

hardware performance and educational scenarios. The perspective mode of Quest 3 allows users to observe the real environment at any time during virtual tasks, avoiding the safety hazards caused by complete immersion; second, the structured design of gamified educational content. The system breaks down learning goals into interactive games. For example, the "time management" module visualizes the time consumption of tasks through a virtual hourglass, and the "resource coordination" training simulates the use conflicts of shared equipment. Students naturally master the priority decision-making method in the process of resolving conflicts. Third, the scalable architecture design. The system adopts a modular resource loading solution. Users can replace learning content and task logic according to their needs, such as replacing chess with language learning, without reconstructing the underlying code. This flexibility enables the system to quickly adapt to the learning management needs of different courses.

In summary, this study built a simulation game system that supports student self-management by integrating the hardware features of Quest 3 and mixed reality technology. The advancement of this project is also to further verify the universality of mixed reality in complex educational scenarios – when technology is no longer limited to the demonstration of specific knowledge points, but builds a complete behavioral training ecosystem, it will have a more profound impact on the shaping of students' cognitive patterns and comprehensive abilities.

In the future, as technologies such as light field display and neural tactile feedback mature, mixed reality is expected to break through the limitations of existing interactive dimensions. For example, by simulating the difference in touch of objects of different materials, students can feel the temperature changes of chemical reactions in virtual experiments; or capture the state of attention through brain-computer interfaces and dynamically adjust the presentation rhythm of teaching content. These technological evolutions will continue to expand the boundaries of possibilities for MR education. The current system development based on Quest 3 is precisely the technical reserve for more complex forms of education.

Reference

1. Website «Meta Quest 3 Developer» [Electronic resource]. – Access mode: <https://developer.oculus.com> – Access date: 03.03.2025.
2. Website «Microsoft Mixed Reality Toolkit (MRTK)» [Electronic resource]. – Access mode: <https://github.com/microsoft/MixedRealityToolkit-Unity> – Access date: 20.03.2025.

UDC 004

ANALYSIS OF AUTOMATED WORKFLOW ORCHESTRATION TECHNOLOGIES

**Chen Bei, master's degree student, Kazakou V., PhD in Engineering, associate professor,
Sokalava H., senior lecturer**

Vitebsk State Technological University, Vitebsk, Republic of Belarus

Abstract. The article provides an overview of one of the workflow orchestration management systems. The article describes the basic structure and functionality of the orchestration system. A conclusion is made about the ease of use of the described system.

Keywords: orchestration, workflow automatization, enhancement with AI.

Automated workflow orchestration is a technology that logically connects dispersed tasks, systems, and services through software tools, aiming to automate end-to-end business processes and enable dynamic optimization. Its core purpose is to reduce manual intervention, improve efficiency, and adapt to flexible adjustments in complex business scenarios.

Compared to traditional Robotic Process Automation (RPA) – which primarily simulates user interface operations (e. g., button clicks or form filling) – workflow orchestration emphasizes deeper integration between systems. For example, it can directly invoke APIs of internal ERP systems, cloud service interfaces, or databases to enable cross-platform data flow and logic control. Concurrently, workflow orchestration complements Business Process Management (BPM): BPM focuses on process modeling and analysis, while orchestration acts as BPM's "execution engine," translating abstract workflows

into actionable task scheduling.

From a technical evolution perspective, early workflow automation relied on scripting (e. g., Shell scripts or Cron jobs) and could only handle simple linear processes. As enterprises' digital demands grew, visual design tools (e.g., Zapier, n8n) became widespread, enabling non-technical users to quickly build automated workflows via drag-and-drop interfaces. In recent years, the integration of AI has further revolutionized this field. For instance, machine learning predicts task duration, while large language models (e. g., ChatGPT) generate complex logic code, significantly enhancing the intelligence of orchestration systems.

Core Components and Key Technologies

A complete automated workflow system typically comprises five core modules (Fig. 1).

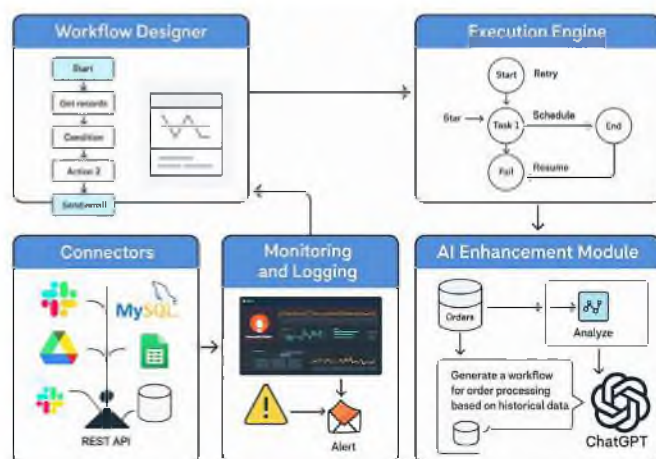


Figure 1 – Automated workflow system core modules

1. **Workflow Designer:** This is the user interface for system interaction, offering a visual drag-and-drop modeling environment. For example, Microsoft's Power Automate allows users to connect applications like Salesforce and Teams via graphical nodes, defining end-to-end logic from data synchronization to approval workflows. Developer-friendly tools (e. g., Apache Airflow) support code-based workflow definitions, ideal for fine-grained control of data pipelines or microservice orchestration.

