Building Digital Twins for semiconductor manufacturing using AWS and Generative AI

Digital twins are transforming industrial enterprises, offering a powerful tool to drive data-driven decision making and optimize operations. A digital twin is a living digital representation of a physical system that is dynamically updated with data to precisely mimic the true structure, state, and behavior of that system.

In the semiconductor and high-tech manufacturing sectors, which face immense complexity and capital-intensive production, digital twins are emerging as a critical technology. These virtual models ingest data from a variety of sources, including sensors, SCADA systems, and production systems, and combine it with 3D CAD models to create comprehensive digital representations. By leveraging the scalability, reliability, and advanced analytics capabilities of the AWS cloud, manufacturers can build and manage these digital twins at scale, integrating and contextualizing data from disparate IT and OT sources. By constructing detailed scenarios of real-world manufacturing situations, these intelligent virtual models can serve as an early-warning system, forecasting events and their likelihood before they occur. They also provide a digital test bed for evaluating strategies, optimizing schedules and sequences, and improving equipment performance - driving substantial gains in efficiency, quality, and time-to-market.

Building a digital twin, especially for highly specialized applications (such as multimachine production scheduling or vehicle routing), can be time-consuming and resource-intensive. The effort often entails designing and developing new digital-twin models, a process that can take six months or longer and incur substantial labor, computing, and server costs. Through the seamless integration of AWS services like IoT TwinMaker and generative AI capabilities, semiconductor and high-tech companies can build and deploy sophisticated digital twins that deliver tangible operational and business benefits. Large language models (LLMs) can create code for the digital twin, accelerating the development process and increasing effectiveness. This ability to generate such output leads to an exciting prospect: LLMs could possibly be used to create a generalized digital-twin solution. These generalized digital wins can then be augmented using other Generative AI techniques and further enhance the value of these digital twins.

AWS IoT TwinMaker helps developers create digital twins of real-world systems like buildings, factories, equipment, and production lines. It provides tools to build digital twins that can optimize operations, increase production, and improve equipment performance. IoT TwinMaker allows using existing data from multiple sources, creating virtual representations of physical environments, and combining 3D models with real-world data to enable a holistic view of operations faster. AWS IoT SiteWise is a managed service that makes it easy to collect, store, organize and monitor data from industrial equipment at scale. This helps make better data-driven decisions by monitoring operations across facilities, quickly computing industrial performance metrics, and creating applications that analyze equipment data to prevent issues and reduce production gaps. The managed industrial Knowledge Graph in AWS IoT TwinMaker gives you the ability to model complex systems and create Digital Twins of your physical systems.

Furthermore, Generative AI unlocks new ways to make data more accessible and easier to use for end users like shop floor operators and managers. You can now leverage natural language to ask AI complex questions, such as identifying standard operating procedures to fix production problems or getting suggestions for potential root causes based on observed alarms. Amazon Bedrock [4] is a managed service designed to simplify the development and scaling of Generative AI applications. It makes it easier for builders to create and manage these powerful AI-powered applications

The purpose of this paper is to show various use-cases where Generative AI can be used to create or augment a digital twin to help diagnose and resolve manufacturing production issues.

Data Producer

Digital twins rely on large volumes of often real-time data from diverse sources, which can be challenging to manage effectively. However, large language models (LLMs) possess advanced "embedding" capabilities that enable significant data compression while preserving essential information. This empowers efficient data transfer and processing within digital twin systems. For instance, in a manufacturing setting, generative AI could organize data from maintenance logs, equipment imagery, and operational videos. The digital twin could then analyze this structured data, identify patterns or anomalies that may not be evident from unstructured data alone, and use those insights to inform decision-making and predictive maintenance strategies.
Furthermore, generative AI tools can supplement the training data used by digital twins by generating synthetic data. If an existing data set, such as maintenance logs, lacks a particular type of defect, generative AI could create a synthetic data set that includes that defect. This helps ensure the digital twin is trained to accurately recognize that defect in the future.

