were used to generate automated responses. Evaluations showed that DMN-guided prompting outperformed traditional chain-of-thought prompting in both accuracy and consistency [8], while students reported high levels of perceived usefulness based on a Technology Acceptance Model (TAM)–inspired survey [9]. These findings demonstrate that combining LLMs with structured decision modeling offers a promising path toward scalable, transparent, and pedagogically aligned feedback generation.

1.3 Objectives and Research Questions

The main goal of this thesis is to create, test, and assess a framework that brings together Decision Model and Notation (DMN) and Large Language Models (LLMs). The intention is to make automated feedback in education more accurate, transparent, and flexible.

The research is motivated by the need to address persistent challenges in existing LLM-based feedback systems—particularly the lack of transparency in decision-making, the difficulty of updating evaluation rules, and the risk for prompts to become locked into one specific use case.. By separating decision logic from prompt text and expressing it in a structured, modular form, the proposed approach aims to make automated feedback systems easier to maintain, adapt, and reuse across different courses or assignments.

The specific objectives of this work are to:

- Develop a DMN-guided prompting framework that translates formalized de-

cision logic into structured inputs for LLMs, allowing for modular updates without full prompt rewriting.

- Investigate the effectiveness of this approach compared to traditional methods such as chain-of-thought (CoT) prompting, focusing on accuracy, consistency, and alignment with pedagogical goals.
- Evaluate the usability and perceived usefulness of DMN-guided feedback among students and instructors, drawing on established models such as the Technology Acceptance Model (TAM).
- Explore the potential for scalability and adaptability by assessing how well the framework supports changes to evaluation criteria, rubrics, and course contexts.

Research Questions:

1. How can Decision Model and Notation (DMN) be effectively integrated with Large Language Models to produce modular, transparent, and pedagogically aligned feedback?
2. In what ways does DMN-guided prompting compare to chain-of-thought prompting in terms of feedback accuracy, consistency, and clarity?
3. How do students and instructors perceive the usefulness, clarity, and reliability of feedback generated through DMN-guided prompting?
4. To what extent can the proposed approach support adaptation to evolving course requirements without substantial re-engineering?

By addressing these questions, the thesis seeks to contribute both a practical tool for educators and broader insights into the integration of structured decision modeling with generative AI in the context of higher education.

1.4 Contributions of the Thesis

The following contributions are made by this thesis:

Technical:

- Introduces a new prompting approach where Decision Model and Notation (DMN) is used to structure, separate, and simplify the decision logic guiding Large Language Models, making it easier to update and understand.
- Provides a working implementation that connects DMN-based decision rules with LLM-driven feedback generation in an educational context.

Methodological:

- Proposes an evaluation strategy that blends quantitative performance measures (such as precision, recall, F1-score, and accuracy) with qualitative insights gathered through surveys grounded in the Technology Acceptance Model (TAM).

- Presents a side-by-side performance comparison between the proposed DMN-guided approach and the conventional chain-of-thought prompting technique.

Practical:

- Validates the framework through its use in a real graduate-level course, showing both effectiveness and positive reception from students.
- Offers practical guidance for educators interested in AI-supported feedback systems that emphasize clarity, transparency, and adaptability over time.

1.5 Structure of the Thesis

The remainder of this thesis is organized as follows:

- **Chapter 2** presents a comprehensive literature review on LLM prompting strategies, DMN, and automated feedback systems.
- **Chapter 3** details the proposed DMN-guided prompting methodology, including the architecture, parsing process, and prompt design.
- **Chapter 4** presents the case study in detail, including the educational context, deployment of the framework, technical implementation, evaluation results, rule-specific insights, and student perceptions of usefulness.
- **Chapter 5** presents the evaluation results, both quantitative and qualitative.
- **Chapter 6** outlines possible directions for future research.
- **Chapter 7** concludes the thesis with a summary of findings.

Chapter 2

Literature Review

2.1 Large Language Models

2.1.1 Overview

Large language models are essentially powerful neural networks trained on enormous amounts of text. By learning patterns in language, they can predict what word is likely to come next, allowing them to produce text that reads naturally and fits the context. Most of today's leading systems are built on the Transformer architecture [10], which uses a method called multi-head self-attention to understand relationships between words—even when those words are far apart in a sentence.

Modern examples include OpenAI's GPT-4o, Anthropic's Claude, Google's Gemini 1.5 Pro, and Meta's LLaMA models. These tools have pushed the boundaries of what AI can do with language. Without being trained for specific jobs, they can handle everything from reasoning through complex problems to summarizing articles, translating languages, and even writing computer code.

2.1.2 Key Capabilities

- **Contextual Understanding:** Some advanced language models can hold onto and understand a large amount of context—whether that's an extended back-and-forth conversation or a lengthy piece of writing [11].
- **Reasoning and Inference:** They're capable of thinking through problems in a sequential way, often called "chain-of-thought" reasoning, which helps them reach more reliable conclusions.
- **Domain Adaptation:** When you give them prompts that are specific to a certain field or topic, they can shift their responses to better fit that subject.
- **Multi-Modal Processing:** A few of these models can even handle different types of input at the same time—text, images, audio, and video—working across them in a coordinated way. GPT-4o, for example, can combine all of these formats smoothly.

2.1.3 Limitations

Despite their power, LLMs suffer from:

- **Hallucinations:** these models may confidently give an answer that sounds right but is wrong [12].
- **Prompt Sensitivity:** Even little changes to the wording of an instruction or inquiry might result in entirely different responses from the model [13].
- **Opaque Reasoning:** Lack of transparency in how decisions are made internally [14].

These limitations motivate structured prompting approaches—such as DMN-guided prompting—that add a formal reasoning layer on top of the model.

2.2 Prompt Engineering

Prompt engineering is the science of designing input instructions in a way that steers how large language models respond. Since these models produce answers based on patterns they've learned from massive amounts of text, the phrasing, structure, and context of a prompt can heavily influence the outcome. This makes prompt engineering an important skill for improving the accuracy, clarity, and usefulness of the model's responses [15, 16]. It's an expanding field of study, with many different approaches under investigation.

2.2.1 Zero-Shot Prompting

In a zero-shot prompting setup, the language model receives only a description of the task or a direct question, without being shown any sample solutions beforehand. Its response is based solely on prior training and whatever guidance is included in the prompt at the moment [17].

Example 2.1:

```
Prompt: "Translate the following sentence into French:
‘The meeting starts at 3 PM.’"
```

Since no illustrations or examples are supplied, the model has to deduce the correct output format and perform the translation entirely from its existing knowledge.

2.2.1.1 Advantages

- **Simplicity:** With zero-shot prompting, there's almost no need for elaborate prompt design. There is no need to prepare sample inputs and outputs; the setup is straightforward, making it a good choice for quick trials or initial testing phases.

- **Low token cost:** Since the prompt is limited to the task instructions and the actual query, it stays brief. A shorter prompt means fewer tokens are processed, which can reduce operational costs and improve processing speed.
- **Quick adaptability:** You can alter or rewrite the task instructions without having to adjust a set of predefined examples. This flexibility allows you to easily test new wording or adapt the same setup to a slightly different use case.
- **Broad applicability:** This approach can be applied to many different kinds of problems, as long as the instructions are written enough for the model to follow without confusion.

2.2.1.2 Limitations

- **inconsistent output quality:** The results can vary noticeably between runs, especially on tasks that require multi-step reasoning or have more than one acceptable solution. Without example cases to guide the process, the model's responses may lack uniformity.
- **Risk of topic mismatch:** When dealing with niche subject matter or industry-specific jargon not well represented in the model's training data, performance can decline. In these situations, the model depends solely on what it has learned previously, with no contextual hints to bridge the gap.
- **Lack of formatting cues:** In the absence of modelled examples, the structure of the output may shift unpredictably. This can pose challenges for assignments that demand a fixed layout, such as code snippets, JSON responses, or standardized forms.
- **No built-in corrective mechanism:** Because there are no reference outputs in the prompt, the model cannot adjust its answers to better align with a specific tone, structure, or content style. This can increase the likelihood of misunderstandings or deviations from the intended format.

Best suited for: Zero-shot prompting works well for straightforward tasks such as basic categorization, direct translations, simple fact-based questions, or short-form content creation. It works very well when the task instructions are unambiguous and the subject matter is already well represented in the model's prior knowledge. In these cases, it offers a fast and efficient way to generate results without the overhead of designing detailed example sets.

2.2.2 Few-Shot Prompting

In few-shot prompting, the request to the model contains a handful of example input–output pairs before the actual question or task. These short demonstrations act as guidance, showing the model the pattern or style it should follow when generating its own response [18].

Example 2.2:

Translate English to French:

```
Translate English to French:

English: The meeting starts at 3 PM.
French: La réunion commence à 15 heures.

English: She is reading a book.
French: Elle lit un livre.

English: I will call you tomorrow.
French:
```

2.2.2.1 Advantages

- **Improved accuracy:** This approach often produces more reliable results, particularly for work that must follow a specific structure or be adapted to a particular subject area.
- **Consistent style and structure:** The inclusion of examples helps the model maintain the same tone, format, and language style throughout its output.
- **No retraining needed:** It works directly with the existing model without the need for additional fine-tuning.

2.2.2.2 Limitations

- **Longer prompt length:** Supplying examples increases the total number of tokens, which can raise processing costs and slow response time.
- **Example-dependent performance:** If the chosen examples are unclear, misleading, or unrepresentative, they can negatively influence the model's responses.
- **Context size constraints:** Adding too many examples can consume the model's available context window, leaving less room for the actual task or relevant background information.

Best suited for:

Tasks such as classification, generating structured outputs, mimicking a particular writing style, or other situations where example data is available but full-scale model fine-tuning is impractical.

2.2.3 Chain-of-Thought (CoT) Prompting

Chain-of-Thought prompting [8] is a prompting method that instructs a language model to lay out its sequential logic procedure before presenting a conclusion. Instead of jumping directly to the answer, the model walks through its thought process in smaller, logical segments. This approach takes inspiration from how people often solve problems—by breaking them into manageable parts and working through each one systematically.

The value of this technique lies in making the reasoning explicit. It helps the model maintain logical flow, reduces the likelihood of skipping steps, and provides a record of how the conclusion was reached. This "visible" thought process can be reviewed by humans, which adds a layer of transparency and makes it easier to verify or debug the model's output.

Example 2.3:

```
Prompt: "Solve: A farmer has 17 sheep, and all but 9 run away.
How many are left? Think step by step."

Model response:
"If all but 9 run away, that means 9 remain. Therefore, the answer is 9."
```

2.2.3.1 Advantages

- **Better accuracy for complex problems:** By handling a problem in stages, the model can address each part carefully, often improving results for multi-step calculations or reasoning tasks.
- **Easier troubleshooting:** The intermediate explanations act as a roadmap, helping pinpoint exactly where an error occurs.
- **Greater clarity:** The explicit breakdown allows end-users to see how the answer was reached, which can be important in fields like teaching, compliance, or research where reasoning is as important as the outcome.

2.2.3.2 Limitations

- **Longer outputs:** Explaining every step increases the length of responses, which can make them slower to read and more costly to generate.
- **Prompt dependency:** This method works best with well-chosen instructions; vague or poorly worded prompts may reduce its effectiveness.

- **False reasoning risk:** Although the final response is accurate, the step-by-step explanation might contain flawed logic or made-up details.
- **Higher resource usage:** Producing more detailed responses can consume additional computational resources, especially when running at scale.

Best suited for: Mathematical word problems, multi-step reasoning tasks, complex logical challenges, long-form planning, and situations where showing the reasoning process is important for trust and verification.

2.2.4 Role-Based Prompting

Role-based prompting involves instructing the language model to adopt a particular identity or point of view before generating a response. This role could be professional—for instance, "You are an experienced business analyst"—or situational, such as "You are a teacher explaining this concept to first-year students." By framing the task through a specific persona, the model is encouraged to produce text that reflects the style, vocabulary, and reasoning one might expect from that role.

This strategy's advantage is its ability to guide the model's tone and focus toward a desired domain or communication style. For example, positioning the model as a financial advisor is likely to prompt formal, data-driven explanations, while casting it as a creative storyteller may elicit more imaginative and emotionally engaging narratives.

In multi-turn conversations, maintaining the assigned role can help ensure consistency of perspective, making interactions feel more coherent and contextually appropriate. Still, role adherence is not guaranteed; if the instructions are unclear or contradicted by subsequent prompts, the model may drift away from the intended persona.

Overall, role-based prompting is a straightforward yet powerful way to influence a model's voice, level of formality, and domain focus without the complexity of custom training [19].

2.2.5 Retrieval-Augmented Prompting

Retrieval-augmented prompting is a method in which the instructions given to a language model are paired with supporting information gathered from external sources such as document collections, structured databases, or specialized knowledge repositories. By supplying this extra context at the time of the query, the model is better equipped to provide answers that are accurate, relevant, and grounded in up-to-date facts rather than relying solely on what it learned during pre-training.

This approach is particularly valuable when working with subjects that change rapidly—such as medical research, legal frameworks, or current events—where relying on a static model could lead to outdated or incomplete responses. The process typically involves a retrieval component that searches for and extracts relevant materials in response to a user's request. These retrieved passages or data points are

then incorporated into the model's input, helping it focus on verifiable sources while producing its answer.

Retrieval-augmented prompting forms the backbone of many modern Retrieval-Augmented Generation (RAG) systems. One advantage of this setup is that it enables source transparency, since the supporting information can be cited or linked. However, the effectiveness of the approach depends heavily on the quality of the retrieved material. Poorly matched, irrelevant, or noisy data can mislead the model, and large amounts of retrieved text can consume the model's context capacity unless they are carefully filtered or summarized.

When properly implemented, retrieval-augmented prompting strikes a balance between the broad linguistic capabilities of large language models and the precision of targeted, authoritative information sources, making it a practical choice for high-stakes or knowledge-intensive applications [20].

2.2.6 Limitations of Existing Prompting Methods

Although existing prompting methods have improved task performance, they often lack a structured framework. In most cases, the decision-making logic is embedded within unstructured, free-text instructions, which makes them difficult to maintain, update, or audit over time. Methods like retrieval-augmented prompting and role-based prompting attempt to address certain limitations by either incorporating outside information sources or shaping the model's behavior through defined personas and task-specific guidance [21, 22]. However, these approaches still fall short of providing a fully standardized representation of the decision process. Decision Model and Notation (DMN) offers a potential solution by introducing a formal, transparent framework for encoding and managing decision rules.

2.3 Decision Model and Notation (DMN)

2.3.1 Origins and Standardization

Decision Model and Notation (DMN) is an internationally recognized standard maintained by the Object Management Group (OMG) [1], a consortium responsible for several widely adopted modeling languages such as UML (Unified Modeling Language) and BPMN (Business Process Model and Notation). DMN was formally introduced to address a longstanding gap in business process modeling—namely, the absence of a common, implementation-independent way to express decision logic that could be equally understood by business analysts, compliance officers, and software engineers.

In contrast to procedural workflow models, which focus on how a sequence of tasks is executed, DMN is concerned solely with what decisions are required and the reasoning or rules behind them. This clear separation between process steps and decision logic enables organizations to modify rules independently of the overall

process design, allowing faster adaptation to changes and reducing the likelihood of errors when updating systems.

Because DMN is an open standard, it can be implemented across different sectors and tools without potential dependence on a single provider. This interoperability allows seamless use between modeling platforms, rule engines, and process automation technologies. Since its initial release, the specification has continued to evolve, incorporating features that improve compatibility with Business Rule Management Systems (BRMS) and support more advanced decision modeling scenarios.