2. **Execution Engine:** This module parses workflow definitions and schedules task execution. Its key challenge lies in managing task dependencies and fault tolerance. Open-source tools like Temporal use

persistent task state history to ensure precise recovery of workflow progress after system crashes or network interruptions, avoiding duplicate executions or data loss.

3. **Connectors:** These bridge cross-system integration. Modern workflow platforms often pre-integrate hundreds of third-party service APIs (e. g., AWS, Slack, MySQL), enabling code-free connectivity. For legacy systems, universal connectors (e. g., Postman-developed modules) support REST or GraphQL interfaces for flexible extension.

4. **Monitoring and Logging System:** Critical for ensuring stability. Enterprise platforms (e. g., UiPath) track real-time execution states and integrate tools like Prometheus and Grafana to generate visual dashboards, helping operators identify bottlenecks or timeout issues. In complex scenarios, systems can trigger automatic alerts or invoke predefined scripts to resolve exceptions.

5. **AI-Enhanced Modules:** Emerging as a breakthrough. For example, an e-commerce platform uses machine learning to analyze historical order data and dynamically adjust inventory replenishment priorities. Other tools integrate large language models, allowing users to describe requirements in natural language (e. g., "Automatically summarize sales reports every Friday and email them"), with the system generating executable workflows—drastically lowering the entry barrier.

Technical Challenges and Solutions

Despite maturity, practical implementation faces challenges:

System Heterogeneity: Legacy and modern systems coexist with varying protocols. The industry addresses this via API standardization (e. g., OpenAPI Specification) and universal adapters. For example, a bank integrated a 20-year-old core system by converting legacy protocols to RESTful APIs via custom connectors.

Long-Running Process Management: Requires robust stability. Manufacturing workflows may span weeks, demanding resilience to disruptions. Tools like Temporal use event-driven architectures with checkpoints to resume from the latest state after failures. A car manufacturer reduced unplanned downtime by 90 % using this approach.

Security and Compliance: Critical in sensitive sectors. A healthcare platform combines OAuth 2.0

authentication with field-level encryption to protect patient data in automated workflows. Full audit logs ensure HIPAA compliance.

Future Trends

With AI and cloud-native advancements, workflow orchestration is evolving toward intelligence and ubiquity:

Intelligence: Large language models redefine human-machine interaction. Tests show GPT-4 can generate executable Airflow code in 10 seconds from prompts like "Monitor server CPU usage; scale resources and alert staff if exceeding 80 %," achieving 92 % accuracy. This lowers design barriers and enables "conversational automation."

Edge Computing: Extends workflows to endpoints. Industrial devices with WebAssembly runtimes execute local quality checks: edge AI analyzes camera images to trigger defect sorting in under 50ms latency, enhancing real-time performance and privacy.

Cross-Ecosystem Interoperability: Leading vendors build open platforms blending blockchain smart contracts with traditional systems. A logistics firm uses Chainlink oracles to optimize routes based on real-time fuel prices, saving \$3M annually.

Conclusion

Automated workflow orchestration has evolved from an efficiency tool to a cornerstone of digital transformation. Its value extends beyond replacing repetitive tasks to unlocking systemic synergies through intelligent scheduling. With AI, future systems will exhibit greater adaptability, serving as intelligent bridges between physical and digital ecosystems.

References

1. Apache Airflow in Action. [Electronic resource]. – Access mode : <https://airflow.apache.org>. – Access date : 20.04.2025.
2. Intelligent Automation: From Tool to Strategy [Electronic resource]. – Access mode : <https://www.gartner.com>. – Access date : 21.04.2025.
3. Zapier Enterprise Automation Whitepaper [Electronic resource]. – Access mode : <https://zapier.com/customer-stories>. – Access date : 22.04.2025.
4. Kubeflow [Electronic resource]. – Access mode : <https://www.kubeflow.org/docs/components/pipelines/>. – Access date : 3.05.2025.

UDC 004.415

A REAL-TIME DATA PROCESSING SYSTEM BASED ON APACHE FLINK

**Guo Rujie, master's degree student,
Dunina E., PhD in Physics and Mathematics, associate professor,
Dzerkachenka P., senior lecturer
Vitebsk State Technological University, Vitebsk, Republic of Belarus**

Abstract. Real-time data processing has become a cornerstone of modern data-driven industries, enabling organizations to extract actionable insights from high-velocity data streams. Apache Flink, an open-source stream processing framework, addresses the challenges of real-time analytics through its robust architecture and low-latency capabilities. This paper explores the core features of Flink, its applications across diverse domains, and the technical challenges in its deployment.

Keywords: stream processing, dynamic resource allocation, Apache Flink.

The exponential growth of data in industries such as finance, manufacturing, and healthcare has necessitated advanced tools for real-time analytics. Traditional batch processing systems fall short in addressing the latency and scalability demands of streaming workloads. Apache Flink, a state-of-the-art stream processing framework, bridges this gap by providing low-latency, high-throughput, and fault-tolerant capabilities. This paper examines Flink's architecture, its applications, and the