Digital Twin Code Generation

A generative digital twin agent could take care of the integration through APIs of the model and the data. Acting as a 'generator', it significantly accelerates the implementation phase of digital twins, offering organizations a simpler approach to adopting this technology. Large language models (LLMs) have the capability to generate code for digital twins, which can speed up the development process and increase efficiency. This suggests that LLMs could be used to create a generalized, universal digital twin solution. The architecture of a digital twin can be represented as a time series graph with nodes and edges. Graph-based LLMs can create a basic model of this digital twin structure, which can then be further developed and adapted for different scenarios and applications. In essence, LLMs have the potential to serve as a starting point for developers to build digital twin solutions across a variety of projects and industries.

AI Assistant Interface

Large language models (LLMs) can generate comprehensive simulation scenarios for digital twins to explore. In this way, LLMs can serve as an intuitive interface for interacting with digital twin simulators. Users can communicate with the digital twin using natural language - for example, asking questions and receiving understandable insights in response. This approach simplifies interactions with complex digital twin systems, making them more accessible to users without extensive technical expertise. By facilitating natural language-based communication, it enables data-backed decision making that is easier for a wider range of stakeholders to understand and utilize. In essence, LLMs help bridge the gap between the technical complexity of digital twins and the needs of end-users, empowering more people to leverage the insights generated by these virtual representations of real-world systems.

Below is an example of a AI Assistant for a cookie manufacturing factory that uses Large Language Model (LLM) Agent that responds to chat user interface (UI). The LLM Agent is implemented using the LangChain framework [5]. LangChain is a flexible library to construct complex workflows that leverage LLMs and additional tools to orchestrate tasks to respond to user inputs. Amazon Bedrock provides high-performing LLMs needed to power this solution, including Claude from Anthropic and Amazon Titan. To implement the retrieval augmented generation (RAG) pattern, an open-source in-memory vector database such as Amazon OpenSearch Service can be used. To help the AI Assistant better respond to the user's domain specific questions, here we ground the LLMs by using the Knowledge Graph feature in AWS IoT TwinMaker [6] and user provided documentation (such as equipment manuals stored in Amazon S3). We also use AWS IoT SiteWise [7] to provide equipment measurements, and a custom data source implemented using AWS Lambda to get simulated alarm events data that are used as input to LLMs and generate issue diagnosis reports or troubleshooting suggestions for the user.

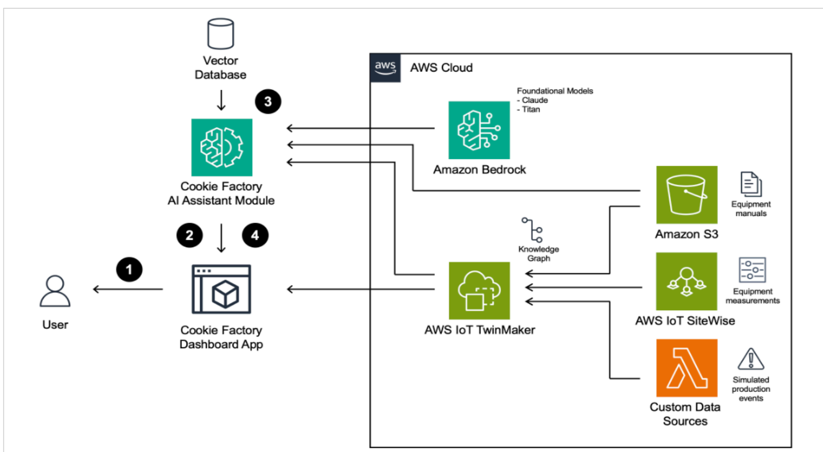

Figure 1: Factory Floor AI Assistant Architecture

The AI Assistant chooses different strategies to answer a user's questions. This allows it to use additional tools to generate answers to real-world problems that LLMs cannot solve by themselves. A typical user interaction flow can be described as follows: The user requests the AI Assistant in the dashboard app. The dashboard app loads the AI Assistant chat UI. The user sends a prompt to the AI Assistant in the chat UI. The LLM Agent in the AI Assistant determines the best workflow to answer the user's question and then executes that workflow. Each workflow has its own strategy that can allow for the use of additional tools to collect contextual information and to generate a response based on the original user input and the context data. The response is sent back to the user in the chat UI. Following is the digital twin of a factory floor.

