Leveraging Large Language Models to Automate SOP in Warehouses Managed by Warehouse Management Systems

Gautham Ram Rajendiran

gautham.rajendiran@icloud.com

Abstract
Warehouses that handle inventory management in logistics and supply chain typically involve the use of Standard Operating Procedures (SOPs) in order to deal with exceptional scenarios or to handle a very rare situation. This paper proposes a conceptual framework that leverages capabilities provided by Large Language Models (LLM) in order to build a system that can automate such SOP interpretation and execution. The system is designed adapting to different scenarios that can arise while working on a SOP. This idea sets the foundation for further research into scalable and adaptive SOP automation for warehouse operations.

Keywords: Warehouse Management, Large Language Model, Artificial Intelligence

Introduction
Warehouse management operations that typically span across borders are becoming increasingly complex due to the variable nature of compliance regulations, value added service requirements and shipping conditions [1]. Standard Operating Procedures are used extensively in order to handle the various scenarios that do not fall into the normal flow of things, this makes sure that warehouse operations remain efficient and takes out the delays due to ad-hoc decision making [2]. However, SOP execution largely relies on manual intervention and requires a warehouse associate to follow execution steps that consumes valuable time which could have been utilized elsewhere.
With the rise of artificial intelligence, particularly Large Language Models (LLMs), there is an opportunity to automate these processes. This paper proposes a conceptual framework to utilize LLMs to interpret SOPs and generate execution steps which can be executed by an API call.
Problem Statement
The conceptual framework presented in this paper aims to address the time consuming and complex nature of work faced by warehouse associates while following SOPs by using LLMs to automate the process. By this method it aims to tackle the following situations.

1. **Manual Processes:** Current SOPs are often followed manually by warehouse associates, requiring them to make decisions based on the procedures outlined. This manual intervention is time-consuming and increases the likelihood of mistakes, especially when dealing with complex regulatory requirements or high volumes of items.
2. **Operational Inefficiency**: SOPs must be flexible to account for different scenarios such as changes in product type, regulatory requirements or geographic location. However, manually adjusting SOPs

for each specific case can slow down operations and introduce bottlenecks.

3. **Regulatory Compliance**: Regulations regarding the handling of hazardous items vary across different regions and countries [3]. For example, in one country, hazardous items may need to be isolated for inspection, while in another, they can be shipped with a warning label. Warehouse associates must manually adjust their actions based on these varying rules, which can lead to errors or non-compliance.

Literature Review

The prior work on warehouse automation ranges from robotic inventory management solutions [4][5], supply chain optimization driven by predictive analytics, and AI-driven operational planning [4].

The involvement of robotics and automation technologies to enhance warehouse efficiency has taken center stage through automated guided vehicles handling AGV and robotic arms performing basic tasks of inventory management, picking, and packing. The research in this area has concentrated on how improvements in warehouse efficiency can be realized through the automation of mundane and/or repetitive manual tasks. There have been related studies with regards to advantages in working with collaborative robots-or, simply, cobots-which work in unison with human workers in order to obtain an even higher level of productivity. An example is Amazon, which uses Kiva robots to assist in moving inventory throughout their fulfillment centers.

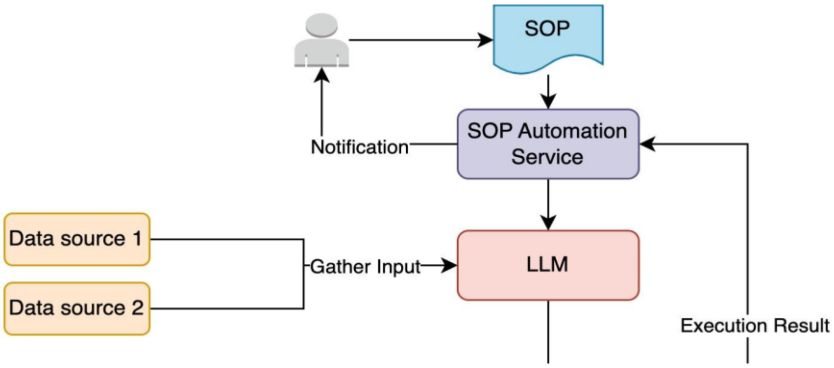
Predictive analytics has now turned into a very critical area of research in optimizing supply chains. Predictive analytics uses machine learning and big data to predict future demands, locate imminent supply chain bottlenecks, and make suggestions toward corrective actions in order not to be delayed. According to research these approaches result in better decision-making and resource allocations that enhance operational efficiency [6].

