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

12.1.2 Defining service metadata and configuration parameters

2025-08-11

Service metadata and configuration parameters form the essential descriptive and operational elements that enable effective service discovery, evaluation, and integration within the MODAPTO ecosystem. These elements go beyond basic technical specifications to provide comprehensive context about service behavior, requirements, and optimal usage patterns.

Core Metadata Elements establish the fundamental identity and context of a service. The service name should follow a consistent naming convention that reflects its function and domain, such as “PredictiveMaintenanceAnalyzer” or “QualityInspectionService”. The unique identifier must be a UUID (Universally Unique Identifier) that remains constant throughout the service lifecycle, even as versions change. Version information follows semantic versioning (MAJOR.MINOR.PATCH) where MAJOR changes indicate breaking changes, MINOR changes add functionality in a backward-compatible manner, and PATCH changes include backward-compatible bug fixes.

Descriptive Metadata provides human-readable information that helps users understand service purpose and value. The short description (limited to 280 characters) appears in service Catalogue listings and should concisely convey the service’s primary function. The long description supports rich text formatting (HTML or Markdown) and should comprehensively explain service capabilities, use cases, benefits, and any important considerations. This description often includes practical examples, typical integration scenarios, and expected outcomes from service usage.

Organizational Metadata establishes accountability and support structures. The affiliation identifies the organization or team responsible for the service, while contact information provides multiple channels for support (email, phone, ticketing system). Service level agreements (SLAs) may be referenced, defining expected support response times and escalation procedures. Licensing information clarifies usage rights, whether open source (MIT, Apache, GPL) or commercial, including any restrictions or requirements for service consumption.

Classification and Discovery Metadata enables efficient service location within large Catalogues. Keywords should include technology terms (e.g., “machine learning”, “OPC UA”), manufacturing processes (e.g., “assembly”, “quality control”), and industry standards (e.g., “ISO 9001”, “IEC 62264”). Categories follow the MODAPTO taxonomy, potentially including multiple classifications such as “Analytics/Predictive” and “Maintenance/Condition Monitoring”. Tags can indicate special characteristics like “real-time”, “certified”, or “GPU-required”.

Configuration Parameters define how services can be customized for specific deployments or use cases. Static configuration parameters are set during service registration and typically include endpoint URLs for External services, port numbers for Internal services, or model identifiers for FMU services. Dynamic configuration parameters can be adjusted at runtime and might include threshold values, sampling rates, or algorithm selection. Each parameter requires detailed specification including name, data type, default value, valid ranges or options, and description of its effect on service behavior.

Quality of Service (QoS) Parameters provide quantitative metrics for service performance expectations. Response time specifications indicate expected latencies for different operation types (e.g., synchronous calls < 100ms, batch processing < 5 minutes). Throughput metrics define processing capacity (e.g., 1000 requests/minute, 10GB/hour data processing). Availability targets specify uptime expectations (e.g., 99.9% availability, planned maintenance windows). Resource consumption estimates help in capacity planning, including CPU utilization, memory requirements, and network bandwidth needs.

Technical Constraints and Requirements document specific conditions necessary for service operation. Platform requirements might specify operating system versions, container runtime specifications, or cloud platform compatibility. Integration requirements could include specific protocol support (MQTT, AMQP), security mechanisms (OAuth2, API keys), or data format constraints (JSON Schema compliance). Dependency specifications list required external services, libraries, or infrastructure components with version constraints.

References

1. IEEE Standards Association. (2022). “IEEE 2757-2022 – IEEE Guide for Implementation of a Service-Oriented Architecture.” IEEE.
2. Industrial Internet Consortium. (2023). “Industrial Internet Reference Architecture v1.10.” IIC.
3. OPC Foundation. (2022). “OPC UA Part 5: Information Model.” OPC Foundation Specification.
4. W3C. (2023). “Data Catalogue Vocabulary (DCAT) – Version 3.” World Wide Web Consortium.
5. ISO/IEC 25010:2023. “Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model.” International Organization for Standardization.
6. NIST. (2023). “NIST Cybersecurity Framework 2.0.” National Institute of Standards and Technology.