2.3.2 Key Components

- **Decision Requirements Diagram (DRD):** At the highest level, the DRD provides a visual map of the decision-making framework in DMN. It illustrates the relationships between individual decisions, the data they depend on, and any linked knowledge sources. By making these dependencies explicit, DRDs give stakeholders a clear overview of how decisions are connected, making it easier to grasp both the scope and the logical flow of the overall process.

- **Decision Tables:** Decision tables organize rules in a structured, tabular form, with input conditions laid out in columns and corresponding outputs arranged in rows. This approach is particularly effective for situations where outcomes depend on several variables, as it systematically covers all possible combinations and removes ambiguity. Their dual readability—by humans and by software—makes them an ideal link between conceptual business logic and its technical implementation.

- **Literal Expressions:** When decision rules are simple enough to be expressed as a short formula or conditional statement, literal expressions provide a compact solution. These are typically written in Friendly Enough Expression Language (FEEL), the official expression syntax for DMN. This ensures that calculations, conditional logic, and rule statements are expressed in a consistent, concise, and standardized way.

2.3.3 Advantages

- **Interpretability:** DMN is designed so that even stakeholders without a technical background can read and understand the decision logic, while still being precise enough for direct execution by automated systems. This shared readability helps bridge the gap between business and IT teams, promoting clearer communication and reducing the risk of misinterpretation.

- **Modularity:** Decision rules are structured as independent components, allowing specific elements to be updated or replaced without altering the entire model. This design aligns well with agile workflows and supports continuous, incremental improvements.

- **Auditability:** The standardized and formal nature of DMN makes it straightforward to review, document, and verify decision logic. In industries with strict regulatory requirements, this capability is essential for demonstrating how particular results were produced.
- **Interoperability:** As a publicly available standard, DMN can be implemented across different platforms and integrated with diverse process automation and decision-support systems, making it adaptable to organizations with varied technology environments.
- **Reduced Cognitive Effort:** The use of visual diagrams and tabular layouts enables decision-makers to interpret complex rules more quickly and intuitively than by parsing lengthy, free-text descriptions.

2.3.4 Applications Beyond BPM

Although DMN originated in the field of business process management and remains widely adopted in sectors such as banking, insurance, and logistics, its range of uses has expanded considerably in recent years:

- **Clinical Decision Support:** DMN can be used to formalize medical guidelines and treatment pathways, providing healthcare systems with consistent and explainable recommendations. Encoding clinical reasoning in this way promotes adherence to established best practices while still allowing flexibility to adapt decisions for individual patient circumstances [23].
- **Legal Reasoning:** DMN can represent legal decision-making processes, for example, by mapping law-defined conditions to case outcomes. This structured approach enhances transparency and supports the automation of compliance verification [24].
- **Educational Assessment:** Preliminary research has applied DMN to grading frameworks, where decision tables map assessment criteria to scores and feedback. This method makes evaluation rules explicit, benefiting both educators and learners by clarifying how grades are determined.
- **Hybrid AI Integration:** DMN is increasingly integrated with machine learning systems in hybrid architectures. A common pattern is to have an AI model produce a prediction, which is then validated or adjusted using a DMN-based rule set to meet regulatory or operational requirements.
- **Policy Automation:** Public sector organizations have begun adopting DMN to codify policy rules in a clear, maintainable format. This enables faster updates when laws or regulations change and ensures that decision logic remains transparent to stakeholders.

2.4 LLM Control Through Structured Representations

The quick expansion of large language models (LLMs) has driven increasing interest in methods for steering and regulating their outputs through structured, formal representations. Unlike conventional free-text prompts, which can be vague and difficult to keep consistent over time, structured approaches embed rules, relationships, or constraints that are directly interpretable by machines into the model's inputs or processing workflow. This added structure helps maintain logical consistency, factual reliability, and alignment with domain-specific standards.

Several strategies have emerged to operationalize this idea:

- **Knowledge Graph Integration:** By connecting LLMs to ontologies or domain-specific knowledge graphs, generated outputs can be aligned with predefined semantic relationships. Incorporating ontology constraints into model queries can prevent contradictions and encourage the use of consistent, standardized terminology [25].
- **Logic-Driven Prompting:**This technique embeds formal logic statements or explicit rule sets directly into the prompt. Incorporating symbolic rules into prompt templates can shape model reasoning, enforce constraints, and reduce the likelihood of fabricated or logically inconsistent responses—an especially important factor in regulated areas such as law and finance [26].
- **Hybrid Symbolic–Statistical Systems:** In this approach, LLMs are combined with symbolic reasoning systems, allowing the generative flexibility of the model to be anchored by deterministic, rule-based logic. One research describes systems in which the LLM manages natural language interpretation and generation, while a symbolic engine performs rule validation, fact-checking, or structured [27].

2.5 Automated Feedback Systems in Education

2.5.1 Historical Background

The idea of using computers to give feedback to learners is not new. Early examples appeared in the late 1960s and 1970s, when rule-based programs began performing simple checks on student work. Grammar checkers built into early word processors and Computer-Assisted Instruction (CAI) platforms were among the first tools to offer automated responses. These systems operated on a fixed set of rules, matching learner input against predefined patterns to deliver scripted feedback. By the 1980s and 1990s, more capable Intelligent Tutoring Systems (ITS) emerged. Drawing on decision trees, error taxonomies, and domain-specific logic, they could adapt responses to a learner's actions. While this was a notable improvement, such systems still had trouble coping with unexpected inputs or unconventional problem-solving

approaches because their behavior was limited by the scope of their programmed rules.

2.5.2 AI-Enhanced Feedback

The shift from purely rule-based systems to those informed by machine learning marked a major turning point. Instead of being constrained by prewritten rules, AI-driven tools learned patterns from large datasets, allowing them to provide feedback that was more adaptable and sensitive to context.

Several notable developments illustrate this shift:

- **Automated Essay Scoring (AES):** Tools like e-rater and IntelliMetric assess writing by analyzing linguistic features, structure, and coherence [28]. Widely adopted in large-scale assessments, these systems save time but have drawn criticism for their opacity and for missing subtler qualities in writing, such as creativity or persuasive style.
- **Programming Tutors:** Systems such as CodeHunt and CodeRunner give targeted feedback on coding exercises, flagging syntax issues, logical errors, and inefficient solutions [29]. By running code against test cases, they can pinpoint exactly where a solution fails.
- **Formative Feedback in STEM:** NLP-based systems have been designed to evaluate step-by-step reasoning in mathematics and science, helping identify misconceptions and offering hints that guide students toward the right answer without simply telling them [30].

This move toward data-driven adaptivity meant feedback could be tailored, at least partially, to an individual learner's history, strengths, and recurring errors.

2.5.3 LLM-Powered Feedback

The newest wave of systems takes advantage of large language models (LLMs). These models go beyond rigid templates, producing feedback that is fluent, conversational, and contextually aware. Their key strengths include:

- **Natural Language Explanations:** Feedback is phrased in everyday language, making it approachable for a wide range of learners.
- **Context-Sensitive Suggestions:** Guidance is shaped by the specific content of a student's work, not just generic rules.
- **Personalized Learning Paths:** Some platforms generate tailored sequences of practice tasks based on the learner's prior work.

Yet, alongside these benefits are some persistent challenges:

- **Consistency and Accuracy:** Without safeguards, the quality of feedback can fluctuate, and occasional factual mistakes can slip through.
- **Rubric Alignment:** Without clear guidance, LLMs might give feedback that doesn't match the instructor's grading standards.
- **Educator Control:** Many teachers lack straightforward ways to adjust how these models operate, since much of the logic is hidden inside complex prompts.

Incorporating Decision Model and Notation (DMN) into LLM-driven feedback systems offers a way forward. DMN's structured, transparent decision rules can keep LLM outputs consistent with course rubrics, making the feedback both flexible and reliable.

2.5.4 Evaluation Techniques for Automated Feedback and Grading Systems

Assessing the quality and impact of automated feedback tools calls for more than a single metric. A combination of approaches helps capture both the technical performance and the educational value of these systems:

- **Automatic Grading Systems:** NLP or LLM models automatically score open-ended responses, saving time in large classes and ensuring consistency. However, they can miss subtle reasoning or creativity, and their performance hinges on model accuracy [31].
- **Automated Feedback Systems:** Provides instant formative feedback, encouraging real-time improvement. While effective for boosting engagement, they sometimes lack depth or precision in complex subject areas [32].
- **System Usability Scale (SUS):** Provides a fast way to measure user satisfaction, but results can be influenced by past experience with similar tools. [33].
- **Text Classification Systems:** Categorize responses for faster analysis and targeted feedback. This is efficient at scale but can oversimplify nuanced reasoning [34].
- **Multimodal Evaluation Systems:** Combine multiple data types (text, visuals, audio) to gain a fuller understanding of student learning. While offering rich insights, these systems demand significant resources and integration effort [35].
- **Content-Overlap Metrics (ROUGE):** Compare generated feedback with expert-written responses based on shared wording. Objective and widely used, but limited in capturing true meaning when synonyms or rephrasings are involved [36].

- **Model-Based Metrics (BERTScore):** Evaluate semantic similarity between generated and expert feedback using contextual embeddings. Strong correlation with human judgment, but more resource-intensive and less transparent [36].
- **Human Evaluation (Multi-Dimensional):** Experts manually review feedback for clarity, accuracy, and tone. This yields rich insights but is time-consuming and can suffer from inconsistency across reviewers [37].

2.6 Integrating Evaluation into Practice

In practice, effective evaluation blends these methods. For example, automatic grading might be paired with human review to combine efficiency with deeper qualitative insights. Semantic similarity checks could be used alongside DMN-based rubric rules to ensure feedback is both accurate and aligned with instructional goals.

A well-designed evaluation plan does more than verify system performance—it helps ensure that automated feedback tools remain trustworthy, fair, and genuinely useful for learning. As these systems continue to evolve, maintaining this balance will be essential for their long-term acceptance in educational settings.

2.7 Identified Research Gap

A close reading of the existing literature points to several persistent shortcomings in how large language models (LLMs) are currently applied, particularly in settings where the stakes are high and the margin for error is small.

First, most LLM-powered systems in education operate without a clear or sustainable mechanism for controlling how decisions are made. While they may produce impressively fluent responses, the reasoning behind those responses is buried deep within opaque model architectures. In a classroom or assessment setting, this lack of transparency is more than an academic concern—it affects trust. Teachers need to know not only what the system decided, but why it decided so, especially when the output influences grades or shapes learning pathways. Without a transparent control layer, maintaining consistent quality over time is difficult, and errors can easily slip through unnoticed.

Second, the dominant method for influencing model behavior—prompt engineering—is still something of an art form. Prompts are often created through personal experimentation, which means they may work well for one instructor or developer but be hard to replicate elsewhere. This ad hoc approach makes it difficult to share, refine, or scale prompts across larger teams. Over time, even small, untracked changes can cause prompts to drift away from curriculum goals or institutional standards, undermining both reliability and fairness.

Finally, there is an underused tool hiding in plain sight: Decision Model and Notation (DMN). Well established in the business world for capturing decision logic in a structured, low-code format, DMN's visual diagrams and rule tables are easy for

both technical and non-technical stakeholders to understand. In theory, this makes DMN a perfect fit for guiding LLM outputs—offering a way to embed grading rubrics, pedagogical rules, and institutional policies directly into the model's decision-making process. Yet despite its potential, there is almost no systematic research applying DMN to LLM prompting in education.

Taken together, these issues open the door to a clear and compelling line of inquiry:

Can integrating DMN into LLM-based feedback systems improve not just accuracy, but also interpretability and ease of use—making them more reliable tools for educators and learners alike?

Chapter 3

Methodology

3.1 Overview of the Proposed Framework

The framework developed in this study introduces a structured way of guiding Large Language Models (LLMs) by relying on Decision Model and Notation (DMN). Rather than packing all the rules and constraints directly into long text prompts, as is common in today's practice, the framework keeps the decision-making logic outside the prompt itself and represents it in a standardized form. This separation is important: it makes the rules easier to read, update, and audit, while the LLM is left to apply them when generating responses.

Why This Approach?

Most current techniques for steering LLMs—prompt engineering, few-shot examples, or fine-tuning—have clear drawbacks. Prompt engineering is often improvised and hard to replicate, since the reasoning steps are hidden inside unstructured text. Fine-tuning requires access to model weights and substantial resources, which is rarely feasible in an educational setting. And while classical rule-based systems provide consistency and control, they lack the flexibility and natural fluency of modern language models.

The proposed framework is meant to bring these two worlds together. By embedding explicit decision logic into the interaction with the LLM, it becomes possible to produce feedback that is not only more reliable but also easier for educators to maintain over time. This is particularly valuable in teaching, where grading rubrics and assessment standards change regularly, and instructors need to ensure that students receive consistent and fair feedback.

3.1.1 Core Structure: DMN Triples

The system works around a modular unit called a DMN triple, which captures each decision in three parts:

1. Input Data Elements – the variables needed to reach a decision (e.g., assignment clarity, test case results).

2. Decision Tables – the rules that connect different combinations of inputs with particular outcomes.

3. Literal Expressions – the text or feedback messages that are tied to each outcome.

This framework provides the model with a sequential structure for reasoning. Instead of asking the LLM to reason in a completely open-ended way, it follows a three-part process: extract relevant inputs, check them against the rules, and return the corresponding output message. It is similar in spirit to Chain-of-Thought prompting, but here the reasoning path is explicitly defined and modular, rather than hidden inside a text instruction.

3.1.1.1 Theoretical Roots

The framework I propose does not come out of nowhere. It builds on three traditions that have been around for quite some time: symbolic AI, explainable AI, and more recent work on prompting strategies for large language models.

- **Symbolic AI and Expert Systems**

  If we look back at the early days of AI, systems were largely symbolic. Expert systems like MYCIN or DENDRAL worked entirely on the basis of explicit rules written by specialists. The appeal of these systems was not just that they produced answers but that every answer could be explained by tracing back through a set of "if–then" conditions. That kind of transparency has always been valuable in education, where being able to show why a student received a certain piece of feedback can matter just as much as the feedback itself. The DMN approach essentially revives this tradition in a modern form: decision tables capture rules clearly, but the reasoning process is still transparent to both humans and machines.

- **Explainable AI**

  In more recent years, there has been a strong push for explainability in AI. This is partly a reaction to the rise of black-box systems like deep learning, which are powerful but often impossible to interrogate. In education, this lack of clarity can be problematic—students and instructors alike need to understand where feedback comes from. DMN naturally provides that kind of traceability. Every outcome links directly to a rule in the decision table, so if a learner wants to know why their assignment received a certain comment, the instructor can point to the logic behind it. This makes the system accountable in a way that many current AI tools are not.

- **Prompting Strategies for LLMs**

  Finally, the design draws on what we've learned from prompting large language models. Research has shown that models perform better when their reasoning

is broken into steps, as in Chain-of-Thought prompting [38]. The problem is that these strategies usually rely on long text templates that are clumsy to maintain. Each time the rules or grading rubrics change, someone has to go back and re-engineer the prompt. DMN offers a neater solution: it externalizes the rules into a model that can be swapped or updated without rewriting the prompt itself. In other words, it takes what works about structured prompting and makes it reusable and easier to manage.

Taken together, these three roots show that the DMN-guided framework is less a radical departure and more a synthesis. It borrows the transparency of symbolic AI, the accountability of explainable AI, and the structured reasoning of prompt engineering, tying them together in a way that fits educational needs.

3.1.1.2 Benefits in Practice

Bringing these ideas together has some very practical advantages.

- **Transparency**

  The first is transparency. Because every decision maps to a specific rule, there is no guesswork involved in explaining how a piece of feedback was generated. This not only builds trust with students but also reassures instructors that the system is operating fairly.

