DEVWKS-2858 - Unleashing the Power of GenAI for Cross-Platform Analytics and AIOps

Joel Jose (joeljos@cisco.com),

Solutions Engineer,

Cisco

AIOps: AI-Driven IT Operations

📌 **Definition:**
AIOps refers to the application of **AI, machine learning (ML), and big data analytics** to automate and enhance **IT operations**, improving performance, monitoring, and incident response.

🔷 **Key Capabilities:**
✅ **Automated Monitoring** – Real-time anomaly detection
✅ **Predictive Analytics** – Forecasts issues before they occur
✅ **Incident Resolution** – AI-driven root cause analysis & remediation
✅ **Event Correlation** – Reduces noise by grouping related alerts
✅ **Self-Healing Systems** – Automates responses to common failures

Types of AI

Generative AI

Synthesize signal to improve user productivity and outcomes

Eg. Automated network troubleshooting and incident resolution

Foundational AI

Make sense of the signal in vast amounts of data

Eg. Predictive analytics, can forecast network failures before they occur

Agentic AI: The Next Evolution in AI Autonomy

📌 **Definition:**
Agentic AI refers to AI systems that can **autonomously plan, make decisions, and take actions** toward achieving goals with minimal human intervention.

🔷 **Key Characteristics:**
✅ **Autonomy** – Operates independently based on objectives
✅ **Adaptability** – Learns and adjusts strategies dynamically
✅ **Proactivity** – Initiates actions rather than just responding
✅ **Multi-step Reasoning** – Breaks down complex tasks into smaller steps
✅ **Real-world Impact** – Can interact with systems, humans, and environments

Ways to Build Agentic AI

Custom Development (Full Control)

Full Control: Building from the Ground Up

- Python + LangChain/LlamaIndex (Memory & Retrieval)
- Python + OpenAI API (Function Calling, Tool Use)
- Python + AutoGPT/BabyAGI/SuperAGI (Autonomous Agents)
- Python + Reinforcement Learning (Gym, RLHF)
- *Python + Custom Logic (Optimized for Efficiency & Control)*

Frameworks & Libraries (Modular Tools)

Modular Tools: Accelerating Development

- LangChain/LlamaIndex (Agent Frameworks)
- CrewAI (Multi-Agent Collaboration)
- Haystack (NLP Agent Framework)
- Hugging Face Transformers (Custom Workflows)

No-Code & Enterprise AI (Business & Automation)

Business & Automation: Empowering Users

- OpenAI GPTs | Microsoft Copilot | Zapier AI
- AWS Bedrock Agents | Google Vertex AI | Salesforce Einstein

Autonomous & Multi-Agent Systems & Game AI

Advanced AI: Autonomy, Collaboration, & Games

- AutoGPT/BabyAGI/SuperAGI (Autonomous Agents)
- Meta CICERO (Negotiation & Diplomacy AI)
- AgentGPT (Browser-based AI Agent)
- Unity ML-Agents/OpenAI Gym (Game AI)
- DeepMind AlphaZero/MuZero (Self-Learning AI)

The Power of Custom Agent Development: Control, Efficiency, and Maintainability

Full Control: Tailor every aspect of your AI agent with direct OpenAI API and Python integration. No framework limitations.

Peak Efficiency: Optimize performance, reduce costs, and minimize token usage for production deployments.

Deep Understanding: Build from scratch for better troubleshooting, debugging, and innovation.

Maximum Flexibility: Integrate seamlessly with any system or data source.

Long-Term Stability: Maintainable codebase, reduced dependency risks, ensuring long-term viability.

Strategic Scaling: Prototype with large models, then deploy smaller, cost-effective on-premise solutions.

Automating Network Remediation: A Step-by-Step Breakdown

Alarm Intake & Aggregation:

- Collect alarms from various sources (e.g., Catalyst Center, SD-WAN Manager, ping results).
- Combine and standardize alarm data into a unified JSON format

Agentic Response Generation (Initial):

- For each alarm, generate an initial agentic response containing key information:
- Alarm Source, Summary, Device Family, Classification, etc.
- Recommended Actions, Tracking Next Steps, Detailed Information, Insights

Workflow Invocation & RCA Trigger:

- Based on alarm classification (e.g., "Urgent"), trigger relevant workflows (e.g., Webex notification).
- Initiate Root Cause Analysis (RCA) for unresolved alarms.

Device Connection & Command Execution:

- Determine the most relevant device for troubleshooting based on alarm data and network topology.
- Execute pre-defined "show" commands on the device to gather diagnostic information.

Iterative RCA Loop:

- Continuously monitor telemetry data (e.g., ping results).
- Use LLM (e.g., Llama) to analyze alarms, show results, and telemetry data.
- Identify the most probable root cause (RCA) using LLM reasoning.
- Generate CLI commands for remediation based on the RCA and runbook instructions.
- Execute remediation commands on the device.

Alarm Resolution & Summary:

- Use LLM to determine if the alarm is resolved based on telemetry data.
- Generate a comprehensive summary of the entire process:
- Original alarm details, RCA findings, remediation steps taken, final outcome.
- Update the alarm status and insights.

Notification & Logging:

- Send notifications (e.g., Webex) with the final summary.
- Log all actions and results for auditing and future analysis.

https://github.com/Cisco-Global-Partner-Engineering/AI-enabled-Outcomes

LinkedIN

https://tinyurl.com/joeljose420

THANKS