

 Status	Not started
--	-------------

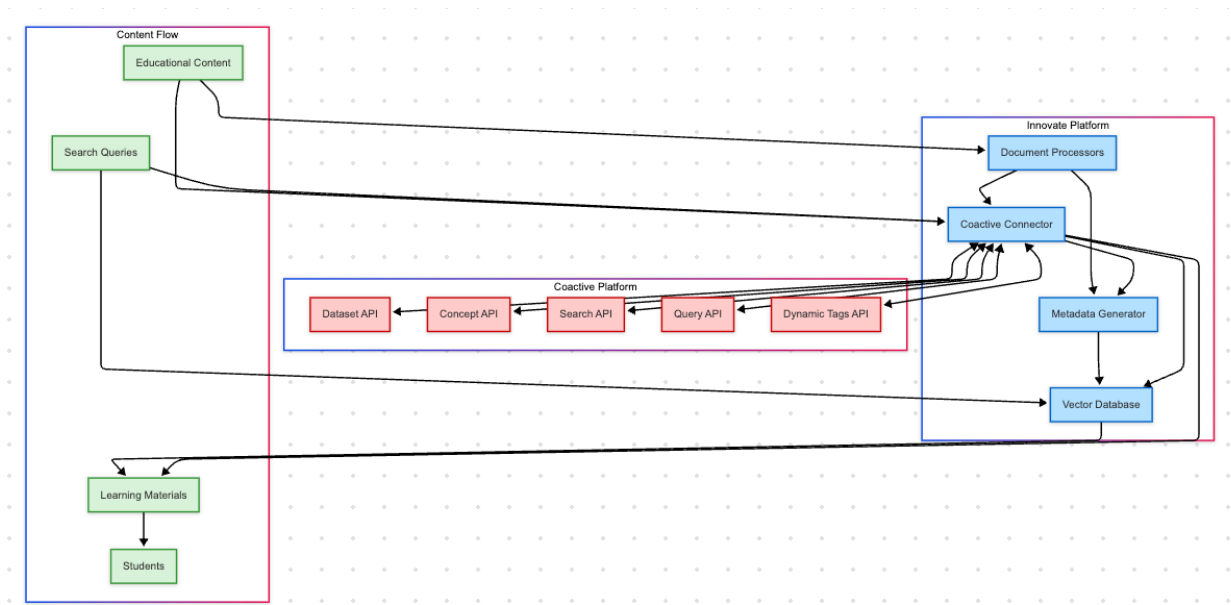
Innovate-Coactive Integration: Technical Specification

Overview

This document outlines the technical integration between Innovate's Unstructured Framework for Educational Content Processing and Coactive's visual understanding API. This partnership will create a comprehensive solution for transforming diverse educational content (text, images, videos) into structured, searchable, and intelligent learning assets.

Architecture

Our integration architecture creates a bidirectional flow between Innovate's text processing capabilities and Coactive's visual understanding:



API Integration Points

1. Authentication & Initialization

```
#integration Notes:
#Innovate's framework will securely store Coactive credentials
#Authentication uses OAuth 2.0 client credentials flow
#Token management handles refresh and expiration automatically
#Connection pooling optimizes API request performance

from unstructured_framework import UnstructuredFramework
from unstructured_framework.connectors import CoactiveConnector

# Initialize our framework with Coactive integration
framework = UnstructuredFramework()
coactive = CoactiveConnector(
    client_id="YOUR_CLIENT_ID",
    client_secret="YOUR_CLIENT_SECRET"
)
framework.register_connector("coactive", coactive)
```