- **Maintainability**

  Another benefit is maintainability. Instructors often revise feedback rules, and doing that in a conventional prompt can be messy. With DMN, the only thing that changes is the decision table; the prompt structure remains stable. That makes the system much easier to update.

- **Scalability**

  The framework is also scalable. A single prompting pipeline can support very different kinds of tasks—for instance, programming assignments and essay grading—simply by swapping in a different DMN file. This flexibility makes it possible to deploy the system across entire courses or even departments without having to rebuild everything from scratch.

- **Accessibility**

  Perhaps most importantly, the approach is accessible to non-programmers. Instructors don't need to know how to write prompts in a careful, technical way. Instead, they can design decision tables in a graphical editor and let the LLM handle the rest. This lowers the barrier to entry and brings more educators into the process of shaping AI feedback.

In short, the framework is designed to be transparent, easy to maintain, flexible across contexts, and inclusive of subject matter experts who might not be technically trained.

3.1.1.3 Remaining Challenges

Still, the design has some limits.

- **Ambiguous Inputs**

  Student work can be unpredictable. Sometimes responses are incomplete, off-topic, or simply unusual. DMN works best when inputs are clean and structured; LLMs can help interpret the messier cases, but this can lead to mistakes.

- **Balancing Strictness and Flexibility**

  There is also the question of balance. DMN is deterministic: given the same inputs, it always produces the same output. LLMs, on the other hand, are probabilistic and can vary in their responses. Finding the right balance between rule-based certainty and generative flexibility is tricky. Too much rigidity, and the system feels unfair; too much flexibility, and it risks inconsistency.

- **Human Oversight**

  Even with these precautions, human oversight remains essential. Feedback is not only about correctness but also about tone, encouragement, and appropriateness for the student's level. These nuances are difficult to encode into rules, meaning that instructors still need to act as a final check before feedback is delivered.

- **Technical and Resource Barriers**

  Lastly, building and maintaining this kind of system does require infrastructure. Preprocessing pipelines, secure data handling, and integration with learning platforms are not trivial. For some institutions, these technical demands may slow adoption.

Briefly, the DMN-guided prompting framework connects several traditions in AI, combining established strengths with emerging opportunities. It has the potential to make AI feedback systems more transparent, adaptable, and collaborative, while also underscoring the areas where human judgment remains essential. Its promise is real, but so are its challenges—particularly the need to balance the precision of rules with the flexibility required in real-world education.

3.2 Framework Architecture

Figure 3.1 illustrates the layered architecture of the DMN-guided prompting system. The framework is designed to separate formal decision logic from free-text prompting, ensuring outputs are both interpretable and auditable. The architecture is composed of several interdependent components:

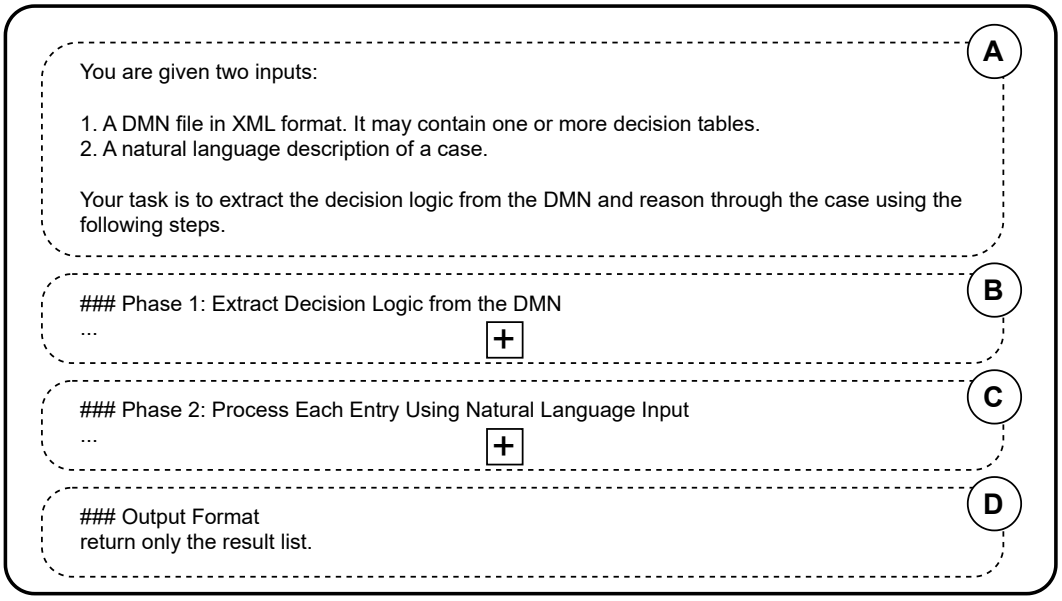

Figure 3.1: The general prompt structure for DMN-guided assessment.

3.2.1 DMN Model Creation

At the foundation of the system, instructors or subject matter experts define decision logic in the form of DMN models. These models can be constructed using graphical modeling environments such as Camunda Modeler, which provide user-friendly drag-and-drop interfaces.

Each decision is represented as a triple:

1. Input data elements (variables or student attributes to be evaluated),
2. Decision table (conditional rules that map inputs to outcomes),
3. Literal expressions (the human-readable feedback or instructional messages associated with each outcome).

This structured representation ensures that decision logic is standardized and can be easily modified without rewriting prompts.

3.2.2 Input Data Preparation

Before evaluation, student submissions—such as essays, process models, or programming assignments—must be transformed into a form that the LLM can process.

A preprocessing script is used to convert raw submissions into structured natural language descriptions.

This abstraction layer ensures that the LLM interprets student work consistently, without needing to parse diagrams, code syntax, or other domain-specific formats directly.

For example, instead of parsing a BPMN diagram, the system generates a text-based summary describing model elements, which can then be evaluated against the DMN rules.

This stage acts as a bridge between raw student data and machine-readable decision logic.

3.2.3 Prompt Construction

Once the DMN and the case description are prepared, they are embedded into a structured multi-part prompt. As shown in Figure 3.1, the prompt is divided into four explicit parts (A–D):

Part A: Provides context by embedding the full DMN XML file and the case description.

Part B: Instructs the LLM to parse the DMN into a structured dictionary of decision triples.

Part C: Guides the model in applying decision logic, matching student input values against the decision table conditions.

Part D: Directs the LLM to return final results in a clean, machine-readable JSON format.

This modular construction reduces ambiguity and ensures that reasoning follows a transparent, repeatable workflow.

3.2.3.1 Part B: Parsing the DMN Model

The first step in the execution pipeline is to parse the DMN content, which is usually provided in XML. Instead of asking the LLM to interpret XML tags directly—an approach that can be error-prone—the framework guides the model to translate each decision into a simplified dictionary format. This makes the rules easier to work with and keeps them consistent.

Each decision is captured as a triple containing(Figure 3.2):

- Rule name – a descriptive label, often given in the DMN annotations.
- Input fields – the variables needed to reach a decision.
- Decision table – the mapping between conditions and their outcomes.
- Literal expressions – the specific feedback messages tied to those outcomes.

B

Phase 1: Extract Decision Logic from the DMN

From the DMN file:
- Identify each decision rule (preferably using annotations as rule names).
- For each rule, extract a structured dictionary entry as follows:

```json
{
  "rule Name": {
    "input": [list of required input field names],
    "decision table": {
      "name": "<decision table name>",
      "content": "<decision table XML snippet>"
    },
    "literal expression": {
      "name": "<output name>",
      "content": "<list of outcomes and corresponding messages>"
    }
  }
}
```

Figure 3.2: Part B contains instructions for parsing DMN into structured triples.

For example, a grading decision might rely on input fields such as clarity, accuracy, and completeness. The decision table would then connect different combinations of these values to outcomes like excellent, satisfactory, or needs improvement. Each outcome is paired with a ready-made message, for example: "Your explanation is clear and thorough" or "Consider adding more detail to improve completeness."

This process protects the LLM from raw XML and instead provides a structured, human-readable format. By packaging each rule in this modular way, the framework ensures that decisions are evaluated one at a time, without unrelated rules interfering.

This step is important because it allows multiple decision rules to exist within a single DMN file, making the framework more flexible. At the same time, it creates a structured knowledge base of triples that guide reasoning, ensuring that each result links back to its rule. By keeping the formatting consistent, the chance of misinterpretation is greatly reduced.

3.2.3.2 Part C: Evaluating the Input Text

Once the rules are parsed, the LLM can move on to the student's work. Figure 3.3 illustrates that this stage applies the dictionary entries to the input text through a four-step process:

1. **Extract Input Values (C1):** This step asks the LLM to look at the input list in the dictionary and pick out the needed values.These inputs then guide the model in finding their corresponding values in the given text (second instruction in C1).

2. **Evaluate the Decision Table (C2):** The extracted values are then compared against the conditions defined in the table. Numeric ranges and symbolic expressions are spelled out clearly—for instance, [40..50] is inclusive of both 40 and 50, while [40..50) includes 40 but excludes 50. Providing these explicit rules

helps the model avoid misinterpretation and keeps the evaluation consistent. Examples and edge cases are often included in the prompt to reduce the risk of hallucinations.

3. **Retrieve the Literal Expression (C3):** Once a matching condition is found, the model gets the feedback message for the matched outcome. Instead of inventing new feedback, it selects from instructor-approved options.

4. **Return the Final Message (C4):** The last step ensures that only the feedback message is returned, usually in JSON format. By filtering out intermediate steps—like reasoning traces or extracted dictionaries—the system delivers clean, unambiguous outputs.

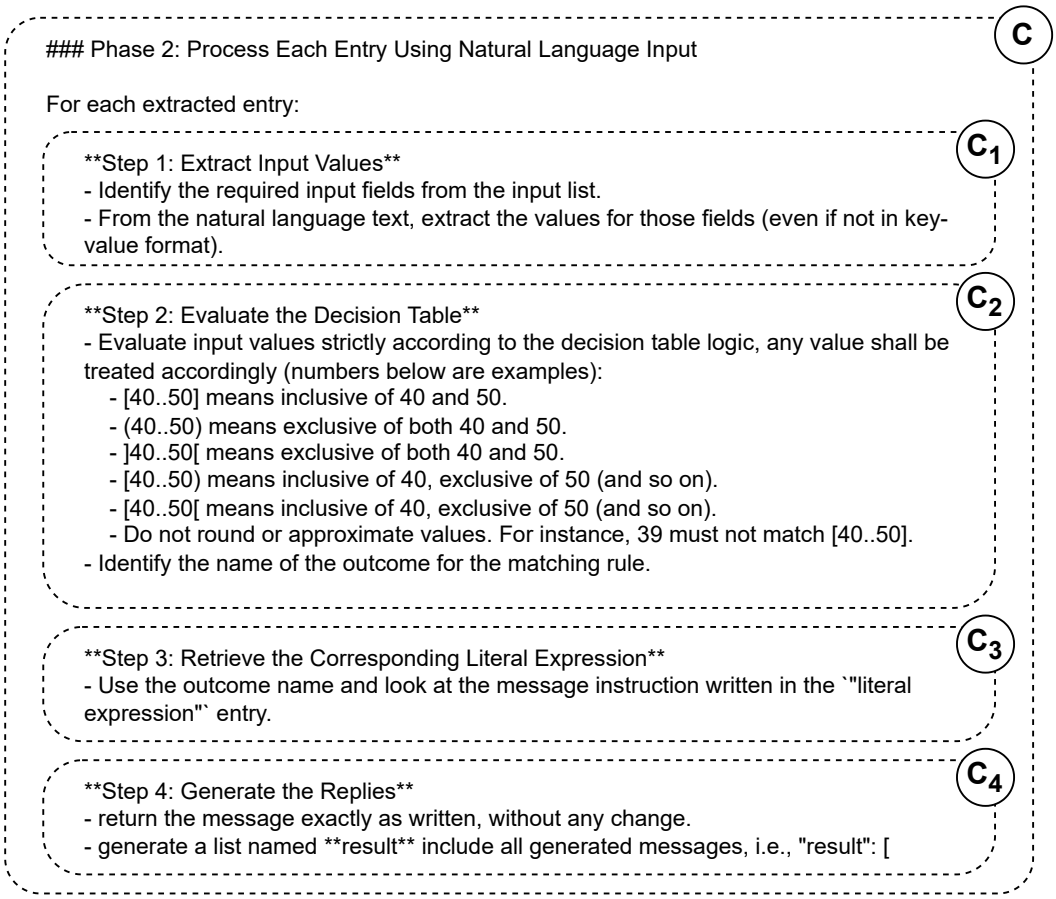

Figure 3.3: Part C instructions include utilising parsed DMN to evaluate input text.

This structure makes the evaluation orderly instead of random. The LLM can still work with natural language, but its choices are firmly guided by DMN rules.

This step can help the system avoid hallucinations by following predefined rules, making responses more reliable. Clear definitions for numbers and symbols help handle tricky cases. Since the same logic is always applied, students get consistent feedback. The outputs are also clean and structured, so they can be easily used in feedback systems.

3.2.3.3 Result Compilation and Final Output

In the final stage, the framework pulls everything together. Each decision triple is evaluated independently, and the results are merged into a single response. The output is delivered in a standardized format.

This setup makes the results easy to check, save, and use again. For example, the output can be stored together with the student's work and the DMN model version, creating a clear record for accountability.

A key strength of this design is that the prompt itself doesn't need to change when rules are updated. If an instructor wants to refine the grading criteria, they simply edit the DMN file, and the framework adapts automatically. Separating decision logic from prompt syntax makes the system far easier to maintain.

The framework brings together a number of practical strengths. Main advantages are :

- **Modularity:** different parts can be created, updated, or reused without disrupting the whole system, which makes it easier to maintain and scale.
- **Interpretability:** it also improves interpretability, since every decision rule and its outcome can be traced back clearly, making things clearer.
- **Collaborative design:** Instructors or subject experts can directly shape the decision logic, rather than relying entirely on technical specialists.
- **Transparency:** Most importantly, the framework improves transparency by replacing improvised prompts with a structured and trackable process. This is especially important for complex, rule-based tasks.

3.3 Integration with LLM APIs

The DMN-guided prompting framework requires reliable access to advanced large language models (LLMs) in order to transform structured rules into natural-language feedback. To achieve this, the system integrates directly with two leading LLM providers through their APIs: GPT-4o via the OpenAI API and Gemini 1.5 Pro via the Google Gemini API. These models were chosen because they represent the current state of the art in reasoning performance and text generation, while also supporting the input/output formats required for structured evaluation.

3.3.1 Goals of Integration

The integration was designed with three main goals:

1. **Deterministic Behavior:** In an educational setting, consistency is critical. To minimize random variations in output, both models were configured with a temperature of 0, ensuring that identical inputs yield identical outputs.

2. **Scalable Evaluation:** Since student assignments may involve multiple decision rules and detailed case descriptions, the `max_tokens` parameter was set high enough to accommodate both full DMN parsing and contextual reasoning completely.

3. **Structured Outputs:** To enable traceability and post-processing, the framework makes use of JSON mode whenever possible. This forces the models to output in a machine-readable format, avoiding the risk of noisy or verbose responses.

3.3.2 API Workflow

The interaction between the framework and the APIs proceeds in a structured pipeline:

1. **Preprocessing:** The DMN model is exported into XML format, and student submissions are converted into structured natural language descriptions.

2. **Prompt Assembly:** A multi-part prompt (Parts A–D) is generated, embedding both the DMN XML and the case description.

3. **API Call:** The prompt is transmitted via HTTPS requests to either the OpenAI or Gemini endpoint, with specified parameters for temperature, token length, and output format.

4. **Response Handling:** The system parses the JSON-formatted result, aligns it with the corresponding decision rule, and stores it alongside the DMN version and student submission for traceability.

