# 9.3.1 DynamicPromptEnhancer Feature

## Feature Goal

To provide an intelligent, context-aware system for dynamically enhancing AI prompts based on user context, question characteristics, and philosophical tone requirements, ensuring personalized, high-quality responses across all app features while maintaining consistent philosophical voices.

## System Architecture

### Frontend Components

- \*\*PromptEnhancementProvider.tsx\*\* - Context provider for prompt enhancement services

- \*\*TonePreviewGenerator.tsx\*\* - Component for generating tone-specific preview responses

- \*\*ContextualPromptBuilder.tsx\*\* - Component for building context-aware prompts

- \*\*UserContextCollector.tsx\*\* - Component for gathering user context data

- \*\*QuestionAnalyzer.tsx\*\* - Component for analyzing question characteristics

- \*\*ConceptIntegrator.tsx\*\* - Component for integrating concept knowledge into prompts

- \*\*PromptTemplateRenderer.tsx\*\* - Component for rendering prompt templates with dynamic data

- \*\*EnhancementSettingsPanel.tsx\*\* - Admin panel for configuring enhancement rules

- \*\*PromptDebugConsole.tsx\*\* - Developer tool for inspecting prompt construction

- \*\*EnhancementMetricsDisplay.tsx\*\* - Component for displaying prompt enhancement metrics

### Backend Components

- \*\*dynamic\_prompt\_service.py\*\* - Core service for prompt enhancement orchestration

- \*\*tone\_template\_manager.py\*\* - Service for managing philosophical tone templates

- \*\*context\_analyzer.py\*\* - Service for analyzing user context and history

- \*\*question\_classifier.py\*\* - Service for classifying and analyzing questions

- \*\*concept\_integrator.py\*\* - Service for integrating relevant concepts into prompts

- \*\*user\_level\_adapter.py\*\* - Service for adapting prompts based on user expertise level

- \*\*prompt\_metrics\_service.py\*\* - Service for tracking prompt enhancement effectiveness

- \*\*template\_renderer.py\*\* - Service for rendering prompt templates with dynamic data

- \*\*enhancement\_rules\_service.py\*\* - Service for managing enhancement rule configurations

- \*\*prompt\_cache\_service.py\*\* - Service for caching and retrieving enhanced prompts

### Database Models

```sql

-- Stores prompt templates for different philosophical tones

CREATE TABLE prompt\_templates (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

tone\_id VARCHAR(50) NOT NULL REFERENCES philosophical\_tones(id),

model\_id VARCHAR(50) NOT NULL,

template\_content TEXT NOT NULL,

version INTEGER NOT NULL DEFAULT 1,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

created\_by UUID REFERENCES users(id),

CONSTRAINT unique\_tone\_model\_version UNIQUE (tone\_id, model\_id, version)

);

-- Stores enhancement rules for dynamic prompt construction

CREATE TABLE enhancement\_rules (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

rule\_name VARCHAR(100) NOT NULL,

rule\_description TEXT,

rule\_type VARCHAR(50) NOT NULL,

rule\_condition JSONB NOT NULL,

rule\_action JSONB NOT NULL,

priority INTEGER NOT NULL DEFAULT 100,

is\_active BOOLEAN DEFAULT TRUE,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

CONSTRAINT valid\_priority CHECK (priority BETWEEN 1 AND 1000)

);

-- Stores prompt enhancement metrics

CREATE TABLE prompt\_enhancement\_metrics (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

interaction\_id UUID REFERENCES ask\_interactions(id),

original\_prompt\_length INTEGER NOT NULL,

enhanced\_prompt\_length INTEGER NOT NULL,

enhancement\_types JSONB NOT NULL,

user\_context\_factors JSONB,

question\_factors JSONB,

response\_quality\_score FLOAT,

processing\_time\_ms INTEGER NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- Stores cached enhanced prompts

CREATE TABLE prompt\_cache (

cache\_key VARCHAR(255) PRIMARY KEY,

prompt\_content TEXT NOT NULL,

tone\_id VARCHAR(50) NOT NULL REFERENCES philosophical\_tones(id),

model\_id VARCHAR(50) NOT NULL,

context\_hash VARCHAR(64) NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

expires\_at TIMESTAMP WITH TIME ZONE NOT NULL,

hit\_count INTEGER DEFAULT 1

);

-- Stores user-specific prompt customizations

CREATE TABLE user\_prompt\_preferences (

user\_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,

preference\_type VARCHAR(50) NOT NULL,

preference\_value JSONB NOT NULL,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

PRIMARY KEY (user\_id, preference\_type)

);