2. Dataset Management

```
#Integration Notes:
#Innovate maps educational courses to Coactive datasets
#Each course creates a corresponding dataset in Coactive
#Metadata synchronization ensures consistency between systems
#Batch processing optimizes large content collections
#Dataset status monitoring ensures processing completion

### What's Happening:
#1) Innovate creates a logical course structure in its system
#2) Coactive connector creates a parallel dataset in Coactive
#3) Visual assets from the course are registered with Coactive
#4) Processing status is monitored and synchronized
#5) Dataset metadata is enriched with educational context
```

```
# Create a dataset for a course
biology_dataset = framework.create_dataset(
    name="Biology_101",
    description="Complete materials for Biology 101 course",
    connector="coactive"
)

# Add visual assets to the dataset
framework.add_assets(
    dataset_id=biology_dataset.dataset_id,
    assets=[
        {"type": "image", "path": "s3://course-materials/biology"},
        {"type": "video", "path": "s3://course-materials/biology"},
    ],
    connector="coactive"
)
```

3. Unified Content Processing

```
# Integration Notes:
# Parallel processing of textual and visual content
# Innovate handles text extraction and analysis
# Coactive processes images and videos
# Results are merged into unified content representations
# Cross-referencing creates connections between text and visual
# Content chunking strategies are coordinated between systems

# What's Happening:
#1) Documents are processed through Innovate's text pipeline
#2) Images and videos are sent to Coactive for processing
#3) Key concepts extracted from text guide visual concept creation
#3) Visual content is analyzed and tagged by Coactive
#4) Metadata from both systems is merged and enhanced
#5) Cross-references are created between related content
#6) Unified content representation is stored in vector database
```

#7) Process mixed educational content

```

result = await framework.process_content(
    content={
        "document": "cell_biology_lecture.pdf",
        "images": ["cell_diagram.jpg", "microscope_image.jpg"],
        "video": "cell_division.mp4"
    },
    options={
        "extract_key_concepts": True,
        "create_visual_concepts": True,
        "generate_dynamic_tags": True,
        "chunking_strategy": "semantic",
        "cross_reference": True
    }
)

```

4. Educational Concept Creation

4. Educational Concept Creation

Integration Notes:

Concepts bridge between educational terminology and visual representation

Innovate extracts key concepts from curriculum and learning objectives

Coactive creates visual concept models from examples

Bidirectional enrichment improves both text and visual understanding

Concept relationships form knowledge graph structure

What's Happening:

#1) Key educational concepts are identified from curriculum

#2) Visual examples of each concept are selected

#3) Coactive creates visual concept models from examples

#4) Models are trained with positive and negative examples

#5) Concept relationships are established

#6) New content is automatically classified against concepts

Concept understanding improves with user feedback

```
# Create educational concepts from course materials
cell_membrane_concept = await framework.create_concept(
    dataset_id=biology_dataset.dataset_id,
    name="CellMembrane",
    description="The cell membrane is a biological membrane that",
    examples=[
        {"asset_id": "membrane_diagram.jpg", "label": 1},
        {"asset_id": "cell_overview.jpg", "label": 1},
        {"asset_id": "nucleus_closeup.jpg", "label": 0} # Negative
    ],
    connector="coactive"
)

# Find content that matches this concept
matches = await framework.get_concept_matches(
    concept_id=cell_membrane_concept.concept_id,
    threshold=0.7
)
```

5. Dynamic Educational Tagging

```
# 5. Dynamic Educational Tagging
# Integration Notes:
# Tags provide flexible, evolving categorization of content
# Innovate defines tag categories based on curriculum structure
# Coactive implements visual recognition of tagged concepts
# Tag prompts bridge between educational language and visual feedback
# Tag hierarchies reflect educational taxonomy

# What's Happening:
# 1. Educational taxonomies define tag categories
# 2. Natural language prompts describe visual characteristics
# 3. Coactive creates dynamic tag models from prompts
# 4. Content is automatically tagged across both systems
```

```

# 5. Tag effectiveness is measured and refined
# 6. Top content for each tag is identified for learning examples
# Create educational tag categories
biology_category = await framework.create_tag_category(
    name="BiologyConcepts",
    description="Key concepts in biology curriculum",
    connector="coactive"
)

# Create dynamic tags for educational concepts
mitochondria_tag = await framework.create_dynamic_tag(
    category_id=biology_category.category_id,
    name="Mitochondria",
    prompts=["powerhouse of the cell", "cellular respiration", "energy production"],
    connector="coactive"
)

# Get top content for educational concepts
top_mitochondria_content = await framework.get_top_content_for_tag(
    tag_id=mitochondria_tag.tag_id,
    limit=10
)