This setup makes the LLM act less like a free text generator and more like a reasoning tool that follows clear rules.

3.3.3 Cross-Model Consistency

Both GPT-4o and Gemini 1.5 Pro were tested under the same conditions, making it easier to separate the effects of the framework from the strengths or limits of each model. For instance, if both models improve in consistency when guided by DMN rules compared to Chain-of-Thought prompting, the added value can be attributed to the framework rather than to a particular LLM.

3.3.4 Benefits of API-Based Integration

Integrating via APIs provides several advantages:

- **Abstraction:** The framework does not need to manage low-level model training or optimization; instead, it leverages high-level API endpoints.

- **Portability:** Different LLMs can be swapped in or out with minimal changes, making the system adaptable to future model releases.

- **Reproducibility:** By fixing parameters such as temperature and enforcing JSON-mode outputs, the system ensures that results can be replicated across runs.
- **Ease of review:** Each API call is logged with metadata including timestamp, DMN version, and model configuration, leaving a complete trace.

3.3.5 Open-Source Implementation

To ensure transparency and replicability, the implementation scripts for API integration, prompt templates, and evaluation routines have been released as open-source resources. This allows researchers and practitioners to adapt the framework to their own domains, whether in education, business process management, or healthcare feedback systems.

3.3.6 Implications for Educational Use

For educators, API-based integration means that they do not need to engage directly with prompt engineering or model tuning. Instead, they can focus on encoding decision logic in DMN, while the framework ensures that the LLMs apply these rules reliably. By sharing the work between parts, the system becomes easier to adopt, particularly among non-technical instructors.

3.3.7 Human-in-the-Loop Verification

While the DMN-guided prompting framework aims to bring more structure and reliability to automated feedback, it is important to remember that no AI-based system is flawless. To ensure the credibility of feedback provided to students, a human-in-the-loop mechanism was built into the process.

First, every piece of feedback generated by the LLM was reviewed by an instructor or subject matter expert before it reached students. This step served as a safeguard against mistakes or misinterpretations that could still occur, even when the decision logic was formally defined. Instructors confirmed not only the factual accuracy but also the appropriateness of tone, ensuring that the feedback was constructive and aligned with educational goals.

Second, during this review process, each feedback instance was explicitly tagged as either correct or incorrect. This binary labeling created a valuable secondary dataset. Over time, this dataset can serve two important purposes: (1) to measure the overall accuracy of the framework under real conditions, and (2) to provide labeled examples for possible fine-tuning of future models. In this way, the human-in-the-loop layer does not simply act as a safeguard but also as a feedback loop that strengthens the system over time.

Finally, this stage provided instructors with a sense of control and trust in the system. Rather than feeling replaced by automation, educators could see themselves as active participants in shaping and refining AI-generated outputs. This sense of

shared responsibility is especially important for adoption in educational contexts, where trust in feedback is central to student learning.

3.3.8 Ethical Considerations

Ethical design choices were also prioritized to make sure the framework operates in a way that respects both students and educators. Four main areas were addressed:

1. **Data Protection and Privacy:** Before any processing, all student submissions were anonymized. Personally identifiable information (PII) was removed or masked, ensuring that the system only worked with the educational content itself. Secure storage protocols were also applied to both DMN models and student work, protecting them from unauthorized access.

2. **Transparency:** Students were informed whenever AI assistance was used in the generation of feedback. Communicating the role of AI in the process helped maintain openness, avoided misconceptions, and gave students the opportunity to understand the origin of their feedback. Transparency was treated not as optional but as a core ethical responsibility.

3. **Bias Monitoring and Fairness:** Bias is a well-documented concern in AI systems. To counter this, feedback was regularly monitored to detect any patterns that might indicate unequal treatment across different student groups or assignment types. The use of deterministic DMN rules already helps reduce bias by basing outputs on clear, predefined rules, but human oversight was still essential to ensure fairness.

4. **Instructor Agency and Responsibility:** Importantly, the system was designed to support educators rather than replace them. By giving instructors the final authority over what feedback students receive, the framework reinforces the ethical approach that teachers remain accountable for student evaluation. AI is positioned as an assistant, not an autonomous decision-maker.

Together, these measures ensure that the framework is not only technically robust but also ethically sound, balancing efficiency with responsibility. The integration of human oversight, data protections, and transparency measures supports both educational integrity and student trust.

Chapter 4

Case Study

4.1 Introduction to the Case Study

The development of the DMN-guided prompting framework was motivated by a practical need: how to make feedback generated by large language models more reliable, transparent, and pedagogically useful in real learning environments. While theoretical discussions and controlled examples can demonstrate the promise of such a framework, its real value can only be judged when applied in practice. For this reason, we designed a case study to evaluate the framework within an actual academic course.

This case study focused on three main objectives.

1. **Technical performance:** First, we tested how well the framework worked by comparing its accuracy with a baseline prompting method. This allowed us to examine whether embedding decision logic genuinely reduces errors or whether it simply adds extra complexity.

2. **Pedagogical utility:** Second, we examined how well the feedback matched instructor expectations and whether it actually helped students learn and improve their work.

3. **User perceptions:** Finally, we looked at how students felt about the system—whether they saw the feedback as clear, reliable, and helpful for their learning.

This study was carried out in the field of business process modeling education, which is especially well suited for this kind of research. Unlike open-ended writing or creative tasks, process modeling is grounded in well-defined principles and established best practices. This makes it possible to encode evaluation criteria as decision rules in DMN and then assess how effectively the framework can apply these rules to real student submissions.

The case study design also ensured that the framework was tested under realistic conditions. Students worked in groups on process models as part of their regular coursework, and their submissions were evaluated using the same criteria that

instructors would normally apply. To maintain fairness and accuracy, every AI-generated comment was reviewed by instructors before being shared with students. This not only safeguarded the instructional quality but also produced a labeled dataset that could be used to measure system accuracy and inform future improvements.

In short, this case study tested the DMN-guided prompting framework in a real classroom. It showed how well the system works technically and how it fits into teaching and learning with both instructors and students. By combining performance results with student feedback, the study gives a clear picture of the framework's strengths, limits, and future potential.

4.2 Educational Context

This case study was carried out in the graduate-level course Business Process Design and Intelligence at the Department of Computer and Systems Sciences, Stockholm University. The course focuses on equipping students with practical and theoretical skills in analyzing, modeling, and improving organizational processes. A total of 24 student groups participated in the study, with each group consisting of around six members working collaboratively throughout the semester.

The central assignment required students to design business process models using Petri nets, a formal modeling language known for its ability to capture concurrency, synchronization, and resource dependencies in processes. Students created their models using the WoPeD modeling tool [39], which is widely used in teaching environments for its intuitive interface and strong analytical features.

To guide their modeling efforts, students were expected to apply established Business Process Redesign (BPR) principles. These included:

- **Triage:** reorganizing cases to ensure efficiency in handling.
- **Parallelism:** enabling tasks to occur simultaneously where appropriate.
- **Automation:** replacing manual tasks with automated solutions to reduce delays.
- **Task elimination:** removing unnecessary activities to streamline processes.
- **Resequencing:** reordering tasks for improved flow and effectiveness.

The assignments were based on clear principles, which made it easier to judge student work. This also helped test the DMN-guided framework, since its rules followed the same principles.

The students approached their tasks using an agile methodology [40], which emphasized incremental improvement and continuous refinement of their process models. This method encouraged experimentation and iterative learning, allowing students to repeatedly test and revise their designs. From a research perspective, this provided a rich and varied set of submissions, capturing both correct implementations of BPR principles and common student mistakes. These variations offered an ideal

basis for evaluating whether the DMN-guided prompting framework could correctly identify strengths and weaknesses in the models.

It was entirely optional to participate in the study. Students were fully informed about the goals and scope of the research, and informed consent was obtained from all participants. To safeguard student privacy, all submissions were anonymized before being processed by the system. This ensured that the focus remained on evaluating the framework rather than on individual student performance.

This particular educational setting was deliberately chosen because business process modeling provides an excellent environment for structured feedback generation. The domain is rule-intensive: models can be objectively assessed against well-defined principles, yet student submissions often exhibit subtle variations in labeling, task structuring, or sequencing. These characteristics created both opportunities and challenges for the framework. On the one hand, the existence of clear evaluation rules made it possible to encode decision logic in DMN tables. On the other hand, the variability in how students expressed their ideas tested the framework's ability to balance deterministic rules with the flexible interpretation needed to handle natural language inputs.

The combination of a well-defined rule system (BPR principles), collaborative student projects, and iterative model refinement made this course a strong environment for evaluating the DMN-guided prompting framework. It provided a realistic but controlled setting in which to measure technical performance, assess the relevance of generated feedback, and explore how such a system might eventually be adopted in broader educational contexts.

4.3 Deployment of the Framework

Figure 4.1 illustrates how the DMN-guided prompting framework was embedded into the course environment. The setup focused on three main roles, each playing a critical role in making the system work:

- **Instructors** – Instructors were responsible for encoding the evaluation criteria into DMN models. Each rule was represented as a decision triple, consisting of input data, a decision table, and a feedback expression. By formalizing their expectations in this way, instructors ensured that feedback rules were explicit, traceable, and consistent. This approach shifted the role of instructors from crafting individual prompts to defining systematic rules that could be reused across multiple assignments.

- **Students** – Students submitted their process models, developed as part of their coursework. These models served as real-world test cases for the framework. Importantly, the submissions were anonymized, protecting student identities while still allowing meaningful evaluation. This setup allowed students to indirectly benefit from AI-supported feedback without compromising ethical safeguards.

- **The DMN-guided system** – The system served as the reasoning engine, applying the encoded rules to student submissions. Using the DMN triples as a structured guide, the system generated tailored feedback automatically. Unlike traditional black-box prompting approaches, this process linked each piece of feedback directly to a rule, ensuring transparency and traceability.

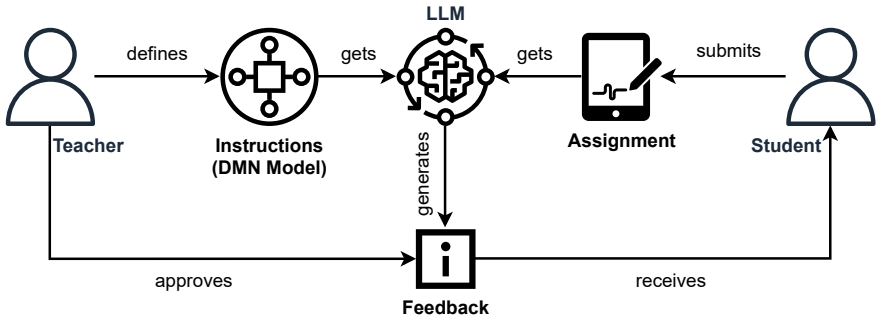

Figure 4.1: Our case study's implementation of the DMN-Guided Prompting Framework.

A central part of the deployment was the human-in-the-loop verification process. The AI-generated feedback was not shared with students immediately. Instead, instructors carefully reviewed each comment and labeled it as correct or incorrect. Only validated feedback was delivered to students. This additional step served multiple purposes:

- It ensured the feedback stayed reliable, prevented the dissemination of potentially misleading feedback.
- It created a labeled dataset for subsequent analysis.

Such a dataset is particularly valuable for future research, as it provides training material for refining and fine-tuning LLMs in educational contexts.

To evaluate robustness and generalizability, the framework was tested with two different large language models: GPT-4o (via OpenAI's API) and Gemini 1.5 Pro (via Google's API). Running both models under the same conditions made it possible to distinguish effects caused by the framework design from those related to the specific capabilities of a given LLM. In other words, the study could test whether the benefits came from the DMN-guided prompting itself, and not just from the behavior of one specific model.

In parallel, a Chain-of-Thought (CoT) [8] prompting setup was implemented as a baseline. CoT prompting has become a popular technique for improving reasoning in LLMs by explicitly guiding them to break problems into smaller steps. However, it lacks the formal structure and interpretability of DMN rules. By comparing DMN-guided prompting to CoT, the study could assess whether embedding explicit decision logic provided measurable improvements in accuracy, consistency, and educational alignment.

Another benefit of this deployment design is that it promoted accountability and transparency. Every feedback instance was traceable to a rule, every reviewed case contributed to the dataset, and every model run was logged with its corresponding DMN version. By keeping this trail of records, we could both assess the system systematically and allow others to repeat the study, confirming and extending our findings.

Finally, this division of roles between instructors, students, and the system supports scalability and broader adoption. Instructors retain control over pedagogical content by defining rules, students receive structured and validated feedback, and the system acts as a reliable assistant rather than a replacement. It creates a foundation for using AI feedback more widely in education, making sure that innovation is matched with careful oversight.

4.4 Technical Implementation

A dedicated software component was designed to bridge the gap between student-created process models and the LLM-based evaluation framework. The tool automatically translated Petri net models into structured natural language descriptions. These descriptions captured not only the overall control flow but also key resource semantics, such as task assignments and dependencies. By doing so, the system avoided the complexity of parsing graphical diagrams directly while still preserving all essential information required for accurate evaluation.

The feedback mechanism was grounded in a set of nine decision rules, each carefully aligned with a recognized principle of Business Process Redesign (BPR). These rules were formalized in Decision Model and Notation (DMN) tables, ensuring that each condition corresponded to a predefined feedback message. For example, if a process failed to apply parallelism correctly, the system would match this case to a DMN rule and retrieve the appropriate instructor-approved feedback. This design not only supported consistent and repeatable evaluation but also helped prevent arbitrary or improvised outputs from the LLM.

Another important aspect of the implementation was its modularity. Since the rules were encoded separately in DMN, they could be updated or extended without requiring changes to the underlying prompt structure. This separation of concerns allows the framework to evolve as teaching practices or course requirements change, making it scalable and adaptable to different learning environments.

To support transparency and encourage adoption by the wider community, the complete implementation has been published openly on GitHub. Along with the code, example DMN files and documentation are provided to make replication and adaptation easier. This openness not only enhances reproducibility but also lays the foundation for collaborative improvement, where educators and researchers can build upon the existing work to refine the framework further.

Finally, by using structured DMN rules as the backbone of the evaluation, the technical setup also served as a safeguard against LLM-specific issues such as

hallucinations. Instead of generating arbitrary text, the models were constrained to select from instructor-defined outputs, increasing both the reliability of the system and the trustworthiness of the feedback delivered to students.

4.5 Evaluation Results

The evaluation of the framework was grounded in instructor-labeled data, which served as the reference point for measuring performance, as shown in Figure 4.2.

For each feedback instance, two criteria were assessed:

- **(i)** Whether the student had correctly applied the intended BPR principle in their process model.
- **(ii)** Whether the system-generated feedback accurately reflected correctness or error.

This dual evaluation allowed us to measure not just raw accuracy, but also the quality and reliability of the automated feedback mechanism.

Group ID	Rule1	Feedback1	Rule2	Feedback2	Rule3	Feedback3	Rule4	Feedback4	Rule5	Feedback5	Rule6	Feedback6	Rule7	Feedback7	Rule8	Feedback8	Rule9	Feedback9
1	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
2	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
3	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
4	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
5	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
6	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
7	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
8	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
9	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
10	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
11	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
12	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE
13	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
14	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE

Figure 4.2: Summary of the instructor-labeled comments for each rule and group.

We computed four standard classification metrics:

- **Precision:**
$$\text{Precision} = \frac{TP}{TP + FP}$$
Proportion of predicted positives that are correct.

- **Recall:**
$$\text{Recall} = \frac{TP}{TP + FN}$$
Proportion of actual positives correctly identified.

- **F1-score:**
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
Harmonic mean of precision and recall.

- **Accuracy:**
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
Overall proportion of correct predictions.