```

## API Endpoints

### Enhance Prompt

```

POST /api/v1/prompts/enhance

```

Request:

```json

{

"base\_prompt": "Respond to the user's philosophical question",

"tone\_id": "socratic",

"model\_id": "claude",

"question": "What is the nature of consciousness?",

"user\_context": {

"user\_level": 7,

"mastered\_concepts": ["dualism", "materialism", "qualia"],

"recent\_interactions": ["mind\_body\_problem", "free\_will"],

"preferred\_complexity": "medium"

},

"enhancement\_types": ["tone", "context", "concepts", "complexity"]

}

```

Response:

```json

{

"enhanced\_prompt": "You are embodying the Socratic method of philosophical inquiry. Your role is to guide the user toward deeper understanding through thoughtful questioning rather than providing direct answers. The user has previously explored the mind-body problem and free will, and has demonstrated understanding of dualism, materialism, and qualia concepts. Respond to their question about the nature of consciousness by asking guiding questions that build on their existing knowledge while introducing new perspectives. Use medium complexity language appropriate for their level 7 understanding.",

"enhancement\_metrics": {

"original\_length": 45,

"enhanced\_length": 387,

"enhancement\_types\_applied": ["tone", "context", "concepts", "complexity"],

"processing\_time\_ms": 78

},

"template\_id": "550e8400-e29b-41d4-a716-446655440000"

}

```

### Get Tone Templates

```

GET /api/v1/prompts/templates/tones

```

Response:

```json

{

"templates": [

{

"id": "550e8400-e29b-41d4-a716-446655440001",

"tone\_id": "socratic",

"model\_id": "claude",

"version": 2,

"is\_active": true,

"created\_at": "2025-05-15T14:30:00Z",

"updated\_at": "2025-05-20T09:15:00Z"

},

{

"id": "550e8400-e29b-41d4-a716-446655440002",

"tone\_id": "pragmatic",

"model\_id": "grok",

"version": 1,

"is\_active": true,

"created\_at": "2025-05-10T11:45:00Z",

"updated\_at": "2025-05-10T11:45:00Z"

}

],

"total\_count": 40

}

```

### Get Template Content

```

GET /api/v1/prompts/templates/{template\_id}

```

Response:

```json

{

"id": "550e8400-e29b-41d4-a716-446655440001",

"tone\_id": "socratic",

"model\_id": "claude",

"template\_content": "You are embodying the Socratic method of philosophical inquiry. Your role is to guide the user toward deeper understanding through thoughtful questioning rather than providing direct answers. Use Socrates' approach of claiming to know nothing while helping others discover knowledge through careful examination of their beliefs.\n\nClaude-specific optimizations:\n- Structure your questions logically and progressively\n- Identify and examine underlying assumptions methodically\n- Use precise philosophical terminology\n- Build arguments step-by-step through questioning\n\n{{user\_context}}\n\nUser Question: {{question}}\n\nPlease respond:",

"version": 2,

"is\_active": true,

"created\_at": "2025-05-15T14:30:00Z",

"updated\_at": "2025-05-20T09:15:00Z",

"created\_by": "550e8400-e29b-41d4-a716-446655440003"

}

```

### Create Enhancement Rule

```

POST /api/v1/prompts/rules

```

Request:

```json

{

"rule\_name": "Advanced Concept Integration",

"rule\_description": "Integrates advanced philosophical concepts when user level is 10+",

"rule\_type": "concept\_integration",

"rule\_condition": {

"user\_level\_min": 10,

"has\_concept\_mastery": true

},

"rule\_action": {

"template": "Include references to advanced concepts like {{concepts}} in your response, assuming the user has a sophisticated understanding of these topics.",

"priority": 200

}

}

```

Response:

```json

{

"id": "550e8400-e29b-41d4-a716-446655440004",

"rule\_name": "Advanced Concept Integration",

"rule\_description": "Integrates advanced philosophical concepts when user level is 10+",

"rule\_type": "concept\_integration",

"rule\_condition": {

"user\_level\_min": 10,

"has\_concept\_mastery": true

},

"rule\_action": {

"template": "Include references to advanced concepts like {{concepts}} in your response, assuming the user has a sophisticated understanding of these topics.",

"priority": 200

},

"is\_active": true,

"created\_at": "2025-06-05T15:30:00Z"

}

```

### Get Enhancement Metrics

```

GET /api/v1/prompts/metrics

```

Response:

```json

{

"metrics": {

"average\_enhancement\_ratio": 4.7,

"most\_used\_enhancement\_types": [

{"type": "tone", "count": 12500},

{"type": "context", "count": 9800},

{"type": "concepts", "count": 7200},

{"type": "complexity", "count": 5100}

],

"average\_processing\_time\_ms": 85,

"cache\_hit\_rate": 0.65,

"enhancement\_impact\_on\_response\_quality": 0.82

},

"time\_period": "last\_30\_days"

}

