OPERATIONAL TECHNOLOGY DEEP DIVE

Beyond Chatbots: How Protocols are Making AI Agents Work on the Shop Floor

Moving from theoretical conversation to standardized operational control.

Supply Chain · AI Governance · OT/IT Convergence · Enterprise Architecture

> The newsletter for operations leaders who prioritize reliable system integration over flashy new features.

The Integration Problem: When AI Gets Stuck Between Systems

There is a significant gap between the polished demos of conversational AI and the gritty reality of an ERP or WMS system running your shop floor. For most manufacturing leaders, "integration" isn't just about making two pieces of software talk; it’s about ensuring that data moves accurately from a physical action—like a pallet being moved or a part being scanned—into the digital record without human intervention or manual correction.

Right now, many organizations are trying to bridge this gap with standard chatbots. They want an operator to be able to ask, "Where is order #402?" and get an immediate answer. While that sounds like progress, it often creates a new silo. The AI isn't actually integrated into the workflow; it’s just sitting on top of the data, acting as a middleman that doesn't have the authority or the "pipes" to change anything.

When you try to force an AI to interact with legacy systems without a standardized way to do so, you end up with brittle connections. You find yourself writing custom code for every new tool the team wants to use. It’s not scalable. If your system is a mess of disconnected databases and manual

workarounds, simply adding a chat interface won't fix the underlying rot. The problem isn't that the AI can't "understand" the question; it's that the AI doesn't have a reliable way to reach into the machinery of the business to pull out the truth or push in a command.

What Operational Agents Actually Need to Do

In an industrial environment, we need to distinguish between a "chatbot" and an "agent." A chatbot answers questions; an agent executes tasks. For a tool to be useful on your floor, it has to do more than provide information—it must participate in the operational flow.

To be truly useful for a production manager or a warehouse lead, an AI agent needs three specific capabilities:

1. **Read:** It must pull real-time data from your inventory systems, maintenance logs, and shipping manifests without someone having to copy and paste it into a prompt.
2. **Write:** It must have the ability to update those records—marking a job as complete, updating a stock level, or flagging a quality deviation in the system of record.
3. **Execute:** It must be able to trigger actions within your existing workflows, such as sending an automated alert to maintenance when a machine hits a specific cycle count or generating a draft purchase order based on current scrap rates.

We aren't looking for a digital assistant that can write polite emails; we are looking for a tool that follows a standard operating procedure (SOP). If the agent cannot interact with your WMS or ERP through a stable, predefined path, it isn't an "agent"—it’s just another screen for the operator to look at. It must be able to move from "What is happening?" to "Do this thing" without human translation in the middle.

Why Chatbots Are Not Enough for Operations

The failure here is **The Conversation Trap**. Many leaders assume that because an AI can hold a coherent conversation, it is ready to manage shop floor operations. This is not true. A chatbot provides information; it does not have "tools."

Without a standardized way to connect the AI's logic to your specific business tools, the system remains trapped in a loop of conversation rather than action. It can tell you that inventory is low, but it doesn't know how to check the warehouse management system (WMS) or trigger a reorder because it hasn't been given the "handle" to do so.

The Conversation Trap	Operational Reality
Chatbot: Answers questions about current stock levels based on a static data pull.	**Agent:** Queries the WMS in real-time and updates the status of an incoming shipment.
Chatbot: Suggests that a machine might need maintenance because it "sounds" like it's struggling.	**Agent:** Monitors vibration sensors and automatically creates a work order in the CMMS.

The Conversation Trap	Operational Reality
Chatbot: Provides instructions on how to perform a specific task.	**Agent:** Logs the completion of a step into the manufacturing execution system (MES).

A chatbot is an information layer. An agent requires an operational layer. If you only build the first one, your team will eventually realize that they still have to do all the manual data entry themselves just to make the "smart" system work.

The Protocol Solution: Standardizing AI Interactions with MCP

To move from a chatbot to an agent, we need a way to standardize how these systems talk to each other. This is where the Model Context Protocol (MCP) comes in. Think of MCP not as a complex piece of software you have to master, but as a universal "plug" for your enterprise data.

Currently, if you want an AI to interact with three different pieces of equipment or two different software platforms, you often have to build three different bridges. It’s custom work every time. MCP changes that by providing a standard way for the AI to find and use tools. Instead of writing specific code for "How to talk to our ERP," you implement an MCP server that tells the AI: "Here is how you access the inventory, here is how you read the production schedule, and here are the rules for updating them."

By using a protocol like MCP, you aren't just making it easier for the AI; you are creating a standardized interface for your own team. It moves the integration out of the "chat" window and into the infrastructure. This means that when an operator asks a question or gives a command, the system doesn't have to guess how to find the answer or execute the task. It follows a predefined path that you have already validated and secured. It turns the AI from a free-form conversationalist into a disciplined executor of your existing processes.

Designing for Operational Governance (MCP in Practice)

When we move toward agentic behavior, the conversation shifts from "Can it do this?" to "Should it be allowed to do this?" In any manufacturing environment, an unauthorized action on a piece of equipment or a change in a shipping record can have real-world consequences. You cannot allow an AI to have "free reign" over your systems.

Governance means defining the boundaries of what the agent is permitted to touch. When implementing MCP, you must define exactly which tools are available to the AI and what permissions those tools carry. For example:

- **Read-Only Tools:** The AI can check stock levels or production status (low risk).
- **Actionable Tools:** The AI can update a work order but requires a human "OK" before it is finalized (medium risk).

- **Critical Systems:** The AI cannot touch machine parameters or safety protocols without direct, authenticated human override (high risk).

You must also decide who owns the "tool definition." If an agent makes an error—for example, if it misinterprets a command and puts in an incorrect quantity for a raw material order—the system needs a clear path for correction. This isn't just about software; it’s about your standard work. Every action the AI takes must be traceable back to a specific protocol you approved. You aren't delegating authority to the machine; you are giving the machine permission to execute an authorized process that a human would otherwise have performed manually.

Three Steps to Moving Beyond Conversation

If you want to move past the "chat" phase and toward actual operational utility, don't start by trying to make your AI smarter. Start by making your data more accessible through standardized pathways.

1. **Audit Your Interaction Points.** Identify three specific tasks where your team currently spends too much time moving data between systems (e.g., copying a serial number from a paper log into an ERP). These are your first candidates for "agentic" automation. Don't start with the most complex problem; start with the one that causes the most repetitive friction.
2. **Define Your Toolset.** Instead of trying to give the AI access to everything, define what it *needs* to do those three tasks. Map out the specific data points and "write" permissions required. This is your blueprint for building MCP-compliant connectors. You are defining the boundaries of its world before you let it inside.
3. **Pilot with Low-Risk Actions.** Start by giving the AI permission to perform actions that have a low cost of failure, such as updating internal status notes or drafting emails based on production data. Only move toward "high-stakes" commands—like changing machine set points or confirming final shipments—once you have proven the reliability of the underlying protocol over several months of operation.

The goal is not to replace your people with AI; it is to remove the "data tax" that keeps your skilled workers from doing their actual jobs. By moving toward a protocol-based system, you ensure that when the technology finally delivers on its promise, it does so within the guardrails of your established quality and safety standards.

Call to Action

What is the single biggest integration point holding back your digital transformation right now? Share this diagnosis with another leader who needs it.

Newsletter replies: newsletter@sixsigmakaizen.com · X: @kaizen_6sigma

Stay curious, stay improving,

David Rodgers

SixSigmaKaizen.com

References: Model Context Protocol and the Future of Agentic Supply Chains