Where:

- TP = True Positive (correctly predicted correct feedback)
- FP = False Positive (incorrectly predicted correct feedback)
- TN = True Negative (correctly predicted incorrect feedback)
- FN = False Negative (incorrectly predicted incorrect feedback)

Table 4.1 provides the comparative results that highlight clear differences between the DMN-guided framework and the baseline Chain-of-Thought (CoT) prompting. Across all configurations, DMN-guided prompting delivered higher precision and more balanced performance.

- **GPT-4o with DMN** emerged as the most effective setup. It achieved a precision of 0.91, recall of 0.90, and F1-score of 0.91, alongside an overall accuracy of 0.87. This indicates that the system was both consistent in detecting correct applications of BPR rules and cautious enough to minimize false positives.

- **GPT-4o with CoT**, in contrast, produced perfect recall (1.00) but very poor precision (0.36). This means the system identified all correct cases but also over-generated feedback, leading to many false positives and lowering the F1-score to 0.53. In practical terms, this would risk overwhelming students with misleading or irrelevant comments.

- **Gemini 1.5 Pro** followed a similar pattern. Under DMN-guided prompting, it achieved an F1-score of 0.71, outperforming the CoT variant, which only reached 0.62. Although its overall performance was lower than GPT-4o, the consistency across models reinforces the generalizability of the DMN-based approach.

Approach	Model	Precision	Recall	F1-score	Accuracy
DMN-guided	GPT-4o	0.91	0.90	0.91	0.87
Chain-of-Thought	GPT-4o	0.36	1.00	0.53	0.54
DMN-guided	Gemini 1.5 Pro	0.60	0.87	0.71	0.65
Chain-of-Thought	Gemini 1.5 Pro	0.55	0.71	0.62	0.52

Table 4.1: Performance comparison of several methods of prompting.

These findings demonstrate the tangible benefits of embedding structured decision logic into prompt design. Whereas CoT prompting leans heavily on the probabilistic reasoning of LLMs—leading to variability and over-generation — the DMN framework keeps the evaluation aligned with unambiguous rules. This not only boosts precision but also ensures that feedback is traceable to explicit criteria, fostering both transparency and reliability.

Looking past numerical results, the results point to broader pedagogical implications. Higher precision means students are less likely to receive irrelevant or incorrect advice, which enhances trust in the feedback system. Balanced recall ensures that key errors are still detected, preventing important learning opportunities from being missed. Taken together, the results suggest that DMN-guided prompting provides a more disciplined and trustworthy foundation for deploying LLMs in educational contexts.

4.6 Rule-Specific Insights

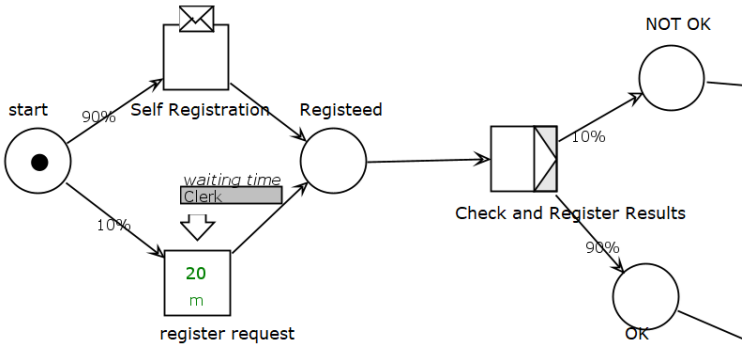

Figure 4.3: Example of a merged task with ambiguous labeling, leading to DMN-Guided framework using GPT-4o feedback misclassification for Rule 2 (task composition).

In addition to the overall performance metrics, a closer rule-level analysis was carried out to better understand how the framework handled individual Business Process Redesign (BPR) principles, as well as to uncover recurring challenges faced by students. This qualitative layer of analysis provided insights into both the strengths of the DMN-guided framework and the learning gaps present in the classroom.

For **Rule 1 (triage)**, all student groups successfully modeled the principle, and the framework—when paired with GPT-4o—identified these cases without error, demonstrating perfect alignment.

Rule 2 (task composition) was implemented correctly by roughly 79% of the groups, and the framework reached an accuracy of 86%. The main source of error came from unconventional task labels (e.g., "check and register result" instead of the expected phrasing related to a *credit* check), which occasionally reduced the system's ability to match the semantics of student models (see Figure 4.3).

For **Rule 3 (knock-out)**, which involves rejecting cases at an early stage (e.g., after a credit assessment), 93% of students applied the principle correctly. The framework matched this with 93% accuracy, slightly outperforming its overall average.

The most challenging principle for students was **Rule 5 (task resequencing)**. Only one group (7%) managed to correctly reorder tasks. Nevertheless, the framework was able to detect the intended redesign in 93% of submissions. This indicates that, although students found dependency reasoning difficult, the structured rules allowed the system to recognize the underlying logic reliably.

With **Rule 6 (deferred choice and contact reduction)**, 64% of groups succeeded in applying the principle, while the framework achieved 79% accuracy. Some groups incorrectly used XOR-splits or introduced unnecessary tasks such as an additional "Wait for Result" step, which contradicted the expected automated design. Figure 4.4 illustrates such a case, where students modeled both the reminder mechanism and a redundant waiting activity. Interestingly, the framework was still able to flag the issue correctly, most likely by drawing on contextual task assignments (e.g., roles such as "Fraud Investigator").

For **Rule 7 (parallelism between automated tasks)**, 71% of student groups modeled the principle correctly, while the framework achieved 79% accuracy. Misunderstandings often arose when students merged two external services into one task, which conflicted with proper process design assumptions.

Finally, **Rules 8 and 9 (parallelism of automated and manual tasks, and sequencing within parallelized tasks)** showed strong performance. Both the students and the framework achieved an accuracy rate of 86%, consistent with the overall evaluation results.

Taken together, these findings highlight that the DMN-guided framework is capable not only of detecting whether principles were correctly applied but also of revealing systematic areas where students struggled. This diagnostic capability gives instructors an additional benefit: beyond automated grading, the system can point to common misconceptions, guiding future teaching interventions.

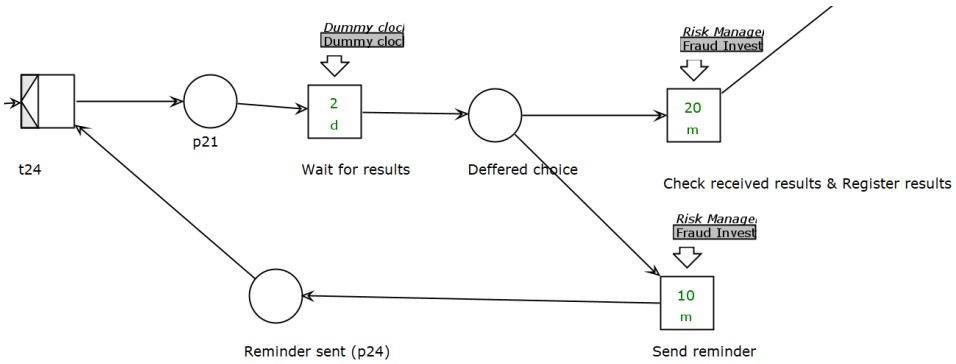

Figure 4.4: Example of an incorrectly modeled deferred choice, where an unnecessary 'Wait for Result' task was added despite the presence of an automated reminder mechanism.

4.7 Perceived Usefulness

To evaluate how students experienced the system in practice, we collected survey responses following the principles of the Technology Acceptance Model (TAM) [9]. TAM is widely used in information systems research to assess how useful and easy-to-use participants perceive a system to be [41, 42, 43]. Figure 4.5 presents the summarized results, both at the aggregate level and broken down by individual items.

At the overall level, the composite Perceived Usefulness (PU) score, calculated as the mean of six survey items, showed a strong positive trend. The median rating was consistently above 5.5 on the 7-point Likert scale, suggesting that most participants agreed or strongly agreed with the system's usefulness. The narrow spread of responses further indicates that opinions were concentrated in the upper part of the scale. Although a small number of outliers appeared (with ratings closer to 3), these did not substantially influence the overall distribution. Taken together, these results suggest that the system was generally regarded as highly useful by the majority of students.

Looking more closely at the individual PU items (PU1–PU6), all six questions exhibited similarly high central tendencies, with medians again clustering at or above 5.5. Some variation in dispersion was observed—for example, responses to PU5 displayed a slightly wider range—but the core pattern of positive evaluation held consistently across items. Even when outliers were present, the median values remained high, reflecting stable agreement with the usefulness of the system.

The alignment across different items, combined with the relatively low variability in responses, supports the internal consistency of the PU construct. This indicates that students viewed the system as useful across multiple dimensions of its functionality rather than in isolated aspects.

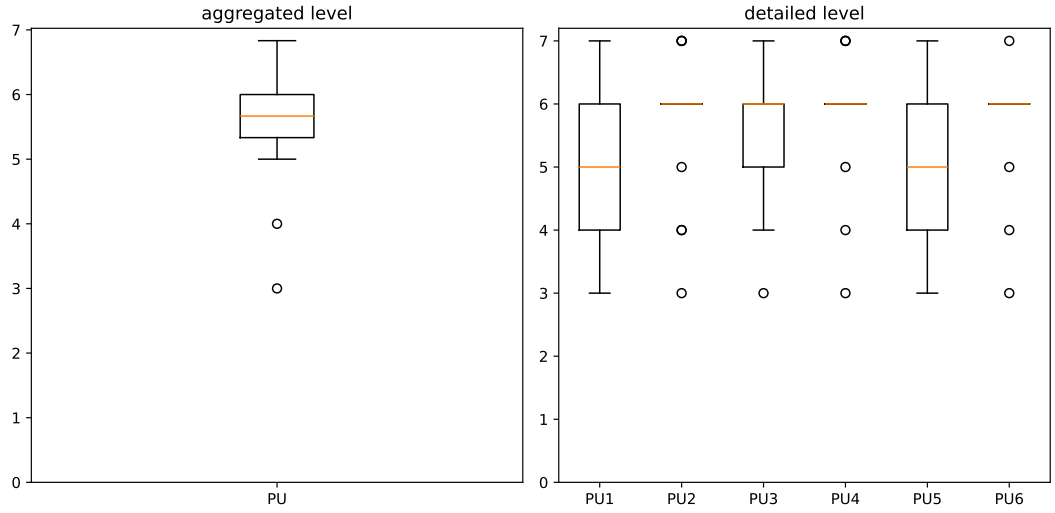

Figure 4.5: Boxplots showing the distribution of Perceived Usefulness (PU) in aggregated and detailed level.

4.7.1 Note on Perceived Ease of Use

The Perceived Ease of Use (PEU) dimension of TAM was not included in this study. Since students interacted with the system only indirectly through the university's Learning Management System (LMS), they did not operate the tool itself. As a result, evaluating ease of use was not relevant in this particular context.

4.8 Summary

The case study provides strong evidence that the DMN-guided prompting framework is both practical and effective within an educational setting. By embedding decision logic into the feedback process, the framework consistently outperformed the baseline Chain-of-Thought prompting approach, particularly in terms of accuracy and reliability. The structured approach enabled the generation of feedback that was not only precise but also transparent and reproducible, thereby addressing common concerns of inconsistency and unpredictability in LLM-based systems. From the perspective of students, the system was well received, as reflected in the high ratings of perceived usefulness.

At the same time, the study also showed areas for refinement. Some challenges were linked to ambiguous or unconventional task labels, which caused occasional misclassifications. More abstract design principles, such as task resequencing, proved difficult for both students to model and the framework to assess consistently, highlighting the need for enhanced support in these complex reasoning scenarios. These findings underscore that while formalized decision logic strengthens the reliability of feedback, continuous iteration is needed to fully capture the nuances of human-created process models.

Beyond technical performance, the case study illustrates the broader value of combining deterministic rule structures with the generative flexibility of large language

models. This hybrid approach allows systems to balance consistency with adaptability, offering both educators and learners feedback that is dependable yet sensitive to natural language variation. In practical terms, the framework fosters accountability, reproducibility, and scalability—qualities essential for integrating AI-based feedback into real-world classrooms.

In summary, the case study not only validates the feasibility of DMN-guided prompting but also outlines a pathway for its future improvement and adoption. The insights gained point toward opportunities for expanding the framework to other educational domains, refining its handling of ambiguous inputs, and leveraging its traceability for long-term use in teaching and assessment. Thus, the study demonstrates both the current strengths of the approach and its potential to evolve into a robust, widely applicable tool for AI-supported education.

Chapter 5

Discussion

5.1 Findings

This chapter moves beyond the case study results reported in Chapter 4 to consider their wider meaning and significance. While the previous chapter presented the empirical findings in detail—including performance metrics, rule-specific outcomes, and user perceptions—the current chapter interprets these results in light of both the theoretical motivations behind DMN-guided prompting and the broader field of Large Language Model (LLM) control strategies. In particular, it examines what the improvements in accuracy, precision, and perceived usefulness suggest about the role of structured decision logic in guiding probabilistic models.

The discussion also situates these findings within the wider landscape of related work, comparing DMN-guided prompting to alternative approaches such as Chain-of-Thought reasoning, template-based systems, and knowledge graph augmentation. By doing so, it highlights both the distinctive contributions of the framework and the challenges it shares with other hybrid AI methods.

In addition, this chapter explores the practical implications of the framework for education and beyond. Within education, the framework offers a pathway toward scalable, transparent, and reviewable feedback systems that can enhance teaching and learning. Outside of education, the same principles could be extended to domains such as healthcare, compliance, and customer service, where structured decision logic is central to ensuring trustworthy AI.

At the same time, it is important to recognize the framework's limitations. These include domain specificity, sensitivity to naming conventions, and the inherent constraints of LLMs when managing vague or incomplete inputs. The chapter, therefore, also outlines possible strategies to mitigate these issues, ranging from ontology integration to retrieval-augmented prompting and conversational clarification mechanisms.

Taken together, this discussion aims to situate the DMN-guided prompting framework not just as a technical innovation tested in a single classroom, but as part of a broader movement toward hybrid, interpretable, and responsible AI. It highlights both the current value demonstrated in the case study and the opportunities for

further refinement and application in diverse contexts.

5.2 Interpreting the Findings

5.2.1 Precision Gains and Trust

One of the clearest outcomes of the evaluation was the marked improvement in precision when moving from Chain-of-Thought (CoT) prompting to DMN-guided prompting. For example, GPT-4o's precision rose from 0.36 under CoT to 0.91 under DMN guidance. This shift represents more than a technical improvement: it directly affects the educational value of the system. In classroom contexts, a false positive—where the AI incorrectly flags or praises a student's work—can be more damaging than a missed detection. Misleading or inaccurate feedback risks confusing learners, reducing their trust in automated systems, and potentially slowing down their progress.

By reducing the rate of false positives, DMN-guided prompting produced feedback that students and instructors could rely on with greater confidence. These precision gains were not achieved by sacrificing recall. On the contrary, the F1-scores indicate that the framework preserved its ability to detect genuine errors while becoming more selective in what it reported. In practice, this balance means students receive feedback that is both targeted and accurate: The system points out mistakes when they happen, without overwhelming students with noise.

From a pedagogical perspective, this improvement strengthens the credibility of automated feedback. Trust is a critical factor in educational technology adoption, and systems that frequently generate incorrect advice can quickly lose student engagement. By contrast, a framework that consistently delivers precise and relevant feedback helps reinforce positive learning behaviors, since students are more likely to act on comments they believe to be valid. For instructors, higher precision also reduces the time needed for manual correction, as fewer AI-generated messages require rejection or revision. This trust-enhancing effect therefore benefits both learners and educators, positioning DMN-guided prompting as a more dependable foundation for classroom use.