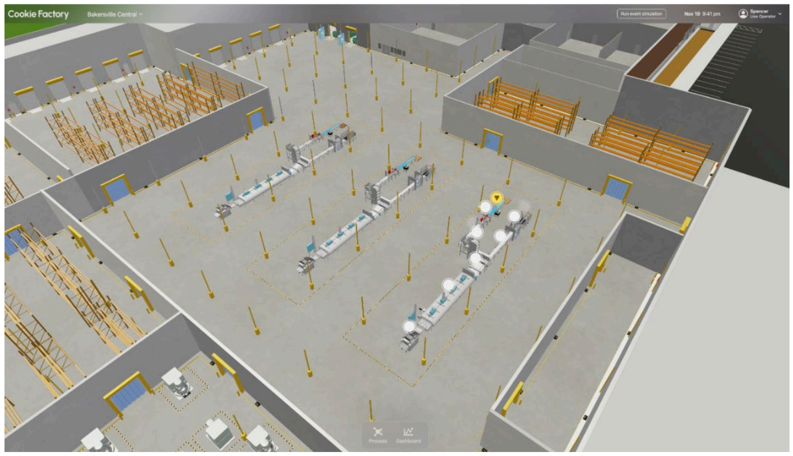

Figure 2: Digital Twin of factory floor.

The dashboard will display an alert, indicating there are more than expected deformed cookies produced by one of the cookie production lines. When the alarm is acknowledged, the AI Assistant panel will open. The event details are passed to the AI Assistant so it has the context about the current event. You can click the "Run Issue Diagnosis" button to ask AI to conduct a diagnosis based on the collected information. Once the diagnosis is completed, the AI Assistant will suggest a few potential root causes and provide a button to navigate to the site of the issue in the 3D viewer. Clicking on the button will change the 3D viewer's focus to the equipment that triggers the issue. From there you can use the Process View or 3D View to inspect related processes or equipment. You can use the AI Assistant to find SOPs of a particular equipment. Try asking "how to fix the temperature fluctuation issue in the freezer tunnel" in the chat box. The AI will respond the SOP found in the documents associated to the related equipment and show links to the original document.

Figure 3: AI Assisted initial issue diagnosis

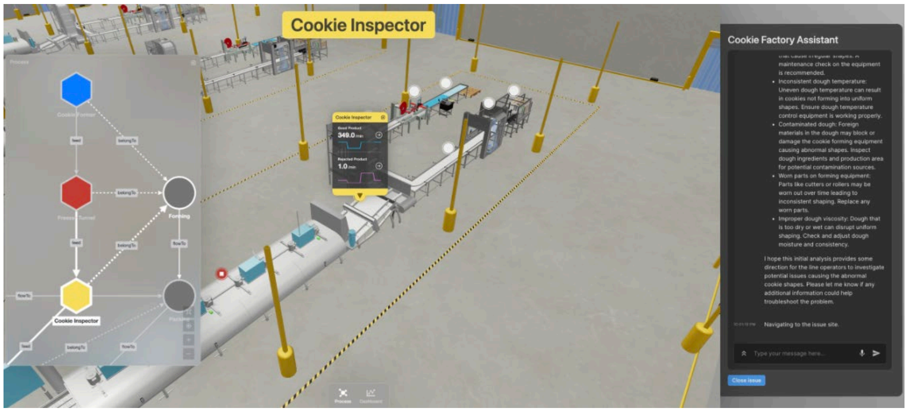

Figure 4: AI Assisted SOP creation

Figure 5 shows a high-level execution flow that represents how user input is routed between multiple LLM Chains to generate a final output.

