Model and Software Documentation

Model Documentation Agent: Detailed Project Documentation

1. Introduction

This document describes the design and functionality of a conversational AI agent built with LangChain and Claude, tailored for automated documentation of risk-related models within Bank of Montreal's Enterprise Risk framework. The agent ingests a multi-file codebase for a model or application along with a user-provided documentation template (in structured JSON), and produces comprehensive, metadata-rich documentation drafts, outlines, and improvement insights.

2. Objectives

1. **Automated Outline Generation:** Create draft outlines aligned to each section of the user's documentation template.
2. **Summaries Engine:** Summarize individual scripts, then generate hierarchical summaries (summaries of summaries) mapping relationships across files.
3. **Section Mapping & Indexing:** Map codebase summaries to documentation sections to build a metadata-rich index for targeted retrieval.
4. **Draft Documentation:** For each template section, retrieve relevant summaries and code snippets to compose documentation content (overview, analysis, recommendations).
5. **Insight & Improvement Suggestions:** Analyze documentation and code to suggest future enhancements, refactoring opportunities, and best practices.
6. **Multi-Codebase Flexibility:** Enable seamless support for diverse codebase types—risk models, TMAs, BMAs, and full applications—by defining dedicated guidelines, outline structures, and prompt templates for each type.

7. **Conversational Fluidity:** Maintain a self-aware, professional yet approachable assistant persona that can dynamically adjust prompts, expose its templates and methods, and handle user-driven prompt modifications.

---

3. Architecture Overview Architecture Overview

3.1 Inputs

- **Codebase Files:** A set of scripts (e.g., Python files) comprising risk models (VAR, PFE, linear regression, time-series analysis), BMA scripts (small process automations like ME reporting), and full applications.
- **SaaS Tool Definitions:** Configuration or API specification files for TMA (Transaction Monitoring Application) SaaS tools such as CCR monitoring platforms.
- **Documentation Template:** A JSON structure defining sections (e.g., Introduction, Methodology, Usage, Limitations, Recommendations, References) and tailored templates per codebase type.

3.2 Core Components Core Components

1. **Data Loader:** Reads the codebase into memory, splitting by file and optionally by logical chunks (classes, functions).
2. **Summaries Engine:** Uses templated prompts to generate:
   - **File Summaries:** Key purpose, logic, data flows per script.
   - **Chunk Summaries:** Smaller units (e.g., functions, classes) when needed.
   - **Hierarchical Summaries:** Aggregated summaries linking files, highlighting dependencies and thematic groups.
3. **Index Builder:** Creates a metadata-rich index mapping summaries and code references to template sections.
4. **Prompt Templates:** Predefined and dynamically modifiable prompts for:
   - Outline drafting per section.
   - Section-level documentation generation.

    - Holistic review and refinement prompts.

5. **Tools (LangChain Wrappers):** Function and batch tools for:
    - Summarization
    - Index querying and retrieval
    - Documentation composition per section
    - Methodology explanation and prompt introspection
6. **Agent Executor:** A LangChain agent configured with Claude, capable of:
    - Parsing user intent (e.g., "Generate Methodology section draft")
    - Selecting appropriate tools
    - Managing dynamic prompt updates
    - Enforcing limits (e.g., number of files processed in one call)

4. Summaries Engine & Indexing

1. **File-Level Summaries:** For each script, the agent runs an LLMChain with a "Summarize this file" prompt, capturing purpose, inputs/outputs, key algorithms.
2. **Hierarchical Summaries:** Group related file summaries (e.g., data loaders vs. model training scripts) into higher-level summaries.
3. **Section Mapping:** Using the documentation template’s section titles and descriptions, the agent assigns each summary to relevant sections in the index (e.g., Methodology, Implementation Details).
4. **Metadata:** Each index entry includes file name, summary text, relevant code snippets, and assigned template section(s).

5. Documentation Generation Workflow

1. **Outline Drafting:** Based on the template structure, generate a nested outline listing all sections and subsections.
2. **Section Content Assembly:** For each section:

    - Query the index for summaries and code snippets tagged to that section.
    - Use a prompt template to draft prose combining summaries, code context, and user guidelines.
3. **Holistic Review:** After all sections are drafted, run a review chain to ensure coherence, consistency in terminology, and completeness.
4. **Improvement Suggestions:** Generate a final section with actionable insights for code enhancements, documentation gaps, and model optimization tips.

6. Agent Persona & Interaction Flow

- **Persona:** A knowledgeable, risk-model documentation specialist acting as a work-buddy.
- **Self-Awareness:** Can report its available tools, loaded template, summary index, and prompt structures.
- **Dynamic Prompting:** Users can inspect or override prompts (e.g., "Show me your outline prompt" or "Modify Methodology prompt to emphasize data validation").
- **Error Handling:** Clarifies when inputs exceed configured limits or if template sections are missing.

7. Future Enhancements

- **Persistent Documentation Repositories:** Version-control generated docs in a database or Git.
- **Semantic Code Search:** Integrate vector embeddings for semantic snippet retrieval.
- **User Feedback Loop:** Allow users to rate section drafts and refine prompts based on feedback.
- **Multi-Format Output:** Support export to Markdown, HTML, PDF, or Confluence pages.
- **Template Library:** Maintain a library of industry-standard documentation templates for various model types.

This comprehensive blueprint outlines the components, workflows, and user experience considerations needed to build an effective **Model Documentation Agent** for enterprise risk models.