```

## User Experience Flow

1. \*\*Initialization & Configuration\*\*

- System loads prompt templates for all philosophical tones

- Enhancement rules are loaded and prioritized

- User context data is initialized from profile and history

2. \*\*Question Submission\*\*

- User submits philosophical question via Ask feature

- Question is analyzed for complexity, topic, and philosophical domain

- User's expertise level and concept mastery are retrieved

3. \*\*Prompt Enhancement Process\*\*

- Base prompt template is selected based on tone and AI model

- Dynamic enhancement begins with sequential rule application:

- Tone-specific instructions are applied

- User context factors are integrated

- Question-specific adaptations are made

- Concept references are incorporated

- Complexity level is adjusted

- Enhanced prompt is validated for length and structure

4. \*\*AI Model Interaction\*\*

- Enhanced prompt is sent to selected AI model via AI Router

- Response is generated with consistent philosophical tone

- Response quality metrics are collected

5. \*\*Continuous Improvement\*\*

- Enhancement effectiveness is measured through:

- User ratings via ContemplativeOrb

- Response quality analysis

- Engagement metrics

- Templates and rules are periodically refined based on data

- User-specific customizations are learned over time

## Advanced Enhancement Techniques

1. \*\*Progressive Complexity Adaptation\*\*

- Analyzes user's historical interactions to determine optimal complexity

- Gradually increases complexity as user demonstrates mastery

- Adjusts language sophistication based on user's philosophical vocabulary

- Example enhancement: "The user has demonstrated understanding of phenomenology concepts. Use appropriate terminology like 'intentionality' and 'noema' without defining these terms."

2. \*\*Concept Mastery Integration\*\*

- Identifies concepts the user has already mastered

- Incorporates references to these concepts in new contexts

- Builds bridges between known concepts and new philosophical areas

- Example enhancement: "The user has mastered concepts of existentialism. Reference how existentialist perspectives on freedom relate to their current question about ethics."

3. \*\*Contextual Continuity\*\*

- Maintains conversation context across multiple interactions

- References previous questions and insights when relevant

- Creates coherent philosophical journeys rather than isolated exchanges

- Example enhancement: "This is a follow-up to the user's previous question about free will. Reference the compatibilist perspective discussed earlier while addressing this new question about moral responsibility."

4. \*\*Personalized Tone Calibration\*\*

- Adjusts philosophical tone based on user preferences and engagement

- Fine-tunes tone characteristics for individual users

- Learns which aspects of each tone resonate with specific users

- Example enhancement: "This user responds well to the questioning aspects of the Socratic tone but prefers more concrete examples. Maintain the questioning approach while incorporating practical scenarios."

## Implementation Considerations

- \*\*Template Management System\*\*

- Version-controlled prompt templates for each tone-model combination

- Admin interface for template editing and testing

- A/B testing framework for template optimization

- Template validation to ensure compatibility with AI models

- \*\*Rule-Based Enhancement Engine\*\*

- Configurable rules with condition-action pairs

- Priority-based rule execution

- Rule effectiveness tracking

- Visual rule builder for non-technical administrators

- \*\*Performance Optimization\*\*

- Intelligent caching of enhanced prompts for similar contexts

- Asynchronous pre-enhancement for likely next questions

- Batch processing of enhancement rules

- Optimized template rendering with minimal string operations

- \*\*Integration Points\*\*

- Seamless integration with AI Router (ai\_router.py)

- Hook into Ask feature workflow

- Integration with Concepts system for knowledge incorporation

- User profile system integration for context data

- \*\*Analytics & Monitoring\*\*

- Detailed metrics on enhancement effectiveness

- A/B testing framework for enhancement strategies

- Performance monitoring for processing time

- Quality impact assessment through user feedback

## Implementation Phases

### Phase 1: Core Enhancement Framework (Weeks 1-2)

- Basic template system for philosophical tones

- Simple context integration (user level, preferences)

- Integration with AI Router

- Basic metrics collection

### Phase 2: Advanced Context Integration (Weeks 3-4)

- User history and concept mastery integration

- Question analysis and classification

- Rule-based enhancement engine

- Template versioning system

### Phase 3: Personalization & Learning (Weeks 5-6)

- User-specific tone calibration

- Learning from user feedback

- Contextual continuity across conversations

- Progressive complexity adaptation

### Phase 4: Optimization & Analytics (Weeks 7-8)

- Performance optimization and caching

- Comprehensive analytics dashboard

- A/B testing framework

- Admin tools for template and rule management