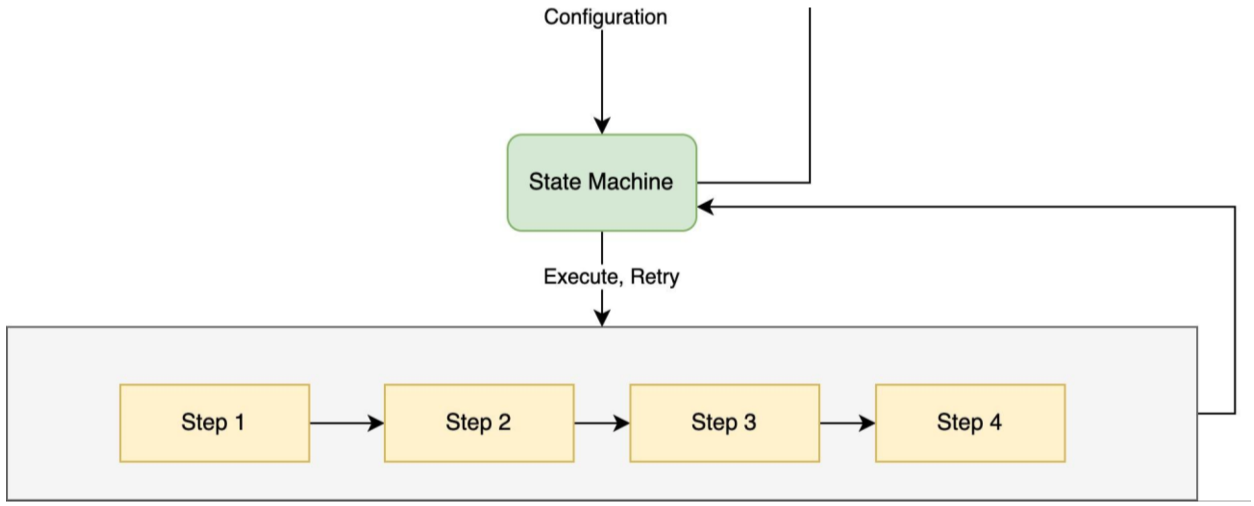
AI-based operational planning is used to increase efficiency of warehouse operations [7] by automating decision-making processes such as route planning for shipments, inventory replenishment and workforce management. AI-driven warehouse management systems (WMS) use data from various sources like customer demand patterns and seasonal variations to optimize workflows.

However, the use of LLMs to interpret SOPs and dynamically adjust the execution steps based on context is a novel concept.

Conceptual Framework

The proposed solution is centered around a LLM capable of dynamically interpreting SOPs and generating executable configurations. These configurations are sent to a state machine system [8] that orchestrates the execution of tasks through the use of dedicated services. The LLM is pre-trained with the information on identifying the correct service to use for a particular use case.





Pre-Training

The model is pre-trained on vast amounts of text data from diverse domains. In this process, the model gets exposed to many use cases in an operational environment including warehouse processes, workflows, SOPs, and associated services such as inventory management, order fulfillment, labeling and shipping. In this manner, the Large Language Model develops general understanding and reasoning skills in the domain of warehouse management. Pretraining in such systems does not imply explicit training for every possible interaction with any service but rather an extensive familiarization within the model of task structure, pattern, and language that define operations in a warehouse that can enable decision making in a warehouse given a SOP.

Once the general-purpose LLM is pre-trained, it can be fine-tuned for more specific mappings into datasets of common SOP tasks and the corresponding services that are used in executing the tasks. For example, taking each of these use cases around warehouses, the LLM may be fine-tuned on data that correlates natural language descriptions of tasks to the appropriate services needed to complete the tasks. Associations like the following are learned from a diverse set of examples of warehouse tasks associated with different services.

LLM as a Decision-Making Engine

LLMs have demonstrated strong capabilities in understanding natural language, making them suitable for interpreting SOPs. The LLM acts as a decision-making engine that can:

1. **Understand the SOP**: The LLM reads the SOPs, which are written in natural language, and identifies the steps that need to be performed. For instance, when handling hazardous materials, the LLM recognizes actions such as retrieving item information, moving the item to a specific location, or labeling the item.
2. **Adapt to Context**: The LLM is provided with contextual information, such as the country where the task is being performed or specific product details. This allows the LLM to adapt its output to fit the requirements of the situation. For example, in one country, the LLM may generate a plan to isolate the hazardous item for inspection, while in another, it may generate a plan to ship the item with a warning label.
3. **Generate Execution Steps**: Once the LLM has interpreted the SOP and contextual information, it generates a dynamic configuration or execution plan. This plan consists of a sequence of actions that

can be executed by a state machine system, with each action mapped to a dedicated service.