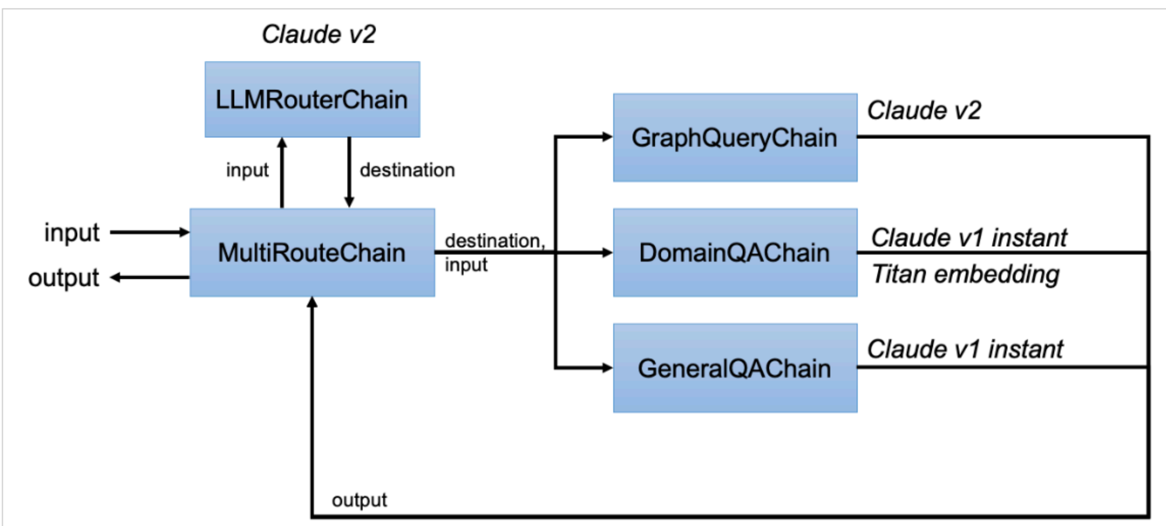

Figure 5: High-level execution flow of LLM agent.

Data Consumers

Generative AI enhances digital twin capabilities by using real-time data from the digital twin to provide richer context for its inputs and outputs. The secure, data-rich environment of the digital twin allows Generative AI to "learn" and expand the scope of what it can do. Digital twins also enable "what-if" simulations that help fine-tune Generative AI for predictive modeling, going beyond the typically backward-looking nature of large language models. Moreover, the constraint engine in digital twins can validate Generative AI outputs to ensure they adhere to physical limits and other parameters, boosting the accuracy of Generative AI. For example, Generative AI-generated code for machinery can be validated by the digital twin before deployment. While still an emerging capability, the synergy between Generative AI and digital twins holds immense potential for a wide range of use cases in the future

In conclusion, the integration of AI and digital twin technologies is crucial for developing systems that accurately mirror their physical counterparts, while also being predictive and adaptive. By combining these two capabilities, a dynamic model can be achieved that learns, evolves, and provides actionable intelligence to inform real-world decisions. The development of digital twin capabilities has accelerated rapidly in recent years, with AI and machine learning algorithms reinforcing and broadening the horizons of both technologies. This has enabled numerous use cases, such as real-time monitoring, predictive maintenance, and scenario planning. The strong collaborative link between AI and digital twins is essential for creating comprehensive, intelligent systems that can represent and predict the behavior of their physical counterparts.

References:

[1] Building an AI Assistant for Smart Manufacturing with AWS IoT TwinMaker and Amazon Bedrock: https://aws.amazon.com/blogs/iot/building-an-ai-assistant-for-smart-manufacturing-with-aws-iot-twinmaker-and-amazon-bedrock/

[2] Mckinsey, Digital twins and generative AI: A powerful pairing https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-forward/digital-twins-and-generative-ai-a-powerful-pairing

[3] Knowledge graph https://docs.aws.amazon.com/iot-twinmaker/latest/guide/tm-knowledge-graph.html

[4] Amazon Bedrock - https://aws.amazon.com/bedrock/

[5] Langchain - https://www.langchain.com/

[6] AWS TwinMaker - https://aws.amazon.com/iot-twinmaker/

[7] AWS IoT SiteWise - https://docs.aws.amazon.com/iot-sitewise/latest/userguide/what-is-sitewise.html