Technical Proposal for 3-Part Design Document Generation Project

Contents

TECHNICAL SCOPE OF WORK

Proof of Concept -1: **Knowledge Graph Generation by Digitization of Static Process Flow Diagrams**

Estimated Duration 2.3 Months

TECHNICALS

INITIAL REQUIREMENTS

- Collaborate with SMEs to define the requirements.
- Select two specific PFDs for the POC.
- Gather sample questions from SMEs to ensure the knowledge graph answers relevant and simple queries effectively.

DEEP LEARNING MODEL DEVELOPMENT

- Develop models to extract components from PFDs, including symbols, connections, and textual annotations.
- Implement advanced CNN Models to recognize and digitize PFD elements accurately. (Refer to Fig 1 for types of entities to be extracted.)

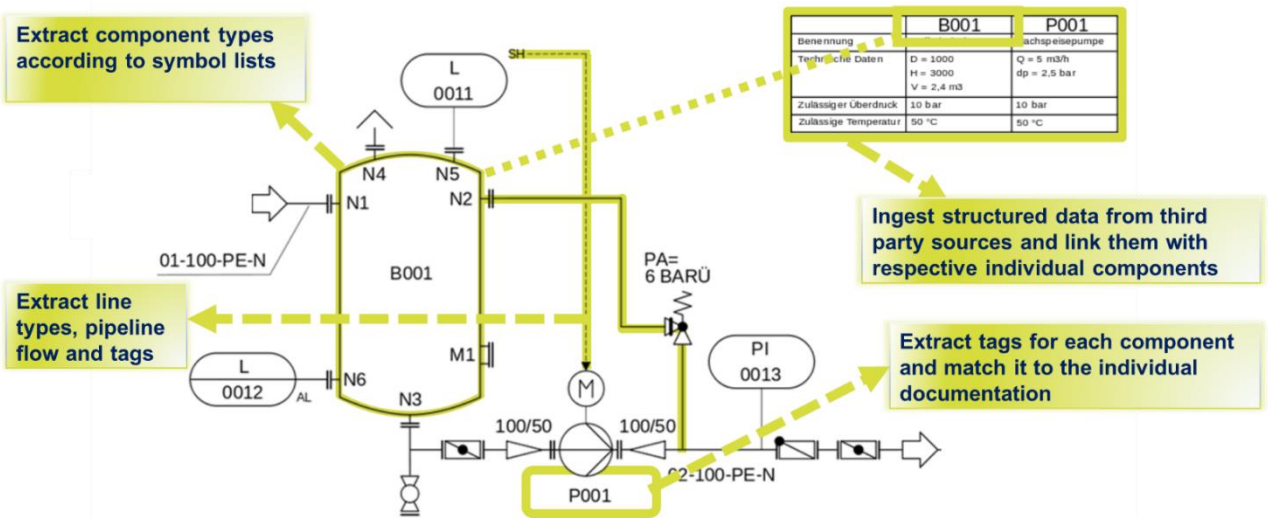

Figure 1: Extract Entities using Deep Learning

KNOWLEDGE GRAPH CONSTRUCTION

- Design and construct the knowledge graph, categorizing and interlinking information appropriately.
- Populate the knowledge graph with extracted and integrated data.