These findings also connect to wider concerns about using AI, especially the challenge of ensuring reliability in high-stakes areas. Just as medical or legal decision-support systems must reduce the generation of invalid results, educational feedback systems must prioritize accuracy to maintain fairness and learner confidence. Within this context, the precision gains observed here highlight the value of combining probabilistic reasoning with deterministic rule structures, offering a model for trustworthy AI that extends beyond education.

5.2.2 Balanced Error Profiles

While Chain-of-Thought (CoT) prompting reached perfect recall in some settings, this caused an unacceptably high number of false positives. What this means is, the

system marked almost every issue, which ensured that real mistakes were detected, but it also overloaded students with irrelevant or incorrect feedback. Such over-generation not only makes the feedback less useful but also risks weakening students' trust in the system, since they may start ignoring even valid comments when these are hidden among errors. Over time, this can create disengagement, as learners come to see the AI not as a helpful assistant but as a source of noise that must be filtered out.

By contrast, the DMN-guided prompting framework delivered a more balanced error profile, combining high recall with a dramatic improvement in precision. This meant that most real errors were still detected, but without the distracting noise of unnecessary warnings. For students, this balance is vital: it ensures that meaningful mistakes are reliably highlighted while also protecting them from cognitive overload or unnecessary discouragement. Rather than receiving too many unhelpful alerts, students are provided with feedback that is targeted, pedagogically meaningful, and explicitly aligned with the principles they are expected to demonstrate. For instructors, the improved precision reduces the need to filter large volumes of incorrect system-generated advice, making the integration of AI feedback into the classroom more feasible and less resource-intensive.

More generally, this finding illustrates a central trade-off in AI-assisted evaluation systems: the tension between sensitivity (recall) and selectivity (precision). Systems that focus only on recall may seem thorough, but they often produce so much irrelevant feedback that their real usefulness is reduced. Conversely, systems optimized for precision alone may become overly conservative, failing to flag important mistakes. The results suggest that DMN-guided prompting offers a principled way to manage this trade-off, linking outputs to explicit decision rules that minimize false positives while still maintaining strong coverage of actual errors.

This balance has wider implications beyond education. In fields such as healthcare, compliance auditing, or legal review, excessive false positives may result in wasted resources, user fatigue, and the neglect of critical signals. False negatives, on the other hand, can cause critical oversights. The DMN-guided approach demonstrates that structured, rule-based reasoning can mitigate both extremes, making AI systems more trustworthy and practical in high-stakes domains.

From a pedagogical perspective, the ability to maintain both high recall and high precision also supports sustainable learning environments. When feedback is both accurate and concise, it strengthens student trust and confidence and keeps learners engaged, while also allowing instructors to rely on the system as real support for teaching. In this way, the balanced error profile of DMN-guided prompting is more than just a technical result—it is also essential for education, making it possible to bring AI into classrooms in a scalable and responsible way.

5.2.3 Rule-Level Insights

The rule-specific analysis (Chapter 4) revealed where the framework excelled and where challenges remain. Deterministic rules with clear, well-defined conditions, such as triage and knock-out, were detected with near-perfect accuracy. These results highlight the strength of the DMN-guided approach in contexts where evaluation criteria can be expressed clearly. In such cases, the framework operates almost as a direct mapping mechanism, with minimal chance of error.

By contrast, rules dependent on naming conventions, such as task composition, were less accurate. Small variations in student phrasing, for example, using "check and register result" instead of the more expected "credit check", lowered detection accuracy. This points to an important limitation: although DMN rules give structured guidance, their effectiveness is still influenced by how closely the input language matches the encoded conditions. Without some form of synonym handling or semantic normalization, even slight linguistic deviations can create mismatches.

More abstract redesign principles, such as resequencing or deferred choice, proved difficult for both students and the system. These principles often require reasoning about task order or conditional dependencies that extend beyond surface-level labeling. For example, correctly modeling deferred choice requires understanding not just the tasks involved but also the temporal and contextual assumptions governing them. Here, the DMN-guided framework still outperformed CoT prompting, but the errors show that abstraction and higher-order reasoning remain open challenges.

Taken together, these results illustrate a layered picture of system capability. At one end, deterministic rules map cleanly to structured feedback, demonstrating the reliability of the DMN approach. On the other hand, abstract or semantically variable rules expose the limitations of rule-based structures when students describe the same idea in different words. This duality suggests that future improvements should combine DMN with complementary techniques, such as lightweight ontologies, synonym dictionaries, or semantic parsing, to bridge the gap between deterministic logic and natural language variability.

Beyond technical implications, the rule-level insights also hold pedagogical value. They provide instructors with a diagnostic lens: where student errors were frequent (e.g., resequencing), the framework's results signal areas in which teaching interventions may be most needed. In this way, the system not only automates feedback but also acts as a mirror for student learning patterns, highlighting both conceptual strengths and recurring wrong ideas.

In this sense, the rule-level analysis shows how DMN-guided prompting contributes not only to technical accuracy but also to the creation of more transparent, reliable, and trustworthy AI systems in education. This naturally leads to the question of how students themselves experienced the system, which is examined in the following subsection on user perceptions.

5.2.4 User Perceptions

Survey results indicated strong student support for the system. Median scores above 5.5 across all TAM items reflect broad agreement that the framework added value. Students often viewed the rule-based feedback as more objective and trustworthy compared to free-form LLM outputs. This consistency is important for building confidence in AI-driven systems, aligning with broader findings in educational technology adoption that emphasize transparency and reliability as reasons for acceptance.

Beyond the numbers, the score distribution (see Fig. 4.5) showed little variation, meaning that most students shared the positive view instead of it coming only from a few very favorable responses. Even items with slightly broader spread, such as PU5, still maintained high median values, reinforcing the overall impression that students consistently valued the system. These stable ratings point to internal coherence in the way students evaluated usefulness: they saw the system as supportive across different dimensions of their learning, not just in isolated aspects.

Equally important was how students interpreted the nature of the feedback. Qualitative comments highlighted that predefined, rule-based messages felt more impartial and less arbitrary than the free-form responses sometimes associated with LLMs. In other words, students trusted the system not only because it was accurate but also because its outputs could be traced back to explicit decision logic. This sense of fairness is crucial in educational contexts, where perceived bias or inconsistency can quickly undermine acceptance of automated tools.

Another noteworthy implication is that student trust appeared closely linked to the transparency of the framework. Because the rules were aligned with principles they had been taught (e.g., triage, parallelism, resequencing), learners could easily connect feedback messages to the underlying course content. This created a perception that the system was reinforcing instruction rather than replacing it, so students perceived the AI as a transparent aid to their learning, instead of as an unclear authority whose decisions could not be questioned.

All in all, these findings suggest that the perceived usefulness of the DMN-guided framework stems not only from its technical accuracy but also from its interpretability and fairness. In line with prior TAM-based studies, acceptance is driven by a combination of effectiveness, clarity, and alignment with student expectations. As such, the system demonstrates how structured prompting approaches can go beyond error detection to cultivate student trust, which is a precondition for successful integration of AI feedback in real classrooms.

5.3 Comparison with Related Work

The DMN-guided framework is part of research on hybrid AI, which combines clear rule-based reasoning with flexible data-driven models. Like symbolic–statistical methods, it uses explicit rules for transparency while still taking advantage of what LLMs can generate. This helps address concerns about black-box AI by making

systems both easier to understand and more adaptable.

Compared with traditional intelligent tutoring systems (ITS), which often relied on static scripts or hand-coded rule trees, the DMN-based design introduces a modular and graphical representation that lowers the barrier for domain experts. This means that instructors can refine or extend evaluation logic without technical training, encouraging a more collaborative and reliable method for maintaining the system over time.

Prompt-engineering approaches such as Chain-of-Thought (CoT) or zero-shot prompting, while useful in many contexts, typically intertwine reasoning rules with natural language instructions. This not only makes them fragile when applied across different domains but also introduces significant overhead when prompts need to be adapted or debugged. By explicitly separating decision logic from prompt phrasing, DMN-guided prompting reduces this fragility and supports more consistent, domain-aligned outputs.

Related work on knowledge graph prompting and ontology-augmented reasoning also highlights the value of structured inputs for improving LLM performance. However, decision tables encoded in DMN offer a complementary advantage: They give a clear link between conditions and outcomes, making it easier to check, review, and match with teaching goals. This makes them particularly suitable for process-oriented domains where decision points can be formalized into explicit "if–then" structures.

Overall, the DMN-guided framework shows that adding a clear, rule-based structure to LLM prompting can improve control and interpretability, while still keeping the flexibility of modern models. While prior efforts have emphasized either the power of generative models or the rigidity of symbolic systems, this approach shows how the two can reinforce each other, providing a foundation for scalable and trustworthy applications in education and beyond.

5.4 Practical Implications

5.4.1 Educational Contexts

For instructors, the framework offers adaptability: decision rules can be updated by editing DMN models without re-engineering prompts. For students, transparency and consistency in feedback foster trust and reduce disputes about fairness. At an institutional level, scalability is supported by the ability to reuse the same framework across multiple courses, simply by substituting different DMN models.

In addition, the explicit mapping between decision rules and feedback provides a pedagogical advantage: students not only receive feedback but can also see the reasoning that underlies it. This reduces the "black-box" problem that often surrounds AI tools in education. Over time, such transparency can encourage students to reflect more deeply on evaluation criteria and develop self-assessment skills. For instructors, the approach reduces grading workload while still allowing them to retain control

over academic standards, since they define the decision logic. From an administrative perspective, the framework provides a uniform structure for documenting how assessments are carried out, which may support approval procedures or quality assurance reviews.

5.4.2 Beyond Education

The approach has clear potential outside of teaching:

- **Healthcare:** Clinical guidelines or diagnostic protocols could be modeled as DMN rules, offering explainable AI support.
- **Compliance:** Regulatory requirements could be encoded for consistent evaluation of documents and processes.
- **Customer Service:** Decision trees for troubleshooting could be embedded as DMN logic, improving the reliability of conversational agents.

Another strength is portability. Since the DMN standard is already widely adopted in industry, the framework can be integrated into existing workflows without requiring custom tools. Organizations already employing business rules or process modeling technologies could extend their infrastructure to AI-supported evaluation with minimal friction. Moreover, in domains where accountability is non-negotiable, such as healthcare or finance, the ability to point to explicit decision tables offers a safeguard against unclear or unverifiable AI outputs. This enhances stakeholder confidence and reduces risks of liability.

5.4.3 AI Governance

The framework also contributes to responsible AI practice. DMN rules provide an explicit and auditable record of decision logic. Transparency is enhanced because both educators and learners can inspect how decisions are derived. Finally, human-in-the-loop verification ensures that AI outputs remain aligned with domain expertise, preventing automation from replacing essential human oversight. These qualities resonate with current international discussions on trustworthy AI, such as the EU AI Act.

In governance terms, the framework operationalizes several principles that are often discussed in theory but rarely implemented in practice: traceability, contestability, and proportionality. Because every output can be traced back to a rule, decisions are contestable in a meaningful way—students, for instance, can query why a particular feedback message was given. Proportionality is supported because the system never generates open-ended judgments but instead selects from rule-constrained options. When viewed together, these features show how DMN-guided prompting can act as a test case for embedding accountability into real AI systems, bridging the gap between regulatory aspirations and applied practice.

5.5 Limitations

While the results of this study are promising, a number of limitations should be acknowledged to provide a realistic picture of the framework's current boundaries.

- **Domain Specificity:** The evaluation was confined to the context of business process redesign education. Although this setting is highly suitable due to its clear rules and established best practices, it remains an open question whether the same benefits will hold in other domains such as healthcare, law, or engineering. Broader testing is needed to establish cross-domain robustness.
- **Input Quality:** The effectiveness of the framework relies heavily on the clarity and structure of both the DMN models and the natural language descriptions of student work. Poorly specified decision rules or ambiguous textual inputs reduce accuracy, suggesting that instructor training and careful model design remain critical components.
- **Naming Sensitivity:** Rules tied to text matching remain vulnerable to inconsistencies in student task labels. For instance, semantically equivalent but differently phrased labels can lead to misclassification. While DMN provides structure, it does not fully solve the semantic variation problem inherent in natural language.
- **LLM Constraints:** Although DMN-guided prompting reduces hallucinations, large language models can still produce incorrect outputs when inputs are vague or under-specified. In such cases, the system may default to plausible but inaccurate reasoning, underlining the importance of human-in-the-loop review.
- **Scalability and Efficiency:** Very large or complex DMN models raise practical challenges. Current LLMs have finite context windows, meaning that feeding large decision tables into prompts risks exceeding token limits. Even when technically possible, efficiency concerns such as latency and computational cost may hinder deployment at scale.
- **Evaluation Scope:** The present study focused on immediate accuracy and user perceptions but did not capture long-term learning outcomes. It remains unclear whether exposure to structured, AI-assisted feedback leads to lasting improvements in student performance or conceptual understanding.
- **Generalizability of Results:** The participant pool was limited to one course at a single institution. Cultural, disciplinary, or institutional differences may shape how both instructors and students perceive and interact with AI-driven feedback, limiting the external validity of these findings.

These limitations highlight areas where further refinement and broader testing are required. Addressing them will be essential if DMN-guided prompting is to mature into a reliable and widely deployable solution across multiple educational and professional contexts.

5.6 Strategies for Mitigation

Several strategies could address the limitations identified above and support the maturation of the framework:

- **Synonym and Ontology Support:** Incorporating lightweight domain ontologies or synonym dictionaries would help normalize variation in task labels, ensuring that semantically equivalent terms (e.g., "credit verification" vs. "check and register result") are recognized consistently.
- **Incremental Domain Testing:** Expanding the framework to domains such as healthcare, legal reasoning, or compliance auditing would test adaptability under different knowledge structures and rule intensities. Such testing would also clarify how domain complexity affects both performance and user trust.
- **Context Window Optimization:** To address scalability, retrieval-augmented generation (RAG) techniques could be used to dynamically select and inject only the most relevant fragments of large DMN models into prompts, reducing the risk of exceeding context window limits while maintaining efficiency.
- **Interactive Clarification:** Allowing for multi-turn dialogue with the LLM could enable the system to request additional details when inputs are ambiguous or incomplete, reducing the risk of hallucinations or misinterpretations.
- **Instructor-Centered Tools:** Developing user-friendly authoring environments for DMN would reduce reliance on technical expertise and ensure that decision models are consistently well-structured. This would also encourage broader adoption by lowering entry barriers for educators.
- **Expanded Evaluation Metrics:** Beyond accuracy, future studies should measure long-term educational outcomes, student learning gains, and instructor workload reduction. This would provide a fuller picture of the framework's impact on teaching and learning.
- **Cross-Institutional Studies:** Conducting pilots in multiple courses and institutions would strengthen the generalizability of findings by accounting for cultural, disciplinary, and contextual differences in how AI-driven feedback is perceived.

Together, these strategies not only address current limitations but also set the stage for extending DMN-guided prompting into new domains and larger-scale deployments, ensuring that the framework remains adaptable, trustworthy, and relevant over time.

5.7 Summary

This chapter has reflected on the evaluation results by situating them within the broader discussion of LLM control strategies and hybrid AI design. The findings show

that DMN-guided prompting offers clear advantages over unconstrained methods such as Chain-of-Thought reasoning, particularly in terms of precision, balance between recall and selectivity, and overall trustworthiness. These gains translate directly into pedagogical value: students receive clearer, more reliable feedback, and instructors benefit from reduced noise and greater transparency in assessment.

At the same time, the analysis highlighted important nuances. Rule-level insights revealed that while deterministic conditions are handled with high accuracy, challenges remain when student phrasing diverges from expected labels or when abstract redesign principles require deeper reasoning. User perception data further emphasized that the usefulness of the system rests not only on technical accuracy but also on its transparency, fairness, and clear alignment with instructional goals.