Dynamic Configuration Generation

The LLM produces a configuration, which is essentially a set of actions or steps that are automatically tailored to the specific situation

1. **Action Steps**: A list of actions that need to be performed in order to complete the task. Each action is associated with a dedicated service that will carry out the task.
2. **Parameters**: The necessary data inputs for each action. For instance, when retrieving item information, the configuration might include the item ID and the order number.
3. **Conditional Logic**: The configuration can include decision points where the next action depends on the outcome of a previous action. For example, if an item is determined to be hazardous, the next step might be to isolate it or apply a warning label, depending on the country's regulations.

Orchestration

Once the LLM generates the configuration, a state machine system is responsible for orchestrating the execution. The LLM has knowledge about the domain specific language that is used to instruct the state machine. The state machine reads the configuration and does the following:

1. **Executes Steps**: Invokes the necessary services, such as retrieving item details, modifying orders, or applying warning labels.
2. **Monitors Execution**: Tracks the progress of each step, ensuring that tasks are performed in sequence and handled correctly.
3. **Handles Errors**: If an error occurs, such as a failure to retrieve item details, the state machine triggers a retry mechanism or escalates the issue to a human operator.

Many contemporary cloud native infrastructure as a service provider provides robust general-purpose state machine implementations that can be put to use. For example Amazon Web Services provides Step Functions [9], Google Cloud provides Google Cloud Composer [10] and Microsoft's Azure provides Azure Logic Apps [11]. For use-cases that need more customization, implementations can refer to the vast body of knowledge around building robust state machine implementations to cater the implementation to its own unique scenario.

State machines use a domain specific language (DSL) [12] that can be understood by the consuming state machine. For the purpose of training the LLM to write this DSL, it is better to choose a state machine implementation that has extensive documentation like those provided by the services mentioned above or other open source implementations that have a significant number of contributions and maintainers.

Dedicated Services

Each task in the SOP is performed by a dedicated service and the LLM has knowledge of which service or set of services to call in order to complete a single step in the SOP. Examples of such services are:

1. **Item Information Service**: Retrieves the details of the item.
2. **Order Modification Service**: Removes or modifies items in the order.
3. **Labeling Service**: Applies warning labels to hazardous materials.
4. **Notification Service**: Sends alerts to stakeholders, such as regulatory bodies or customers.

Example Use Case: Hazardous Item Management

The entry point into the system is the SOP Automation Service which manages interaction between the user and the model. The inputs to this service can be a SOP or a simple prompt from the user, following which the service will invoke the LLM client for prompt interpretation and configuration generation.

In order to demonstrate the practical implementation of this LLM based automation solution, let us consider the scenario where the warehouse operator needs to deal with a hazardous item wherein they usually follow a SOP. And as an example, let us say that the steps to follow would vary based on the item being processed in the same SOP.
Item A: When a hazardous item is found, place it in the holding area for further inspection and ship the rest.
Item B: Ship the hazardous item but add a warning to the shipping label.
The LLM would now generate two kinds of steps for each of the above use cases:
Steps to process an item that needs to be placed in a holding area.
steps:
action: getItemInfo
service: Item Information Service
parameters:
itemId: "H123"
action: removeItemFromOrder
service: Order Modification Service
parameters:
orderId: "O456"
itemId: "H123"
action: moveItemToHolding
service: Inventory Movement Service
parameters:
itemId: "H123"
location: "Inspection Area"
action: notifyRegulatoryBody
service: Notification Service
parameters:
message: "Hazardous item H123 moved to Inspection Area for Country."

Steps to process an item that can be processed with a warning label.
steps:
action: getItemInfo
service: Item Information Service
parameters:
itemId: "H123"
action: applyWarningLabel
service: Labeling Service
parameters:
itemId: "H123"
label: "Hazardous Material"
action: proceedWithShipping
service: Shipping Service
parameters:

orderId: "O456"
action: notifyCustomer
service: Notification Service
parameters:
message: "Order O456 containing hazardous item H123 will be shipped with a warning label for Country B."