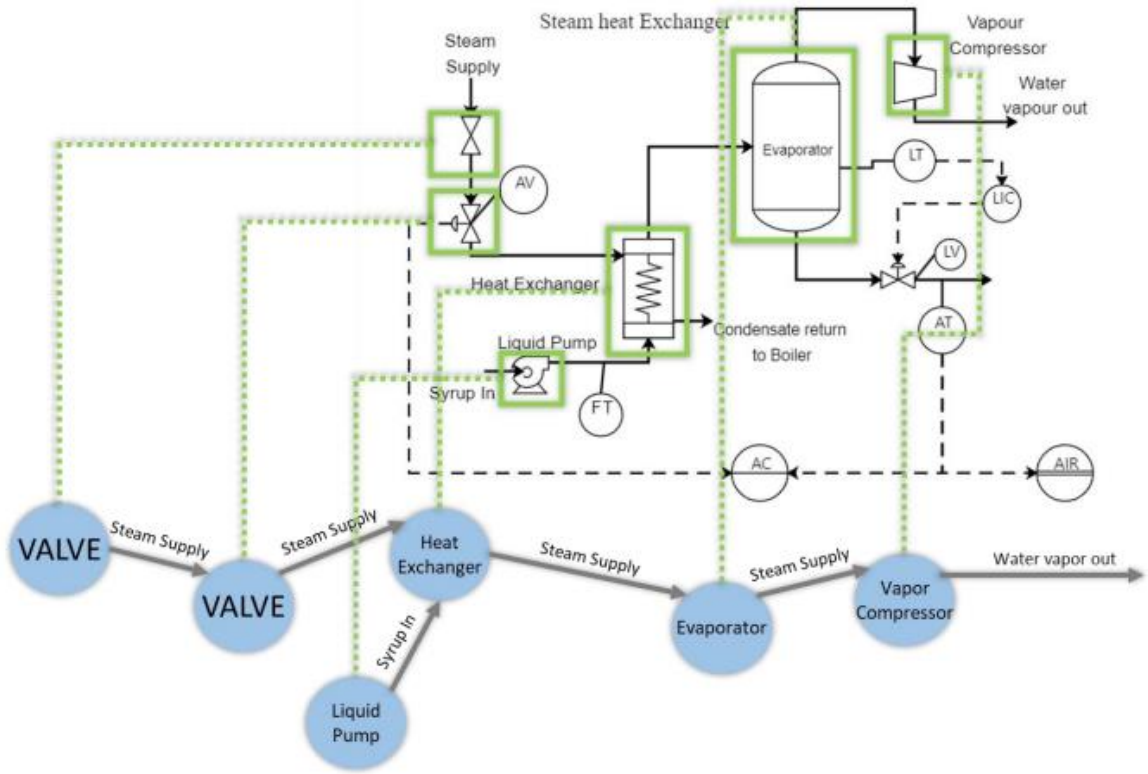

Figure 2: A P&ID diagram digitized and componentized into a Knowledge Graph

CONNECT WITH OPEN SOURCE LLMS

- Establish API connections with open source LLMs that are locally installed.
- Compare results with commercially available LLMs like GPT to verify accuracy.

RETRIEVAL AUGMENTED GENERATION (RAG)

- Develop RAG algorithms enabling users to query the knowledge graph.
- Ensure responses are contextually accurate, minimizing hallucinations.

