Computer Science and Engineering 2025, 15(3): 75-78
DOI: 10.5923/j.computer.20251503.02

Hierarchical AI Agent Framework for SAP Business Process Automation

Ajay Verma

MCA – SAP Technical Architect, Dayton, Ohio, United States of America

Abstract This research paper presents a comprehensive framework for leveraging a hierarchical AI architecture to automate SAP business processes. With the increasing complexity of enterprise resource planning (ERP) systems, particularly SAP, there is a growing demand for intelligent, modular, and collaborative AI solutions. This paper introduces a three-level AI agent hierarchy tailored for SAP environments: Level 1 (Executive Agents), Level 2 (Managerial Agents), and Level 3 (Operational Agents). The study outlines the roles, interactions, and technical implementations of each agent class using state-of-the-art technologies like LangChain, large language models (LLMs), vector databases, and orchestration layers. Furthermore, we discuss practical use cases, benefits, challenges, and prospects of AI-driven SAP automation.

Keywords Hierarchical AI agents, SAP automation, LangChain, Business process orchestration, ERP systems, AI-driven workflow

1. Introduction

1.1. The Role of SAP in Enterprise Environments

SAP (Systems, Applications, and Products in Data Processing) is a leading ERP system that integrates core business functions including finance, supply chain, human resources, production, and customer relationship management. SAP systems are widely adopted in industries such as manufacturing, retail, energy, healthcare, and finance.

1.2. Challenges in SAP Process Management

Managing SAP processes requires domain expertise, high accuracy, and compliance with internal and external policies. The major challenges include:

- Complexity of workflows
- Data silos and integration issues
- Manual interventions causing delays and errors
- High costs of customization and support

1.3. Emergence of AI Agents

AI agents are autonomous or semi-autonomous software entities capable of perceiving their environment, reasoning, and taking actions to achieve specific goals. Recent advancements in LLMs and orchestration frameworks have enabled the deployment of AI agents capable of handling complex enterprise workflows, including those in SAP.

* Corresponding author:
ajaverma@yahoo.com (Ajay Verma)
Received: May 17, 2025; Accepted: Jun. 9, 2025; Published: Jun. 13, 2025
Published online at http://journal.sapub.org/computer

2. Overview of Hierarchical AI Architecture

2.1. Definition and Purpose

A Hierarchical AI architecture consists of multiple specialized agents operating under a coordinated framework. These agents collaborate to automate tasks, solve problems, and optimize performance across SAP modules.

2.2. Hierarchical Agent Design

We propose a three-tier hierarchical model:

- **Level 1: Executive Agent (Strategic Decision Makers)**
- **Level 2: Managerial Agents (Process Coordinators)**
- **Level 3: Operational Agents (Task Executors)**

Figure 1. High-Level Architecture for 3 level AI Agent hierarchical model

Each layer performs distinct functions but interacts with others through defined protocols.

3. Level 1: Executive Agents

3.1. Role and Responsibilities

Executive Agents operate at the top of the hierarchy. They are responsible for:

- Strategic decision-making
- Setting global goals and KPIs
- Interfacing with business leaders and interpreting high-level objectives
- Allocating tasks to Managerial Agents

3.2. Example Tasks

- Determining quarterly supply chain optimization goals
- Deciding budget distribution across departments
- Identifying areas for digital transformation within SAP

3.3. Technologies Used

- LangChain orchestrator for prompt chaining
- LLMs like GPT-4 or Claude for strategic reasoning
- Integration with BI tools and SAP Analytics Cloud

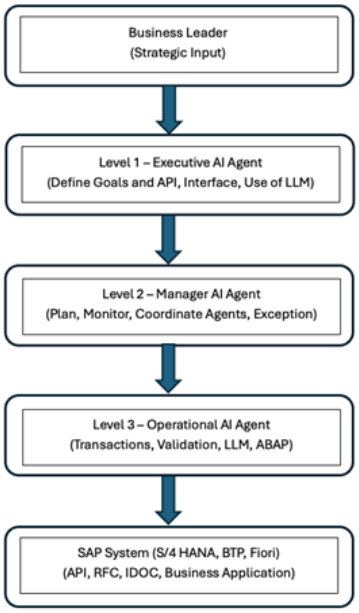

Figure 2. High-Level Reference Architecture – Flow Diagram

Supporting Components:

- LangChain / LangGraph (Agent Orchestration)
- Vector DB (Memory, Embeddings)
- Security: RBAC, Audit Logs
- Integration: SAP BTP APIs, RPA tools

4. Level 2: Managerial Agents

4.1. Role and Responsibilities

Managerial Agents translate strategic goals into executable plans. Their responsibilities include:

- Coordinating with Operational Agents
- Monitoring workflow execution
- Handling exception management
- Ensuring policy compliance

4.2. Example Tasks

- Breaking down a procurement optimization goal into discrete activities
- Scheduling batch jobs in SAP S/4HANA
- Validating compliance in invoice processing

4.3. Technologies Used

- Vector databases for context management
- LangGraph or AutoGen for agent collaboration
- API integration with SAP Business Technology Platform (BTP)

5. Level 3: Operational Agents

5.1. Role and Responsibilities

Operational Agents perform specific, low-level tasks within SAP. These agents:

- Execute transactional activities
- Extract and validate data
- Interact with SAP Fiori apps and GUI
- Log activities for traceability

5.2. Example Tasks

- Creating purchase orders
- Running Material Requirements Planning (MRP)
- Posting journal entries

5.3. Technologies Used

- RPA bots
- Python scripts and ABAP connectors
- LLMs for code generation and SAP note interpretation

6. Prototype Development and Benchmarking

6.1. Prototype Development

To validate the proposed hierarchical AI agent framework, a prototype will be developed using:

- **Environment**: SAP S/4HANA sandbox + SAP BTP
- **Agent Framework**: LangChain + LangGraph + AutoGen
- **LLMs**: GPT-4, Claude, or open-weight models via Hugging Face
- **RPA Layer**: UiPath or SAP Intelligent RPA
- **Interface**: SAP Fiori, SAP GUI Scripting, API integration

Use Case Focus:

Initial development will focus on automating the **Procure-to-Pay (P2P)**.

