Integrating AI agents into the Industry 5.0 ecosystem

Francesco Tufano, Antonio Fresa (Coritel), Ilaria Guarini, Massimo Iovene, Raffaele De Santis (Ericsson Telecomunicazioni S.p.A.), Mario Vento (Università degli studi di Salerno), Francesco Adinolfi (Innovaway S.p.A.)

***Abstract* — This paper presents a comprehensive architecture for simulating smart factory environments using autonomous agents and generative AI. Traditional manufacturing systems face significant challenges in dynamically reconfiguring production environments, often requiring extensive manual programming and downtime during adaptation. In contrast, the proposed system dynamically configures production units (assemblers, packagers, inspectors) based on user-defined throughput requirements. A central mother robot, the Niryo Ned robot, similar to those used in Amazon warehouses, is tasked with transporting these units to their designated locations, enabling dynamic reallocation of resources on the factory floor. Our novel contribution lies in the integration of multiple specialized AI agents through the CrewAI framework, creating a collaborative system that autonomously responds to changing production requirements without human intervention. The architecture orchestrates domain-specific agents such as a Programmer Analyst, Python Coding Expert, and Niryo Workspace Expert to analyze, generate, and apply code changes in real time. To implement and test this system, a physical Ned robot connected to a 5G network is employed, demonstrating real-world applicability in factory settings. Results are rendered through a dynamic web UI featuring real-time messaging and execution logs, while configuration data is persistently managed. This research highlights the synergistic application of Retrieval-Augmented Generation (RAG), LangChain tools and Large Language Models (LLMs) to build explainable, adaptable, and interactive AI-driven manufacturing workflows.**

Keywords* — *Smart Factory, CrewAI, LLMs, Agent-based Programming, Flask, Retrieval-Augmented Generation

I. INTRODUCTION

The emergence of Industry 5.0 represents a paradigm shift in manufacturing, emphasizing human-machine collaboration, sustainability, and adaptability. While Industry 4.0 focused on digitization and automation, Industry 5.0 demands systems that can dynamically reconfigure in response to changing production requirements without extensive human intervention. Traditional approaches to factory automation typically require manual reprogramming and significant downtime when production needs change, creating a critical efficiency gap in modern manufacturing environments. The widespread adoption and commercial impact of ChatGPT catalyzed a global acceleration in LLM research and development, propelling these models into a diverse array of strategic industrial applications. The release of Gemini marked a further leap, introducing multimodal processing and enhanced context management, thus reinforcing the feasibility of autonomous, intelligent systems at scale. This technological trajectory has enabled the emergence of AI agents [1] autonomous software entities underpinned by LLMs. These agents do not merely automate predefined tasks; rather, they embody a novel paradigm in which systems can autonomously interpret complex, domain-specific instructions, perform sophisticated operations with minimal human intervention, and adapt workflows dynamically in response to environmental changes. The core functionality of these agents is structured around a cyclical **observe-plan-act** model. When embedded in physical environments, agents can interface directly with sensors, robotic systems, and industrial machinery blurring the lines between digital and physical domains. This physical AI capability enables autonomous agents not only to monitor and optimize real-world operations (e.g., in smart factories or logistics) but also to perform physical actions such as manipulating objects, navigating spaces, or collaborating with humans in shared workspaces. However, several significant challenges remain in implementing truly adaptive manufacturing systems. Current smart factory implementations often rely on pre-programmed routines with limited ability to respond to dynamic changes. The integration of physical robotics with AI decision-making systems remains fragmented, requiring specialized expertise across multiple domains. Additionally, there is a notable absence of frameworks that enable non-technical manufacturing personnel to reconfigure production flows without deep programming knowledge. To address these gaps this study introduces a novel framework that integrates user-guided input with LLM-powered AI agents within a simulated smart factory environment. Unlike traditional static systems, the proposed solution features a coordinated team of agents that iteratively optimize factory configurations. These agents dynamically update operational scripts in real time and interact with users via a conversational interface, enabling immediate feedback and adaptive control of industrial processes.

II. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

This section outlines the workflow adopted to implement the proposed smart factory simulation architecture, describing the systematic selection and integration of technologies to meet specified functional and performance objectives. The system concerns the configuration of a central "mother robot" (Niryo Ned) responsible for physically relocating operational resources (assemblers, packagers, inspectors) within the simulated factory environment, following behavior patterns like those used in Amazon's automated warehouses. The system employs specialized autonomous agents operating in a coordinated manner: the **Programmer Analyst agent** analyzes user input to extract application-level logic; the **Python expert agent** is responsible for interpreting high-level manufacturing requirements (e.g., "increase throughput by 30%") into logical modifications needed in the production configuration, so bridging the gap between user needs and technical implementation. A third agent, the **Workspace Expert agent**, manages the physical workspace configuration, including spatial positioning of production units, validating feasibility of proposed layouts, and ensuring optimal resource allocation**.** The **Routing agent** serves as the system orchestrator, directing user inputs to appropriate specialized agents and maintaining the overall workflow sequence. Through the RAG tool, agents can access existing scripts, retrieve relevant components, and dynamically generate code that adheres to the spatial and functional constraints of the simulated environment. **CrewAI** [2] was selected due to its straightforward configuration, robust functionality, and ease of integration. For implementing the RAG approach [3], **LLamaIndex** was specifically chosen for its simplicity of setup, efficient retrieval capabilities, and seamless integration capabilities with the other selected technologies. At the core of the architecture, the agent framework operates by organizing multiple specialized autonomous agents into a coordinated CrewAI "crew," where each agent is assigned a distinct role and objective. Agents interact with each other through a structured task management system, collaborating intelligently to achieve higher-level goals without centralized micromanagement. Tasks are dynamically allocated and executed either sequentially or hierarchically depending on the process definition, allowing complex problem-solving workflows to emerge through decentralized agent cooperation. This decentralized but organized structure enables flexibility, scalability, and robustness in managing real-time user demands and operational reconfigurations.