The LLM decides the steps depending on the information contained in the SOP that provides details on the ways to handle specific types of hazardous items.

The YAML [13] instruction generated will be passed into a state machine that can consume the instruction and action upon each step.

The state machine implementation contains the logic to consume the instruction generated by the model. It also has a mapping of the service name provided in the configuration to service client details like the client location, methods exposed and parameters. It has robust retry strategies and well-defined error handling mechanisms.

Once the steps defined by the model in the configuration are successfully executed, the state machine provides a response back to the SOP automation service which is then surfaced to the user.

Benefits of the Proposed Approach

Adaptability

The LLM-based framework offers high adaptability due to its capability of creating execution steps based on real-time contextual information like country-specific regulations. This flexibility ensures that warehouses can handle different situations dynamically without manual reprogramming.

Error Reduction

By automating the interpretation and execution of SOPs, the framework significantly reduces the likelihood of human error.

Improved Efficiency

Automating decision-making processes reduces the time needed for manual interventions, increasing overall operational efficiency. Tasks that would typically take hours for a warehouse associate to complete can be done automatically by the system. This provides the associate additional time to process regular orders.

Challenges and Future Considerations

Data Quality

The LLM's effectiveness depends on the accuracy of its input data, including up-to-date SOP information. It will be essential to maintain a real-time link with regulatory databases to ensure compliance.

Model Reliability

While LLMs are powerful, they are not infallible. Implementing validation steps to verify the LLM’s output is crucial to prevent incorrect or non-compliant execution plans from being implemented. Mechanisms for Ground Truth verification and detecting data drifts should be coupled with proper alarms and monitors in order to keep the model relevant and ensure high accuracy.

System Integration

Integrating LLM-generated configurations with existing warehouse management systems (WMS) and ensuring seamless communication between services is a challenge that needs to be addressed for successful deployment.

Conclusion

This paper proposes a conceptual framework for using Large language models to automate SOP execution in warehouses and presents the concept of a functionality of dynamically creating configurations for SOP automation that can cater to any operational context. It also explores the implementation in the context of a real-world use case. Future research will focus on implementations based on this framework, scalability considerations and the reliability of using LLM.

References

1. Faber, Nynke, M. B. M. De Koster, and Ale Smidts. "Organizing warehouse management." International Journal of Operations & Production Management 33.9 (2013): 1230-1256.
2. Khairuddin, Muhamad Munzir, et al. "Regulator perspective on Halal air cargo warehouse compliance." International Journal of Supply Chain Management 7.3 (2018): 202-207.
3. Sun, P. and Gu, L., 2021. Optimization of Cross-Border e-Commerce Logistics Supervision System Based on Internet of Things Technology. Complexity, 2021(1), p.4582838
4. Fatima, Zainab, et al. "Production plant and warehouse automation with IoT and industry 5.0." Applied Sciences 12.4 (2022): 2053.
5. Baker, Peter, and Zaheed Halim. "An exploration of warehouse automation implementations: cost, service and flexibility issues." Supply Chain Management: An International Journal 12.2 (2007): 129-138.
6. Carbonneau, Real, Kevin Laframboise, and Rustam Vahidov. "Application of machine learning techniques for supply chain demand forecasting." European journal of operational research 184.3 (2008): 1140-1154.
7. Hompel, Michael, and Thorsten Schmidt. Warehouse management: automation and organisation of warehouse and order picking systems. Springer Science & Business Media, 2006.
8. Schneider, Fred B. "Implementing fault-tolerant services using the state machine approach: A tutorial." ACM Computing Surveys (CSUR) 22.4 (1990): 299-319.
9. Amazon Web Services, "AWS Step Functions," Amazon Web Services. [Online]. Available: https://aws.amazon.com/step-functions/.
10. Google Cloud, "Cloud Composer," Google Cloud. [Online]. Available: https://cloud.google.com/composer.
11. Microsoft, "Azure Logic Apps documentation," Microsoft Learn. [Online]. Available: https://learn.microsoft.com/en-us/azure/logic-apps.
12. Mernik, Marjan, Jan Heering, and Anthony M. Sloane. "When and how to develop domain-specific languages." ACM computing surveys (CSUR) 37.4 (2005): 316-344.
13. Ben-Kiki, Oren, Clark Evans, and Brian Ingerson. "Yaml ain't markup language (yaml™) version 1.1." Working Draft 2008 5.11 (2009).