6.2. Benchmarking Strategy

To assess real-world performance and ROI, key benchmarks will include:

Table 1

Metric	Traditional RPA	Hierarchical AI Agents	Expected Advantage
P2P Cycle Completion Time	30 minutes per cycle	12–18 minutes per cycle	Up to 60% faster
Exception Handling Effort	Manual intervention, 5 min/task	AI-driven resolution in 1 min	80% reduction in manual work
Deployment Time	4–8 weeks (custom scripts)	3–10 days (modular agents)	Faster rollout with fewer errors
Change Adaptability	Hard-coded; slow to adapt	Prompt-driven; LLM-aware logic	Highly flexible to change
Invoice Matching Accuracy	85–90% on average	95%+ with fine-tuned models	Better accuracy in finance ops

Benchmarks will be validated through A/B testing in parallel SAP test environments, using real business scenarios (e.g., onboarding 5 vendors and processing 100 of POs).

7. Technical Implementation

7.1. Agent Orchestration with LangChain and LangGraph

- Chain of Thought (CoT) design for Executive Agents
- Event-driven communication between agents
- Memory management via vector embeddings

7.2. Integration with SAP Systems

- Use of SAP BTP for API access
- RFC and IDoc protocols for legacy systems
- Business Application Studio for extensions

7.3. Data Management

- Secure handling of SAP master data
- Knowledge graphs to model SAP entities
- Role-based access controls for compliance

7.4. High-Level Pseudocode

```
# ---------------------------
# Executive Agent: Strategy
# ---------------------------
def executive_agent():
    objectives = get_procurement_objectives()
    kpis = define_kpis(objectives)
    procurement_plan = create_procurement_strategy(objectives)
    dispatch_to_managerial(procurement_plan)

# ---------------------------
# Managerial Agent: Sales Planning
# ---------------------------
def managerial_sales_agent(procurement_plan):
    # Break down strategy into specific tasks
    task_list = [
        "Validate and create incoming sales order",
        "Coordinate credit checks",
        "Monitor order fulfillment"
    ]
    for task in task_list:
        assign_task_to_sales_agent(task)

# ---------------------------
# Managerial Agent: Procurement Planning
# ---------------------------

def managerial_procurement_agent(procurement_plan):
    # Break down strategy into specific tasks
    task_list = [
        "Initiate vendor selection",
        "Validate budget allocation",
        "Generate purchase requisition",
        "Monitor invoice matching"
    ]
    for task in task_list:
        assign_task_to_operational_agent(task)

# ---------------------------
# Operational Agent: Execution
# ---------------------------

def operational_agent(task):
    if task == "Initiate vendor selection":
        vendors = fetch_approved_vendors()
        shortlist = run_llm_based_ranking(vendors)
```

```
        log_selection(shortlist)

    elif task == "Validate budget allocation":
        budget = query_sap_budget_module()
        if not validate_budget(budget):
            escalate_to_managerial("Budget error")
        else:
            approve_requisition()

    elif task == "Generate purchase requisition":
        data = gather_item_data()
        create_sap_purchase_requisition(data)

    elif task == "Monitor invoice matching":
        match = auto_match_invoice_to_po()
        if not match:
            resolve_with_llm_or_escalate()

    log_task_completion(task)


# ---------------------------
# Orchestration Layer
# ---------------------------

def orchestrate_p2p():
    exec_plan = executive_agent()
    managerial_agent(exec_plan)

# Entry point
if __name__ == "__main__":
    orchestrate_p2p()
```

8. Use Cases

8.1. Procure-to-Pay Automation

- Operational agents create POs and match invoices
- Managerial agents resolve mismatches
- Executive agent optimizes vendor contracts

8.2. Financial Reconciliation

- Operational agent posts entries
- Managerial agent audits transactions
- Executive agent forecasts quarterly margins

8.3. Employee Onboarding

- Operational agents provision SAP access
- Managerial agent ensures compliance training
- Executive agent aligns with HR strategy

9. Benefits

- Enhanced process efficiency
- Reduction in manual errors
- Cost savings from reduced support efforts
- Real-time insights through autonomous reporting

10. Challenges

- Complexity in orchestration and debugging
- Data security and compliance concerns
- Change management in large organizations
- Dependency on model accuracy

11. Prospects

- Incorporation of self-improving agents
- Integration with decentralized SAP landscapes
- Domain-specific LLMs for SAP
- Quantum computing and AI for predictive SAP analytics

12. Conclusions

This hierarchical AI agent framework offers a robust, modular, and scalable approach to SAP automation. By clearly segmenting responsibilities and leveraging advanced technologies, enterprises can automate complex SAP processes more effectively. While challenges remain in implementation and security, the long-term benefits suggest a transformative potential for AI-driven ERP systems.

REFERENCES

[1] SAP Business Technology Platform.

[2] LangChain and LangGraph Official.

[3] UiPath and SAP Intelligent RPA Developer Guides.

[4] IBM Research on Hierarchical Systems in ERP.

[5] OpenAI, Anthropic, and Hugging Face LLM research papers.