III. Experimentation

The experimentation phase was conducted to validate the proposed system in a realistic, network-connected environment simulating a smart factory scenario. Central to the experimentation was the integration of a physical collaborative robot — **Ned by Niryo** [4] — connected via **5G network** to the backend system, enabling real-time, low-latency control and feedback. The robot served as the physical executor of tasks generated by the AI-driven planning pipeline, specifically those involving the relocation of components such as assemblers, packagers, and inspectors on a test bench replicating factory floor dynamics. The experimentation followed a structured workflow, where the user initiated a production configuration. During the experimentation phase, a key component of the system was the implementation of a conversational user interface designed to abstract technical complexity. Users interacted with the system by issuing natural language commands such as "increase packaging throughput by 30%" which were parsed by the Route Agent and forwarded to the appropriate specialized AI agents. This allowed non-technical personnel to dynamically influence production logic without writing code or understanding the underlying architecture. The interface was embedded in the web UI, enabling real-time dialogue and feedback between the user and the AI agents. This interaction triggered a POST request to a Flask-based backend API, which launched the CrewAI multi-agent pipeline. The Route Agent acted as the coordinator, parsing the incoming configuration and orchestrating the execution of downstream agents. The workflow of agents was triggered, i.e. each agent per role could accomplish its specific task, like logic extraction, code adaptation, and workspace reconfiguration. A preview of reconfigured workspace was provided over the GUI to enable human feedback before proceeding with deployment on the robot. The resulting Python code was then executed through Ned's API, allowing the robot to perform the task. These scripts were deployed via a dedicated Python module interfacing directly with the robot using the PyNiryo API. However, given the complexity of the setup, the performance is not easily measurable. After repeated system executions we encountered a failure rate of about 20%. Main reasons are related to the low quality of the object recognition caused by poor environmental conditions like lighting settings and medium resolution of embedded cameras. Furthermore, some motors kept shutting off not by wrong inputs of generated code, but by unstable voltages. However, it should be noted that on several occasions, the CrewAI agents did not generate correct software configurations. This was mainly due to the limited accuracy of small-sized, open-source LLMs that are not golden models, which significantly restricts the overall quality of the generated configuration.

This study does not present a quantitative evaluation of the proposed architecture due to the lack of a real-world baseline and an industrial partner with relevant domain expertise. Without such collaboration, it was not possible to gather comparative data or apply standard metrics. Consequently, the work remains a theoretical assessment of the approach. Quantitative validation is planned for future work in partnership with an Industry 5.0 stakeholder.

IV. Conclusion and Potential Impacts

This agent-based smart factory system demonstrates how modular, autonomous AI agents can collaboratively handle domain-specific inputs, adapt in real time, and execute complex tasks with minimal human intervention. By integrating these agents with robotics over 5G, it creates a seamless connection between digital intelligence and physical operations within a highly adaptive architecture. The project aimed to advance expertise in AI agent methodologies. Looking ahead, AI agents are expected to operate with greater autonomy, enabling hyper-personalized services, self-healing systems, and broader access to development tools for non-experts. They are reshaping software development through AI-first architectures and autonomous DevOps. Their societal impact is profound, revolutionizing industries by automating workflows, enhancing efficiency, and unlocking human creativity. AI agents are tackling challenges in healthcare, education, and sustainability by analyzing large datasets and offering insights. In global markets, they provide strategic advantages and help bridge skill gaps. Across sectors (e-commerce, manufacturing, healthcare, finance, HR, IT, and marketing) AI agents are transforming operations, driving innovation, and redefining human-machine collaboration. Their rise marks a fundamental shift in how we design and interact with intelligent systems.

References

[1] Autonomous Industrial Control using an Agentic Framework with Large Language Models https://arxiv.org/abs/2411.05904?utm_source=chatgpt.co

[2] Duan, Z., & Wang, J. (2024). *Exploration of LLM Multi-Agent Application Implementation Based on LangGraph+CrewAI*. arXiv preprint arXiv:2411.18241. https://arxiv.org/abs/2411.18241

[3] Lewis, P., Perez, E., Piktus, A., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33, pp. 9459–9474. https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[4] Niryo robot: https://niryo.com/