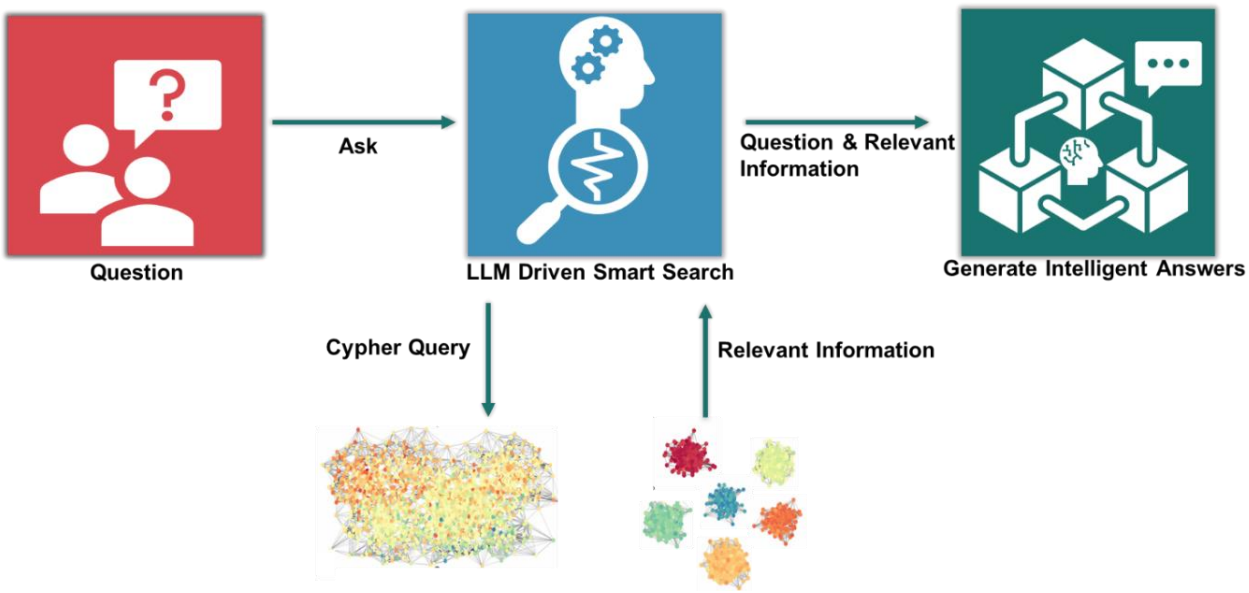

Figure 3: Use LLMs to talk to your data with minimal hallucinations using RAG

DELIVERABLES

- Detailed requirement analysis document.
- Deep learning models for PFD component extraction.
- Fully functional and populated knowledge graph.
- Knowledge graph visualization
- RAG activated LLM Integration with knowledge graph for analytics queries.
- Demonstration of RAG capabilities with sample queries.

SUCCESS CRITERIA

- Accurate digitization of PFD elements with minimal errors.
- Fully functional and populated knowledge graph.
- Open Source LLM integration and validation of RAG.
- Reliable and contextually accurate query responses using RAG.

Proof of Concept – 2: **Unstructured data digitization and Gen-AI Powered Process Template Generation**

Estimated duration 2.6 months

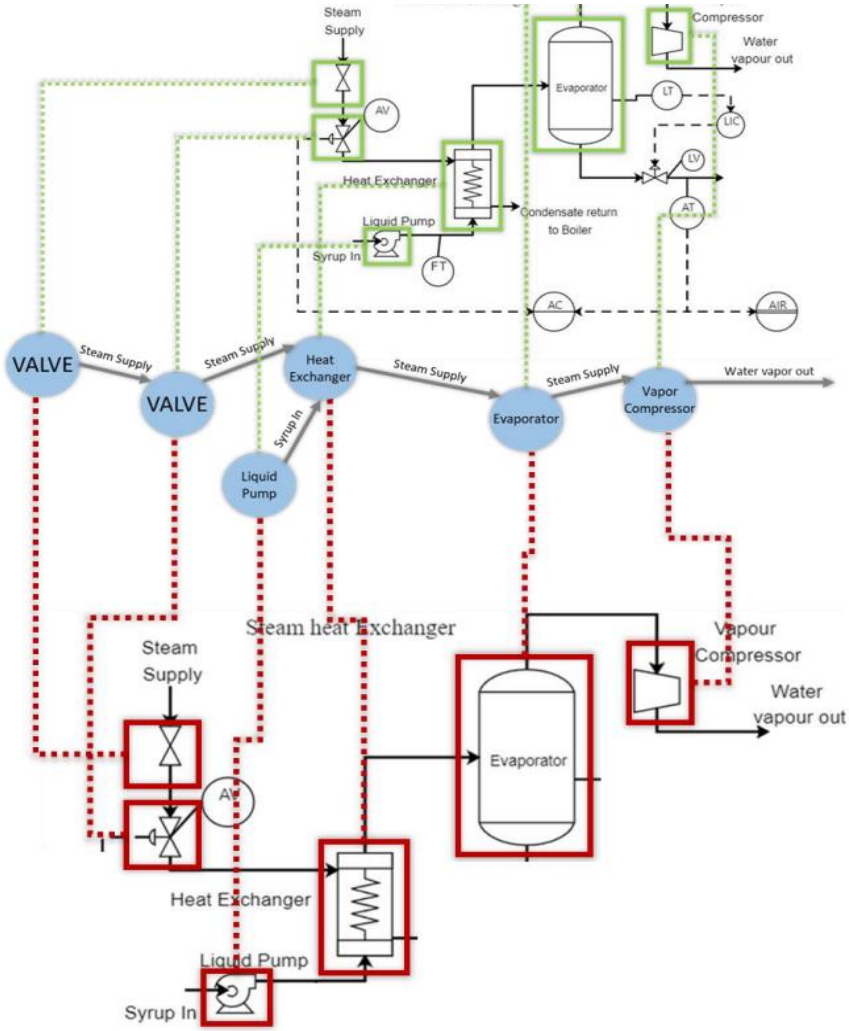

Figure 4: Creation of new P&ID/PFD diagrams using templatized components present in the knowledge graph (In RED).

