

MODAPTO [101091996]: Modular Manufacturing and Distributed Control via Interoperable Digital Twins



11.3.1 Designing service workflows and dependencies

2025-08-11

Designing effective service workflows and dependencies is a critical step in creating orchestrations that achieve manufacturing objectives while maintaining flexibility, performance, and reliability. This process involves identifying the services to be orchestrated, defining their execution sequence, and establishing the data flows between them.

Workflow Design Process

The process of designing service workflows typically follows these steps:

1. **Requirements Analysis:**
 - Identify the business process or manufacturing scenario to be orchestrated
 - Define the inputs required and outputs expected from the overall workflow
 - Determine any constraints or requirements regarding timing, reliability, or performance
2. **Service Identification:**
 - Identify the services needed to fulfill the requirements
 - Verify service availability in the Service Catalogue
 - Analyze service interfaces to understand input/output specifications
3. **Workflow Structure Definition:**
 - Define the sequence of service invocations
 - Identify parallel execution opportunities
 - Establish conditional branching points based on service outputs or system state
 - Determine error handling and recovery paths

Dependency Management

Managing dependencies between services is a critical aspect of workflow design:

1. **Data Dependencies:**
 - Identify how outputs from one service become inputs to another
 - Define data transformations needed between services
 - Establish data validation requirements at service boundaries
2. **Execution Dependencies:**
 - Determine which services must complete before others can start
 - Identify opportunities for parallel execution where dependencies allow
 - Establish synchronization points for converging parallel paths
3. **Resource Dependencies:**
 - Identify shared resources accessed by multiple services
 - Establish resource allocation and release patterns
 - Prevent deadlocks and resource contention

Workflow Patterns

Several common workflow patterns can be applied when designing orchestrations:



1. **Sequential Flow:**
 - Services execute one after another in a defined sequence
 - Each service completes before the next begins
 - Simple to implement but may not optimize performance
2. **Parallel Flow:**
 - Multiple services execute simultaneously
 - Results are collected and combined at a synchronization point
 - Improves performance but requires careful dependency management
3. **Conditional Branching:**
 - Execution path determined by conditions evaluated at runtime
 - Different services invoked based on data values or system state
 - Enables flexible workflows that adapt to changing conditions
4. **Iterative Flow:**
 - Services or sequences repeated until a condition is met
 - Used for processes requiring repetition or refinement
 - Requires careful termination condition definition

Implementation in MODAPTO Orchestrator

In the MODAPTO Orchestrator, workflow design is implemented through:

1. **Microservice Definition:**
 - Creating a Microservice container for the orchestration
 - Defining the overall purpose and interface of the orchestration
2. **Operations Configuration:**
 - Configuring individual Operations within the Microservice
 - Selecting appropriate connectors for each Operation
 - Defining input/output specifications for each Operation
3. **Orchestration Logic:**
 - Implementing workflow logic using JavaScript in the JavaScript Engine Connector
 - Using the `callMicroservice` function to invoke other services
 - Implementing conditional logic, data transformations, and error handling

The Orchestrator's JavaScript Engine Connector plays a central role in implementing complex workflows, as it allows arbitrary logic to be defined while providing the `callMicroservice` function that enables the invocation of other services from within the orchestration.

References

1. MODAPTO Consortium. (2023). Orchestrator User Manual. MODAPTO Project Documentation.
2. van der Aalst, W. M. P., & van Hee, K. M. (2004). Workflow Management: Models, Methods, and Systems. MIT Press.
3. Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns. Addison-Wesley.
4. OMG. (2013). Business Process Model and Notation (BPMN) Version 2.0. <https://www.omg.org/spec/BPMN/2.0/>