When compared with related approaches, the framework demonstrates how explicit decision logic can bridge the gap between the interpretability of symbolic methods and the flexibility of modern LLMs. Its practical implications extend beyond education, offering potential value in domains where accountability and structured decision-making are central, such as healthcare, compliance, and customer service. At the same time, limitations such as domain specificity, naming sensitivity, and scalability must be carefully addressed.

Taken together, the discussion positions DMN-guided prompting as a step toward auditable, interpretable, and sustainable AI systems. By combining probabilistic language models with explicit rule structures, the framework not only advances the conversation on AI reliability but also provides a practical foundation for real-world deployment. The next chapter will build on these insights to outline future research directions, focusing on how the framework can be refined, extended, and adapted to broader contexts.

Chapter 6

Future Work

6.1 Foundational Concepts

6.1.1 Executive Summary

This report presents a series of research proposals designed to extend the DMN-guided prompting framework, which was developed to create more reliable, transparent, and modular automated feedback in educational settings. The original thesis demonstrated the framework's superior performance over conventional Chain-of-Thought (CoT) prompting, particularly in achieving higher precision and a more balanced error profile.

Building on these successes, this roadmap outlines six specific research directions to address the framework's identified limitations and advance it toward broader applicability and greater intelligence. The proposals focus on enhancing the system's robustness through automated self-refinement and semantic normalization, advancing its intelligence with retrieval-augmented and multi-turn capabilities, and establishing a formal governance model to ensure its ethical and sustainable deployment across diverse domains.

6.1.2 Rationale and Context

The most important contribution of the past research was the design, implementation, and evaluation of a framework that integrates Decision Model and Notation (DMN) with Large Language Models (LLMs). This approach responded to persistent challenges in automated feedback systems, including a lack of transparency, high maintenance costs, and limited reusability.

The DMN-guided framework successfully addressed these issues by externalizing evaluation logic into structured, easy-to-update DMN models, which then guided the LLM's reasoning. The case study, conducted in a graduate-level course, provided empirical evidence that this method improved both accuracy and consistency while receiving positive responses from students.

The purpose of this roadmap is to formalize and extend the high-level suggestions outlined in the *Future Work* chapter of the thesis. By transforming these ideas into

specific, actionable research projects, this document offers a practical roadmap for the next phase of development.

6.1.3 Acknowledged Thesis Strengths and Limitations

The DMN-guided framework's effectiveness was evidenced by its strong performance in a real-world case study. A comparison with a CoT baseline revealed a significant advantage in accuracy and reliability. Specifically, the framework paired with GPT-4o achieved an F1-score of 0.91, dramatically outperforming the CoT approach's 0.53.

However, the previous work also highlighted a number of limitations that serve as the foundation for future research:

- **Domain Specificity:** Validation was limited to one academic field.
- **Naming Sensitivity:** Challenges in handling varied phrasing.
- **Scalability and Efficiency:** Large DMN models risk exceeding LLM context windows.

These challenges are critical areas for the next phase of development, aiming to move from a prototype into a robust, generalizable solution.

6.2 Enhancing Framework Robustness and Adaptability

6.2.1 Research Direction: Automated DMN Model Refinement

6.2.1.1 Why We Propose This

The thesis used human-in-the-loop review to verify AI-generated feedback. This process, while effective, produced valuable labeled data of "DMN model errors." Future work can leverage this data to let the LLM propose corrections, shifting instructors into a supervisory role rather than manual maintainers.

6.2.1.2 Proposed System Architecture

- **Error Tagging Module:** Instructors label the reason for misclassifications (e.g., "Missing Rule," "Ambiguous Input").
- **LLM Analysis Engine:** An LLM analyzes tagged errors and proposes DMN rule changes.
- **Change Management Module:** Instructors review and approve AI-suggested updates.

This creates a closed-loop improvement cycle, reducing maintenance effort while retaining human oversight.

6.2.2 Research Direction: Integrating Ontology and Synonym Mapping

6.2.2.1 Reasoning Behind It

A key limitation was sensitivity to naming variation (e.g., "check and register result" vs. "credit check"). A semantic normalization layer would map varied student phrasings to uniform terms before DMN evaluation.

6.2.2.2 Proposed Methodology

- **Domain Ontology:** Define canonical terms and common synonyms. Example: "credit check," "check and register result," "verify credit" → all mapped to Credit_Verification.
- **Semantic Normalization Engine:** Preprocess natural language input to normalize phrasing.
- **Revised Evaluation Flow:** Apply DMN rules only to standardized inputs.

The Table 6.1 illustrates how varied student phrasings are mapped to standardized terms and then evaluated using DMN rules. For example, when a student writes "check and register result", the semantic normalization engine interprets it as Credit_Verification. The associated DMN rule then applies the knock-out principle, where the process can terminate if the credit check fails. Similarly, the phrase "delay until result" is standardized as Deferred_Choice, which requires that the decision be deferred only when involving non-manual tasks. Finally, "change the order of steps" is mapped to Task_Resequencing, and the system verifies that the reordering does not place a task before its required dependency.

Student Phrasing Examples	**Standard Term**	**DMN Rule Condition**
"check and register result"	Credit_Verification	Credit_Verification exists and leads to a knock-out
"delay until result"	Deferred_Choice	Deferred_Choice is used with a non-manual task
"change the order of steps"	Task_Resequencing	Task_Resequencing moves a step before its dependency

Table 6.1: Illustrative Canonical Term Mapping

This mapping ensures that even when students use different wordings, the framework consistently evaluates their models against the same underlying rules. It addresses the naming sensitivity problem by introducing a normalization layer that aligns natural language with deterministic logic, thereby improving both accuracy and fairness in automated feedback.

6.3 Advancing Framework Intelligence and User Interaction

6.3.1 Research Direction: Architecting a Retrieval-Augmented DMN Framework

6.3.1.1 Why This Matters

One of the main challenges of the current approach is scalability. Embedding an entire DMN model directly into a prompt quickly runs into the limits of an LLM's context window, especially when the decision model contains many rules. As the framework grows to support larger or more complex domains, this becomes increasingly impractical. Retrieval-Augmented Generation (RAG) offers a way forward by ensuring that only the most relevant decision rules are included in the prompt. This not only avoids overloading the model with unnecessary information but also makes the system faster and more efficient.

6.3.1.2 Proposed Architecture

To implement this idea, the framework could be redesigned around three main components:

- **DMN Repository:** All DMN rules are stored in a structured database (e.g., a vector database) where they can be efficiently indexed and searched.
- **Retrieval Engine:** When a new case or student submission is processed, the engine identifies and retrieves only the subset of rules that are directly relevant to that input.
- **Dynamic Prompt Assembly:** A concise prompt is then constructed by combining the student submission with the retrieved rules. This ensures that the LLM receives exactly the information it needs—no more, no less.

This architecture would allow the framework to handle thousands of rules without exceeding context limits, making it more scalable, efficient, and applicable in real-world settings where decision models can be very large.

6.3.2 Research Direction: Developing an Interactive Multi-Turn Feedback System

6.3.2.1 The case for this direction

The current framework processes inputs in a single pass. While efficient, this approach breaks down when faced with incomplete or ambiguous submissions. In such cases, the model either produces an incorrect result or fails to respond meaningfully. A dialogue-based system could overcome this limitation by allowing the AI to ask clarifying questions before applying decision rules. This interaction not only reduces

the risk of errors but also mirrors the way human instructors guide students through uncertainty, making the system both more accurate and more engaging.

6.3.2.2 Illustrative Dialogue Flow

Table 6.2 presents a simple example of how an interactive exchange might unfold. A student submits a process model with an ambiguous sequence. Instead of guessing, the AI requests clarification. Once the student provides the missing detail, the AI applies the appropriate DMN rule and delivers targeted feedback.

Turn	Role	Dialogue / Action
1	Student	Submits process model (input ambiguous).
2	AI	"Could you clarify the order of Task A and Task B?"
3	Student	"Yes, Task A must be completed before Task B."
4	AI	Applies Task_Resequencing rule: "Your model for Task B is incorrect. Task B requires Task A's output, so they cannot be parallel."

Table 6.2: Illustrative Dialogue Flow for Multi-Turn Clarification

This kind of exchange transforms the framework from a one-way feedback generator into an active learning companion. By engaging students in short clarification dialogues, the system ensures that feedback is not only technically correct but also pedagogically meaningful.

6.4 Generalizing and Sustaining the Framework

6.4.1 Cross-Domain Generalization and Evaluation

While the framework has shown promising results in the context of business process redesign education, its true potential lies in its adaptability to other fields. A structured plan for cross-domain generalization is therefore essential.

Phase 1: Domain Analysis and Model Formalization. The first step would involve close collaboration with domain experts to translate their knowledge into DMN-based rules. For example, in healthcare this might mean encoding diagnostic pathways, while in law it could involve formalizing compliance checks. Engaging experts early ensures that the resulting DMN models capture the nuances and priorities of each field.

Phase 2: Tool Development for Non-Technical Experts. A critical barrier to adoption is the technical expertise required to author DMN models. Building user-friendly authoring tools, such as graphical editors or guided templates, would allow instructors, clinicians, or auditors to contribute directly without needing programming skills. This democratization of model creation is essential for long-term sustainability.

Phase 3: Applied Case Studies. Finally, conducting real-world evaluations in domains such as healthcare, law, or compliance would test both technical robustness

and user acceptance. These studies would provide valuable insights into domain-specific challenges (e.g., medical terminology, legal ambiguity) and highlight how the framework can adapt across different professional contexts.

6.4.2 Comprehensive Long-Term Evaluation

Beyond domain expansion, it is equally important to study the framework's long-term impact. Accuracy metrics alone provide only a partial picture; sustainability requires measuring educational, organizational, and human factors over extended periods.

Quantitative Dimensions. Long-term studies should track objective indicators such as student performance, reduction in recurring errors, and instructor workload savings. In professional domains, equivalent measures could include compliance error reduction, time saved in audits, or accuracy of clinical decisions. These metrics would clarify whether the framework produces cumulative improvements over time, rather than short-term gains only.

Qualitative Dimensions. Equally important is capturing how users experience and perceive the system. Surveys based on established models such as TAM can quantify acceptance, while interviews and focus groups provide richer insights into usability, trust, and perceived fairness. Such methods also highlight unanticipated effects, such as whether students develop stronger self-assessment skills or whether professionals feel more confident in their decision-making.

Together, cross-domain generalization and long-term evaluation ensure that the DMN-guided framework does not remain a one-off academic prototype, but instead evolves into a mature, widely applicable, and sustainable approach to integrating structured decision logic with LLMs.

Metric	**Data Source**	**Frequency**
Student Grades	LMS records	End of each semester
DMN Log Accuracy	Framework logs	Monthly aggregation
Instructor Workload	Manual logs	Weekly
Perceived Usefulness	Survey (TAM)	End of semester
Qualitative Feedback	Interviews/Focus Groups	Mid- and end-semester

Table 6.3: Longitudinal Study Metrics and Data Collection Plan

6.5 Ethical Governance and Responsible AI Deployment

6.5.1 Proposed Governance Model

As AI systems increasingly influence decision-making in education and beyond, questions of ethics, transparency, and accountability cannot be overlooked. The DMN-guided framework naturally supports these principles, but a more explicit governance model strengthens its credibility and ensures compliance with emerging regulations such as the EU AI Act.

Transparency. All DMN models should be openly documented and, where possible, made available to stakeholders. This allows both students and instructors to see the exact decision logic behind each piece of feedback, helping to build trust and reducing the risk of "black box" decision-making.

Auditability. Governance requires that every system action can be traced. Version control of DMN models, combined with logs of feedback outputs, provides a complete audit trail. This enables institutions to verify why a particular decision was made and ensures accountability if errors or disputes arise.

Fairness. Even deterministic rules can unintentionally reflect bias. Regular review cycles should therefore be established to check DMN rules for unintended consequences, such as consistently disadvantaging certain groups of students or misrepresenting particular patterns of work.

Security and Privacy. Since both student data and institutional rules are sensitive, strong safeguards are necessary. Anonymization, encryption, and compliance with data protection protocols should be integral parts of the system design, ensuring responsible handling of personal and institutional information.

Together, these practices demonstrate how the framework can act not only as a technical solution but also as a model for responsible AI deployment in real-world contexts.

6.6 Conclusion and Research Roadmap

6.6.1 Synthesis of Directions

The research roadmap presented in this chapter outlines a coherent path forward for advancing DMN-guided prompting. Each proposed direction builds on the strengths of the current framework while addressing specific limitations identified in the case study.

First, automated DMN refinement and semantic normalization are essential to improving robustness. These strategies reduce the maintenance burden on instructors and mitigate the sensitivity to naming variation, ensuring that the system remains reliable over time.

Second, integrating retrieval-augmented methods and multi-turn interactions will expand scalability and intelligence. These enhancements allow the framework to handle larger and more complex models while engaging in meaningful dialogue with users when inputs are unclear.

Finally, embedding ethical governance into the framework ensures that future deployments are not only effective but also aligned with broader societal expectations. By combining transparency, auditability, fairness, and privacy, the system can serve as a test case for how interpretable decision models can guide LLMs responsibly.

Taken together, these directions provide a clear and practical roadmap for maturing the framework into a more powerful, trustworthy, and widely applicable solution.

Chapter 7

Conclusion

7.1 Introduction

This work set out to solve an important problem in applying Large Language Models (LLMs) to structured, high-stakes tasks: lack of clarity, poor maintainability, and limited reusability of systems where complex decision logic is embedded within unstructured prompts. Motivated by the limitations of this ad-hoc approach to prompt engineering, this research set out to design, implement, and evaluate a novel framework that integrates Decision Model and Notation (DMN) with LLM prompting. The objective was to introduce a more principled, low-code, and interpretable methodology for controlling LLM behavior. The work was grounded in a real-world educational case study focused on generating automated feedback for a graduate-level course on business process redesign, providing a practical testbed to validate the framework's effectiveness against a strong baseline.

7.2 Summary of Contributions

This research makes five primary contributions to the fields of applied AI, educational technology, and software engineering:

1. **A Novel Prompting Methodology:** The thesis introduces a new architecture where DMN serves as a transparent, externalized reasoning layer for LLMs. By separating decision logic from the wording of prompts, the framework allows domain experts to define, manage, and review evaluation rules without needing to wrestle with complex prompt engineering.

2. **A Modular, Triple-Based Architecture:** The framework encodes each decision as a self-contained "DMN triple" comprising inputs, a decision table, and literal expressions. This modular design enhances maintainability, reusability, and scalability, since individual rules can be modified without affecting the entire system.

3. **A Comprehensive Educational Case Study:** The framework was fully deployed and validated in a graduate-level course. This end-to-end implementa-

tion included data preprocessing from student submissions, automated prompt execution, and a rigorous human-in-the-loop verification process, demonstrating its practical viability.

4. **A Rigorous Comparative Evaluation:** Empirical evidence showed that DMN-guided prompting significantly outperforms Chain-of-Thought (CoT) prompting. The DMN-guided approach delivered dramatic improvements in precision, F1-score, and overall accuracy, demonstrating superior reliability and trustworthiness.

5. **An Open-Source Implementation:** To ensure reproducibility and encourage further research, the complete implementation—including the code, example DMN models, and workflows—has been released as an open-source artifact.

7.3 Answers to Research Questions

The findings of this thesis provide clear answers to the four research questions posed at the outset:

RQ1: How can DMN be integrated with LLMs to improve control, interpretability, and maintainability?

By encoding decision logic in modular DMN triples and instructing the LLM to systematically apply these rules, the framework externalizes the "brain" of the evaluation process. Instructors can update feedback criteria by editing DMN models without rewriting prompts, thereby improving transparency, control, and maintainability.