TECHNICALS

RAG POWERED TEMPLATE CREATION FOR PFD DOCUMENTS

- Analyse existing knowledge graph to identify common elements of PFDs and structural patterns.
- Cluster similar common components/elements/sub processes using graph data science
- Develop templates encapsulating these commonalities.

Template Analysis

- Perform a detailed analysis of PFDs to ensure comprehensive template coverage.
- Incorporate feedback from SMEs to refine template accuracy and relevance.

Further Knowledge Graph Enhancement with Plot Plan Diagrams and Unstructured Data Ingestion

- Enhance the knowledge graph with digitization of unstructured data like technical field reports/reviews and Plot Plan Diagrams to encapsulate SME knowledge.
- Use the enhanced knowledge graph to provide technical recommendations on process templates.

Abstract/Summary Generation through RAG

- Generate Abstracts/Summary Templates using RAG to minimise hallucinations.
- Take in precise inputs for the knowledge graph to ensure technical quality
- Generate summary report for SME quality control

Deliverables

- Digitization and knowledge graph integration of Plot Plan Diagrams and unstructured data like field reports/technical reviews.
- Comprehensive library of PFD templates.
- Detailed analysis report of common PFD elements and structures.
- Generated PFD query templates whose outputs match the PFDs used to create the knowledge graph.
- Generated Abstract/Summary report that passes SME quality checks.

Success Criteria

- Enhanced knowledge graph with unstructured field reports and Plot Plan Diagrams data
- Highly accurate RAG driven PFD query templates, ensuring that they match the PFDs used to create the knowledge graph.
- Effective utilization of templates and knowledge graph data.
- Minimal occurrence of hallucinations abstract and summary.
- Successful validation of Abstract/Summary report.

Proof of Concept – 3: **Gen AI-Driven Process Diagram Generation, Validation and integration with existing environment**

Estimated duration 2.3 Months

TECHNICALS

GENERATION OF PROCESS FLOW DIAGRAMS(PFDS) AND PLOT PLAN DIAGRAMS USING GEN-AI

- Implement RAG and generative AI algorithms that utilize the templates created in the previous POC from knowledge graph data to generate actual Process Flow Diagrams.
- Develop methods to ensure AI-generated PFDs are accurate and meet required specifications.

IMPLEMENT FLEXIBILITY OF COMPONENTS USING GENAI FOR DIAGRAM GENERATION

- Create a library of PFD templates for use in AI generation processes.
- Utilize similarity analysis of previous POC to generate recommendation templates for similar components for real world situations
- Ensure flexibility in generated diagrams so that they can be modified according to various process scenarios.

INTEGRATION WITH ASPEN SOFTWARE FOR VALIDATION

- Use ASPEN systems for validation of generated diagrams.
- Generated diagrams to be ingested by aspen systems to run simulation testing.
- Ensures accuracy of generated diagrams from SMEs
- Knowledge transfer and training to designated engineers to help operate and test the system.
- Help the team create the development roadmap so as to implement it for production scale.

DELIVERABLES

- Creating Process Flow Diagrams from templatized/RAG based queries.
- Implementing recommendation templates for real world situations – from the data provided to the system.
- Scenario based diagrams output contextualizing the process scenario presented.

- Enabling seamless connect with ASPEN system for simulation testing.
- Conduct training and Knowledge transfer sessions for the team
- Build the solutions development and scaling roadmap to implement the solution for production.
- Documenting the steps and methods for future use.

Success Criteria

- Optimizing the process flow diagrams created for high level of accuracy.
- Ensure Recommendation templates deliver output as per expectations set.
- Enabling Operationalizing the connect with ASPEN simulation testing product.
- Ensuring that the solution behaviour is consistent with inputs and contexts provided and is validated by SMEs as expected behaviour from an automation assistant.