```

6. Unified Educational Search

```

# 6. Unified Educational Search
# Integration Notes:
# Search queries are processed by both Innovate and Coactive systems
# Results are merged with intelligent ranking algorithms
# Filters apply across both textual and visual content
# Search context includes educational relevance factors
# Results include content from multiple modalities

# What's Happening:
# 1. Natural language query is processed

```

```

# 2. Query is sent to both Innovate's vector database and Coactive
# 3. Text-based results are retrieved from Innovate
# 4. Visual-based results are retrieved from Coactive
# 5. Results are merged and ranked by relevance
# 6. Educational context influences result ranking
# 7. Unified results present diverse content types

# Search across all educational content types
search_results = framework.search(
    query="How do mitochondria produce ATP?",
    filters={
        "content_type": ["document", "image", "video"],
        "course": "Biology 101",
        "difficulty": "intermediate"
    },
    limit=10
)

```

Technical Implementation Details

CoactiveConnector Class

```

class CoactiveConnector:
    """Connector for Coactive API integration"""

    def __init__(self, client_id, client_secret):
        self.client = Coactive(
            client_id=client_id,
            client_secret=client_secret
        )
        self.datasets = {} # Cache for dataset information

    async def create_dataset(self, name, description):
        """Create a dataset in Coactive"""
        dataset = self.client.dataset.create(

```

```
        name=name,
        description=description
    )
    self.datasets[dataset.dataset_id] = dataset
    return dataset

async def add_assets(self, dataset_id, assets):
    """Add assets to a Coactive dataset"""
    for asset in assets:
        if asset["type"] == "image":
            self.client.dataset.add_assets(
                dataset_id=dataset_id,
                data_path=asset["path"]
            )
        elif asset["type"] == "video":
            self.client.dataset.add_assets(
                dataset_id=dataset_id,
                data_path=asset["path"]
            )

async def create_concept(self, dataset_id, name, description,
    """Create an educational concept in Coactive"""
    concept = self.client.concept.create(
        dataset_id=dataset_id,
        name=name,
        description=description,
        threshold=threshold
    )

    # Add examples if provided
    if examples:
        labels = [{"asset_id": ex["asset_id"], "label": ex["label"]}
        self.client.concept.update_labels(
            concept_id=concept.concept_id,
            labels=labels
        )
```



```
        return concept

    async def search(self, dataset_id, query, limit=10, offset=0):
        """Search visual content in Coactive"""
        return self.client.search.search_dataset(
            dataset_id=dataset_id,
            query=query,
            limit=limit,
            offset=offset
        )
```

Educational Use Cases

1. Enhanced Learning Materials

The integration enables automatic enhancement of educational content by:

Extracting key concepts from textbooks and lecture notes

- Extracting key concepts from textbooks and lecture notes

Finding relevant visual examples for each concept in Coactive

- Finding relevant visual examples for each concept in Coactive

Creating interactive learning modules with text and visual content

- Creating interactive learning modules with text and visual content

Generating quizzes based on both textual and visual understanding

- Generating quizzes based on both textual and visual understanding

2. Multimodal Search for Learning

Students can search across all educational materials with natural language:

"Show me diagrams of cell division phases"

- "Show me diagrams of cell division phases"

"Find videos explaining photosynthesis"

- "Find videos explaining photosynthesis"

"What are the key differences between mitosis and meiosis?"

- "What are the key differences between mitosis and meiosis?"

The system will return relevant content from both text and visual sources.

3. Visual Concept Mapping

The integration creates visual concept maps for educational topics:

Automatically identify visual representations of key concepts

- Automatically identify visual representations of key concepts

Create relationships between related concepts

- Create relationships between related concepts

Generate visual learning paths through complex topics

- Generate visual learning paths through complex topics

Provide alternative visual explanations for difficult concepts

- Provide alternative visual explanations for difficult concepts