RQ2: How does DMN-guided prompting compare to CoT prompting in predictive accuracy?

In the case study, DMN-guided prompting paired with GPT-4o achieved a precision of 0.91 and an F1-score of 0.91. By contrast, the CoT baseline achieved perfect recall but with poor precision (0.36) and a much lower F1-score (0.53). DMN-guided prompting thus offered a more balanced and pedagogically sound error profile.

RQ3: How do end-users perceive the usefulness of a DMN-based reasoning system?

The TAM-based survey revealed high levels of perceived usefulness. Median scores above 5.5 indicated strong agreement that the system added value. Qualitative comments highlighted that rule-based feedback felt more objective, fair, and trustworthy than unconstrained AI outputs.

RQ4: What are the broader implications of using formal decision models to govern LLM outputs?

The results confirm that DMN-guided prompting supports principles of trustworthy AI by creating explicit, auditable trails for every decision. This "glass box" approach offers transparency and accountability and is well-suited for other high-stakes domains where oversight and explainability are essential.

7.4 Implications

The findings carry significant implications:

- **Educational Technology:** Provides a scalable model for consistent, rule-aligned automated feedback. Transparency fosters student trust, while instructors retain control over evaluation criteria.
- **AI Governance:** Shows a clear way forward for designing AI systems. DMN models serve as decision logs that can be inspected for compliance and fairness, aligning with regulatory principles.
- **Cross-Domain Deployment:** The methodology is transferable to domains such as healthcare (clinical decision support), finance (compliance auditing), and customer service (policy enforcement).

7.5 Limitations

Despite promising results, several limitations remain:

- **Domain Specificity:** The framework was validated in one academic context (business process redesign); generalization to other domains remains untested.
- **Naming Sensitivity:** Performance is affected by variations in task naming and phrasing, requiring greater semantic flexibility.
- **LLM Context Constraints:** Current context limits restrict the size and complexity of DMN models that can be processed in a single prompt.

7.6 Future Outlook

As outlined in Chapter 6, these limitations inform clear directions for future work:

- **Domain Adaptation:** Extending the framework to healthcare, law, and compliance.
- **Integration with RAG:** Using retrieval-augmented generation to handle large and complex DMN models.

- **Multi-Turn Interaction:** Developing a dialogue-based agent that asks clarifying questions when faced with ambiguity.
- **Ontology Support:** Building semantic normalization layers to reduce errors caused by naming mismatches.

7.7 Closing Remarks

This thesis has shown that DMN-guided prompting is a viable and effective way to control LLM behavior in structured decision-making. By leveraging a standardized and interpretable modeling language, it bridges the gap between human expertise and AI execution, enabling more transparent, collaborative, and consistent feedback.

In a landscape where AI must be explainable, auditable, and aligned with human values, this study signals a transition from ad-hoc prompting practices to a more systematic and principled framework for AI control. It provides both a technical foundation and a practical blueprint for ensuring that LLMs act not only intelligently but also responsibly.

Bibliography

[1] *Decision Model and Notation Specification*. Tech. rep. 1.6. Object Management Group - Standards Development Organization, 2024. URL: https://www.omg.org/spec/DMN (cit. on pp. 1, 3, 4, 13).

[2] Ekaterina Bazhenova, Francesca Zerbato, Barbara Oliboni, and Mathias Weske. "From BPMN process models to DMN decision models". In: *Information Systems* 83 (2019), pp. 69–88. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2019.02.001 (cit. on p. 1).

[3] Steven Mertens, Frederik Gailly, and Geert Poels. "Enhancing Declarative Process Models with DMN Decision Logic". In: *Enterprise, Business-Process and Information Systems Modeling*. Ed. by Khaled Gaaloul, Rainer Schmidt, Selmin Nurcan, Sérgio Guerreiro, and Qin Ma. Cham: Springer International Publishing, 2015, pp. 151–165. ISBN: 978-3-319-19237-6 (cit. on p. 1).

[4] Shahrzad Khayatbashi, Viktor Sjölind, Anders Granåker, and Amin Jalali. *AI-Enhanced Business Process Automation: A Case Study in the Insurance Domain Using Object-Centric Process Mining*. Accepted in BPMDS 2025. Springer, 2025 (cit. on p. 3).

[5] Michael Parker, Caitlin Anderson, Claire Stone, and YeaRim Oh. "A Large Language Model Approach to Educational Survey Feedback Analysis". In: *International Journal of Artificial Intelligence in Education* (June 2024). DOI: 10.1007/s40593-024-00414-0 (cit. on p. 3).

[6] Kurt Sandkuhl. "LLM-Assistance for Quality Control of LLM Output". In: *Perspectives in Business Informatics Research*. Ed. by Václav Řepa, Raimundas Matulevičius, and Emanuele Laurenzi. Cham: Springer Nature Switzerland, 2024, pp. 36–50. ISBN: 978-3-031-71333-0 (cit. on p. 3).

[7] Massimiliano De Leoni, Paolo Felli, and Marco Montali. "Integrating BPMN and DMN: modeling and analysis". In: *Journal on Data Semantics* 10.1 (2021), pp. 165–188 (cit. on p. 4).

[8] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. "Chain-of-thought prompting elicits reasoning in large language models". In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NIPS '22. New Orleans, LA, USA: Curran Associates Inc., 2022. ISBN: 9781713871088 (cit. on pp. 4, 11, 36).

[9] Fred D Davis. "Perceived usefulness, perceived ease of use, and user acceptance of information technology". In: *MIS quarterly* (1989), pp. 319–340 (cit. on pp. 4, 42).

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762 (cit. on p. 7).

[11] Yilun Zhu, Joel Ruben Antony Moniz, Shruti Bhargava, Jiarui Lu, Dhivya Piraviperumal, Site Li, Yuan Zhang, Hong Yu, and Bo-Hsiang Tseng. *Can Large Language Models Understand Context?* 2024. arXiv: 2402.00858 [cs.CL]. URL: https://arxiv.org/abs/2402.00858 (cit. on p. 7).

[12] Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. "The Curious Case of Hallucinations in Neural Machine Translation". In: Jan. 2021, pp. 1172–1183. DOI: 10.18653/v1/2021.naacl-main.92 (cit. on p. 8).

[13] Feng Nie, Meixi Chen, Zhirui Zhang, and Xu Cheng. *Improving Few-Shot Performance of Language Models via Nearest Neighbor Calibration*. 2022. arXiv: 2212.02216 [cs.CL]. URL: https://arxiv.org/abs/2212.02216 (cit. on p. 8).

[14] Guoheng Sun et al. *CoIn: Counting the Invisible Reasoning Tokens in Commercial Opaque LLM APIs*. 2025. arXiv: 2505.13778 [cs.AI]. URL: https://arxiv.org/abs/2505.13778 (cit. on p. 8).

[15] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. "Prompt Engineering in Large Language Models". In: *Data Intelligence and Cognitive Informatics*. Ed. by I. Jeena Jacob, Selwyn Piramuthu, and Przemyslaw Falkowski-Gilski. Singapore: Springer Nature Singapore, 2024, pp. 387–402. ISBN: 978-981-99-7962-2 (cit. on p. 8).

[16] Sriram Ramanathan, Lisa-Angelique Lim, Nazanin Rezazadeh Mottaghi, and Simon Buckingham Shum. "When the Prompt becomes the Codebook: Grounded Prompt Engineering (GROPROE) and its application to Belonging Analytics". In: *Proceedings of the 15th International Learning Analytics and Knowledge Conference*. LAK '25. New York, NY, USA: Association for Computing Machinery, 2025, pp. 713–725. ISBN: 9798400707018. DOI: 10.1145/3706468.3706564 (cit. on p. 8).

[17] Victor Sanh et al. *Multitask Prompted Training Enables Zero-Shot Task Generalization*. 2022. arXiv: 2110.08207 [cs.LG]. URL: https://arxiv.org/abs/2110.08207 (cit. on p. 8).

[18] Ethan Perez, Douwe Kiela, and Kyunghyun Cho. *True Few-Shot Learning with Language Models*. 2021. arXiv: 2105.11447 [cs.CL]. URL: https://arxiv.org/abs/2105.11447 (cit. on p. 10).

[19] Houda Louatouate and Mohammed Zeriouh. "Role-based Prompting Technique in Generative AI-Assisted Learning: A Student-Centered Quasi-Experimental Study". In: *Journal of Computer Science and Technology Studies* 7 (Apr. 2025), pp. 130–145. DOI: 10.32996/jcsts.2025.7.2.12 (cit. on p. 12).

[20] Jian Lang, Zhangtao Cheng, Ting Zhong, and Fan Zhou. *Retrieval-Augmented Dynamic Prompt Tuning for Incomplete Multimodal Learning*. 2025. arXiv: 2501.01120 [cs.CV]. URL: https://arxiv.org/abs/2501.01120 (cit. on p. 13).

[21] Chinimilli Venkata Rama Padmaja and S Lakshminarayana. "Enhancing Language Models Through Prompt Engineering - A Survey". In: *2024 IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT)*. 2024, pp. 117–121. DOI: 10.1109/ICISSGT58904.2024.00033 (cit. on p. 13).

[22] Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. "Towards Revealing the Mystery behind Chain of Thought: A Theoretical Perspective". In: *Advances in Neural Information Processing Systems*. Vol. 36. Curran Associates, Inc., 2023, pp. 70757–70798. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/dfc310e81992d2e4cedc09ac47eff13e-Paper-Conference.pdf (cit. on p. 13).

[23] Matthijs Berkhout, Koen Smit, and Johan Versendaal. "Decision discovery using clinical decision support system decision log data for supporting the nurse decision-making process". In: *BMC Medical Informatics and Decision Making* 24 (Apr. 2024). DOI: 10.1186/s12911-024-02486-3 (cit. on p. 15).

[24] Neel Guha et al. *LegalBench: A Collaboratively Built Benchmark for Measuring Legal Reasoning in Large Language Models*. 2023. arXiv: 2308.11462 [cs.CL]. URL: https://arxiv.org/abs/2308.11462 (cit. on p. 15).

[25] Aleksei Makin. *Ontology-Driven Knowledge Management Systems Enhanced by Large Language Models*. Dec. 2024. DOI: 10.13140/RG.2.2.24648.23043 (cit. on p. 16).

[26] Chinimilli Venkata Rama Padmaja and S Lakshminarayana. "Enhancing Language Models Through Prompt Engineering - A Survey". In: *2024 IEEE International Conference on Intelligent Systems, Smart and Green Technologies (ICISSGT)*. 2024, pp. 117–121. DOI: 10.1109/ICISSGT58904.2024.00033 (cit. on p. 16).

[27] Karthik Valmeekam, Sarath Sreedharan, Matthew Marquez, Alberto Olmo, and Subbarao Kambhampati. *On the Planning Abilities of Large Language Models (A Critical Investigation with a Proposed Benchmark)*. 2023. arXiv: 2302.06706 [cs.AI]. URL: https://arxiv.org/abs/2302.06706 (cit. on p. 16).

[28] Marie Stevenson. “Book review Shermis, M.D. & Burstein, J. (Eds) (2013). Handbook of Automated Essay Evaluation: Current applications and new directions. Routledge: New York and London. ISBN-10: 0415810965”. In: *Journal of Writing Research* 5 (Oct. 2013), pp. 239–243. DOI: 10.17239/jowr-2013.05.02.4 (cit. on p. 17).

[29] Richard Lobb and Jenny Harlow. “Coderunner: a tool for assessing computer programming skills”. In: *ACM Inroads* 7.1 (Feb. 2016), pp. 47–51. ISSN: 2153-2184. DOI: 10.1145/2810041. URL: https://doi.org/10.1145/2810041 (cit. on p. 17).

[30] Elisabeth Bauer, Martin Greisel, Ilia Kuznetsov, Markus Berndt, Ingo Kollar, Markus Dresel, Martin Fischer, and Frank Fischer. “Using natural language processing to support peer-feedback in the age of artificial intelligence: A cross-disciplinary framework and a research agenda”. In: *British Journal of Educational Technology* 54 (May 2023), pp. 1222–1245. DOI: 10.1111/bjet.13336 (cit. on p. 17).

[31] Qinjin Jia, Jialin Cui, Haoze Du, Parvez Rashid, Ruijie Xi, Ruochi Li, and Edward Gehringer. “LLM-generated Feedback in Real Classes and Beyond: Perspectives from Students and Instructors”. In: *Proceedings of the 17th International Conference on Educational Data Mining*. Ed. by Benjamin PaaÃŸen and Carrie Demmans Epp. Atlanta, Georgia, USA: International Educational Data Mining Society, July 2024, pp. 862–867. ISBN: 978-1-7336736-5-5. DOI: 10.5281/zenodo.12729974 (cit. on p. 18).

[32] Wei Dai, Jionghao Lin, Hua Jin, Tongguang Li, Yi-Shan Tsai, Dragan Gasevic, and Guanliang Chen. “Can Large Language Models Provide Feedback to Students? A Case Study on ChatGPT”. In: July 2023, pp. 323–325. DOI: 10.1109/ICALT58122.2023.00100 (cit. on p. 18).

[33] Joel Eapen and Adhithyan V S. “Personalization and Customization of LLM Responses”. In: *International Journal of Research Publication and Reviews* 4 (Dec. 2023), pp. 2617–2627. DOI: 10.55248/gengpi.4.1223.123512 (cit. on p. 18).

[34] Jussi Jauhiainen and Agustin Garagorry. “Evaluating Students’ Open-ended Written Responses with LLMs: Using the RAG Framework for GPT-3.5, GPT-4, Claude-3, and Mistral-Large”. In: (May 2024). DOI: 10.48550/arXiv.2405.05444 (cit. on p. 18).

[35] Chang Lu and Maria Cutumisu. “Integrating Deep Learning into An Automated Feedback Generation System for Automated Essay Scoring”. In: Oct. 2021 (cit. on p. 18).

[36] Gloria Ashiya Katuka, Alexander Gain, and Yen-Yun Yu. *Investigating Automatic Scoring and Feedback using Large Language Models*. 2024. arXiv: 2405.00602 [cs.CL]. URL: https://arxiv.org/abs/2405.00602 (cit. on pp. 18, 19).

[37] Kathrin Seßler, Arne Bewersdorff, Claudia Nerdel, and Enkelejda Kasneci. *Towards Adaptive Feedback with AI: Comparing the Feedback Quality of LLMs and Teachers on Experimentation Protocols*. Feb. 2025. DOI: 10.48550/arXiv.2502.12842 (cit. on p. 19).

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903 (cit. on p. 23).

[39] Thomas Freytag. "Woped–workflow petri net designer". In: *University of Cooperative Education* (2005), pp. 279–282 (cit. on p. 34).

[40] Amin Jalali. "Teaching business process development through experience-based learning and agile principle". In: *Perspectives in Business Informatics Research: 17th International Conference, BIR 2018, Stockholm, Sweden, September 24-26, 2018, Proceedings 17*. Springer. 2018, pp. 250–265 (cit. on p. 34).

[41] Amin Jalali. "Evaluating perceived usefulness and ease of use of cmmn and dcr". In: *International conference on business process modeling, development and support*. Springer. 2021, pp. 147–162 (cit. on p. 42).

[42] Amin Jalali. "Evaluating user acceptance of knowledge-intensive business process modeling languages". In: *Software and Systems Modeling* 22.6 (2023), pp. 1803–1826 (cit. on p. 42).

[43] Amin Jalali, Fabrizio Maria Maggi, and Hajo A Reijers. "A hybrid approach for aspect-oriented business process modeling". In: *Journal of Software: Evolution and process* 30.8 (2018), e1931 (cit. on p. 42).