Learner Attribute & Strategy Framework

This document outlines the framework for capturing learner attributes and using them to dynamically generate course content. The system is designed to be direct, efficient, and scalable.

1. The "How": Implementation Plan

This section details the technical approach for capturing, storing, and using the attributes.

a. Initial Diagnostic Assessment

Before the course begins, we perform a two-part assessment to create the baseline `Learner Strategy Profile`.

1. Self-Report Survey (Capturing Preferences):

- **What it is:** A set of static, template-based questions defined by the platform.
- **Purpose:** To capture the learner's goals and content style preferences.
- **Example Questions:**
    - "What is your primary goal for this course? (Mastery / Practical Skills / Overview)"
    - "How do you prefer to learn? (Simple & Clear / Balanced / Academic & Detailed)"

2. Prerequisite Assessment (Assessing Knowledge):*

- **What it is:** A short, dynamically generated quiz.
- **How it's created:** The instructor defines prerequisite topics (e.g., "Data Structures") as metadata for the course. The **Quiz Generation Agent** is then tasked to create a few fundamental questions on these topics.
- **Purpose:** Not to gatekeep, but to identify specific concepts where the learner might need help. If a learner struggles, the system will offer a **quick refresher** on that topic before starting the relevant module.

b. Ongoing Data Capture Methods

1. Explicit Feedback (Post-Module Pop-up):

- After each module, a simple "Did you understand?" prompt is shown. If "No," follow-up questions diagnose the specific issue (e.g., "Need more examples?").

2. Implicit Feedback (Analysis of User Actions):

- **Quiz Results:** A learner's performance on end-of-lesson and end-of-module quizzes provides a direct measure of their conceptual understanding and is a primary driver for adapting the next module.
- ~~**Chat Log Analysis:** The system analyzes questions asked to the SME agent to detect confusion or curiosity.~~

c. Data Storage: The Learner Strategy Profile

All attributes are stored in a standardized JSON object. This object is the dynamic configuration file for each learner.

Example `Learner Strategy Profile`:

```
{
  "learner_id": "user-12345",
  "course_id": "cs-101-os",
  "strategy_parameters": {
    "goal": "mastery",
    "skill_level": "beginner",
    "content_style": {
      "language_simplicity": 0.8,
      "example_density": "high"
    }
  },
  "knowledge_gaps": ["data_structures_recursion"],
  "feedback_log": [
    {
      "module_id": "mod-01-processes",
      "quiz_score": 0.65,
      "understood": false,
      "request": ["more_examples"]
    }
  ]
}
```

Database: This profile is stored in a **NoSQL Document Database** (e.g., Google Firestore, MongoDB) for schema flexibility.

d. System Flow: Agent Communication

1. **Request:** A user begins a new module.
2. **Fetch & Update:** The Orchestrator agent fetches the user's `Learner Strategy Profile`. It reviews recent quiz scores and feedback from the `feedback_log` to update the `strategy_parameters` for the upcoming module.
3. **Delegate:** The Orchestrator sends the updated `strategy_parameters` object in an API call to the **Content Generation Head**.

4. **Generate:** The agent uses the received strategy to generate tailored content. For example, a low quiz score on a topic will trigger the next module to include a brief review and more foundational examples of that topic.

INSTRUCTOR WORKFLOWS

1. SME Registration
   Instructors can register their own SME by choosing LangGraph components from a list.
   (For MVP we will do this for ElasticSearch + VLLMOpenAI only)
   For example,
   a. Chat Head -> Show a list of supported LangChain LLM Providers (Chat models - Docs by LangChain) and allow Instructors to provide URL to their hosted instance of LLM, the prompt to be used, etc.
   b. Knowledge Base -> Show a list of support LangChain Retrievers (Retrievers - Docs by LangChain) and allow instructors to fill the necessary details of their hosted instance, auth, etc

2. Course Creation
   Instructor can create courses by performing the following steps:
   a. Choose an SME of their choice
   b. Upload the content (PDFs, DOCX, PPTX, images, etc.)
   c. Optionally set any learning objectives
   d. **Async Processing**: LangGraph orchestrates the following automated workflow:
      LangGraph Nodes:
      i. `convert_to_markdown`: Uses MarkItDown to convert all uploads to structured Markdown, preserving document hierarchy (headings, sections, tables, code blocks, images)
      ii. `parse_document_structure`: Extracts heading hierarchy (H1-H6), breadcrumb trails, and section relationships from Markdown
      iii. Hierarchical Chunking and create ElasticSearch store
      iv. Use MarkDown Hierarchy to identify Lessons and Modules and Knowledge Components and Learning Objectives within the Modules
   e. **Review & Publish**

LEARNER WORKFLOW

1. Module Start
    - Display an Initial Form to learner.
    - Capture initial attributes/preferences (e.g., wants *brief overview* vs *detailed explanation*, prior knowledge, learning goals).
    - Store these attributes in learner's profile.
2. Learning Module
    - Deliver content (adapted to current attributes).
    - Example: If learner chose *brief overview*, module stays concise; if learner later shifts to *detailed*, next modules adapt accordingly.
3. Feedback Form (after each module)
    - Collect learner's feedback:
        - Was content too easy/too hard?
        - Preferred more detail or less detail?
    - Update learner profile attributes dynamically.
    - Adjust future module delivery (e.g., switch from overview → thorough).
4. Module Quiz
    - Conduct quiz for that module.
    - Store learner's quiz score topic-wise.
    - Mark weak vs strong areas.

5. Next Module
    - Adapt content delivery according to:
        - Updated learner attributes (from feedback).
        - Past quiz performance.
6. Final Quiz (End of Course)
    - Aggregate topic scores.
    - Give more weightage to weak subjects identified earlier.
    - Ensure learner gets a balanced assessment, but focused on their weaker areas.

Flow Diagram

[Start Module]

↓

[Initial Form → Capture learner attributes]

↓

[Module Content Delivery (based on attributes)]

↓

[Feedback Form → Update attributes]

↓

[Module Quiz → Store topic scores]

↓

[Next Module → Adapted based on feedback]

↓

...

↓

[Final Quiz → Weighted on weak topics]

↓

[Course End]