IMPORTANT NOTES

1. While this technical proposal was made, the exact quantum of inputs and expected outputs are still being decided.
2. Once the exact data for inputs and expected outputs are well defined by, we shall prepare a set of measurable metrics to assess the success criteria of each POC project more objectively for each of the POC's stated above.
3. The measurable metrics so prepared at the beginning of each POC project will be in consultation with SME's, participating technology engineers and the Softlink Data Scientists, Graph database specialists and data engineering specialists.

ANNEXURE – I

SOFTWARE REQUIREMENTS

This section of the document meticulously details the software requirements crucial for the generation of design documents. It begins by specifying the Python environment, pinpointing Python 3.8 as the foundational version, and enumerates a comprehensive list of required libraries. These libraries, including Torch for PID symbol detection, opencv-python for image preprocessing, and others like NLTK and TensorFlow, are essential for various functionalities ranging from optical character recognition to deep learning and text processing.

Additionally, this section outlines the database requirements, emphasizing the use of Neo4j Enterprise and Neo4j GDS throughout the duration of the project as it will form the single source of truth for the entire knowledge graph. The graph database along with Ontotext (to manage different ontology maps created during different stages of the project) will form the base for all genAI systems to generate content without hallucination and mistakes. Graph visualization needs are addressed with the inclusion of Linkurious Enterprise. Furthermore, the document lists additional software necessities such as GPUs, preferably NVIDIA, and various development tools like Microsoft Visual Studio Code and Notepad++.

Operating systems form an integral part of the software requirements, with Linux specified for development and server infrastructure, and Windows VM recommended for remote access to these servers. This comprehensive outline of software requirements ensures that all technical aspects are thoroughly covered, providing a robust framework for the project's successful execution.

PYTHON ENVIRONMENT

Version: 3.8

- Libraries:
    - Torch (v1.12): for PID symbol detection
    - opencv-python (v4.9): for image preprocessing
    - scikit-image (0.22): for post-processing
    - pytesseract (0.3): for optical character recognition
    - NLTK, spaCy, or Sentence Transformers for text processing
    - TensorFlow, PyTorch for deep learning frameworks
    - Pinecone, Vectr for vector-based retrieval

- PyPDF2, pdfquery, Camelot for PDF parsing
- Flask or Django for web interface (optional)
- RESTful API library like FastAPI (optional)
- matplotlib
- numpy
- pandas
- Pillow (v10.0)
- pypdf
- langchain

(More libraries maybe required during development according to need of the project)

DATABASE:

Neo4j - Enterprise-grade GraphDB to store and process the Graph Data and easily integrate with Big-Data cluster.

Ontotext 1 Node 8 Core 256GB RAM: To create and store ontology maps**. (optional)**

GRAPH VISUALIZATION:

- Linkurious Enterprise (FOC Trial license for POC)

ADDITIONAL SOFTWARE:

- GPUs (dependent on workload) (Preferably NVIDIA)
- Local LLM models (eg: MISTRAL 8X7B configured with API keys)
- Windows snipping tool for cropping images.
- Microsoft Visual Studio Code/PyCharm IDE for python code development
- Notepad++ windows

OPERATING SYSTEMS:

- Linux (Red Hat or Debian) for development and server infrastructure
- Windows VM for accessing Linux servers with remote desktop software.

HARDWARE REQUIREMENTS

Server(s) with:

- CPUs: Sufficient cores to handle workload (dependent on specific tasks and data size) min 16.
- RAM: 256 - 512 GB minimum, potentially more depending on data size and processing needs
- GPUs: GPUs with sufficient memory and processing power (dependent on model complexity and workload)
- Storage: Sufficient space for data storage, model training, and generated documents

Windows VM with:

- CPU: Minimum 16 cores
- RAM: Minimum 32 GB
- Remote desktop software

Network Requirements

1. High-speed network connection for communication between development machines, servers, and the Windows VM.

Additional Considerations

- **Security:** Implement appropriate security measures for data access, storage, and communication.
- **Scalability:** Consider the potential for future growth and ensure the infrastructure can be scaled accordingly.
- **Backup and Disaster Recovery:** Implement robust backup and disaster recovery strategies to protect against data loss.
- **Monitoring and Logging:** Implement monitoring and logging systems to track performance and identify potential issues.
- **Documentation:** Thoroughly document the project setup, configuration, and operation procedures.