A Semantic Framework to Compose RESTful Services

Luís Antonio de Almeida Rodriguez[1,*,†], Francisco de Oliveira[1] and
José Maria Parente de Oliveira[1]

[1]Aeronautics Institute of Technology, São José dos Campos-SP, Brazil

Abstract

The widespread adoption of RESTful services has led to a fragmented representational ecosystem, where the lack of a formal semantic description model hinders the reliable and automated composition of services. Reliance on non-standardized and non-machine-processable documentation introduces integration inconsistencies in distributed functionalities, requiring manual intervention to resolve ambiguities. This paper proposes an ontology-based semantic orchestration model for the composition of RESTful services, addressing two critical gaps: (i) the absence of a semantic model for describing RESTful service features; and (ii) the lack of semantic matching mechanisms for semantic orchestrations. This model proposes an ontological approach for the semantic orchestration of atomic and composite RESTful services' processes, enabling automated discovery and aggregation. This paper also presents an experimental evaluation within the context of the ICAO's SWIM program, demonstrating the effectiveness of composing RESTful shared services. Moreover, the approach reduces manual integration efforts compared to existing methods. The results suggest that the framework provides a replicable paradigm for large-scale service composition, representing an advance toward sustainable interoperability in heterogeneous distributed systems.

Keywords

RESTful services, Service Composition, OWL-S, Ontologies

1. Introduction

The growing adoption of RESTful services as a dominant architecture for web-based system integration has brought significant advancements in scalability, simplicity, and interoperability over the last decade. This type of service on the Web [1] is typically lightweight, stateless, and aligned with HTTP standards, making it suitable for cloud-native applications, public data platforms, and domain-specific APIs. However, despite their technical advantages, the RESTful paradigm suffers from a profound limitation when it comes to composing multiple services into a coherent workflow [2, 3]. Unlike earlier service-oriented approaches such as SOAP with WSDL and UDDI, RESTful services often lack formal mechanisms for description, discovery, and orchestration, which hampers their reuse and automation across heterogeneous environments.

One of the critical challenges in composing RESTful services lies in their documentation. In most cases, RESTful APIs are described informally through human-readable resources such as HTML pages, Swagger/OpenAPI files, or vendor-specific guides [4]. Although these formats may help developers during manual implementation, they do not allow machine-based reasoning or semantic search [5]. As a result, service integration becomes an error-prone, non-scalable, and context-dependent process that depends heavily on developer interpretation and custom integration logic. This becomes particularly problematic when attempting to aggregate services provided by independent organizations, each adhering to different naming conventions, data structures, and interface assumptions.

Proceedings of the 18th Seminar on Ontology Research in Brazil (ONTOBRAS 2025) and 9th Doctoral and Masters Consortium on Ontologies (WTDO 2025), São José dos Campos (SP), Brazil, September 29 – October 02, 2025.

*Corresponding author.
†These authors contributed equally.
rodriguezlaar@gmail.com (L. A. d. A. Rodriguez); francisco.oliveira@ifsp.edu.br (F. d. Oliveira); parente@ita.br (J. M. P. d. Oliveira)
linkedin.com/in/luÃŋs-antonio-rodriguez-msc-641b3021 (L. A. d. A. Rodriguez); www.francisco.ifsp.edu.br (F. d. Oliveira); https://www.linkedin.com/in/jose-m-parente-de-oliveira-53188618/ (J. M. P. d. Oliveira)
0009-0002-8669-2425 (L. A. d. A. Rodriguez); 0000-0001-5308-7355 (F. d. Oliveira); 0000-0002-7803-1718 (J. M. P. d. Oliveira)

The absence of a unified standard to semantically describe the behavior and capabilities of RESTful services exacerbates the situation [3, 5]. Without a shared vocabulary or formal representation of the functionalities of the service, there is no reliable way to automate the discovery, matching, or composition of services based on their meaning or role in a process. This severely limits the potential for dynamic service orchestration in domains that demand agility, such as meteorology, logistics, healthcare, and air traffic management. For these sectors, the ability to semantically describe and connect service processes is fundamental to achieving interoperability, scalability, and intelligent automation.

This paper addresses these issues by proposing a semantic framework for describing and orchestrating RDF Knowledge Graphs representing RESTful services using ontologies, specifically by extending an OWL-S-based model presented in [5], which extends OWL-S to RESTful architectures. The identified core problem is the absence of an agreed-upon semantic representation for service processes that enables semantic orchestration and reasoning, which will enable automatic composition based on QoS criteria by any API. The proposed solution introduces an ontologically grounded approach for describing RESTful service capabilities, including the associations between atomic and composite processes, with an agreed-upon semantic language that integrates and allows automated extraction of information from the ontologies. A case study based on a meteorological service platform used in aviation (e.g., REDEMET-API) presents the effectiveness and applicability of this model within the broader context of global civil aviation and SWIM (System Wide Information Management).

2. Semantic Standardization in SWIM and the Role of OWL-S

The System Wide Information Management (SWIM) program, coordinated by the International Civil Aviation Organization (ICAO), is a global initiative that standardizes information exchange and fosters interoperability across civil aviation systems [?]. SWIM incorporates data exchange models, governance methodologies, and infrastructure patterns to ensure efficient, secure, and machine-readable distribution of air traffic information. Its primary objective is to enhance the availability, accessibility, and consistency of air navigation data across national and organizational boundaries. By complementing traditional human-to-human communication with structured and automated information flows, SWIM establishes an environment where stakeholders benefit from high-quality semantic data exchange, which is critical for safety, scalability, and international collaboration [?].

The alignment of SWIM with W3C Semantic Web standards is achieved through the OWL-S ontology, which provides the formal structure to represent service capabilities, behaviors, and grounding protocols in a machine-interpretable manner. ICAO and EUROCONTROL documentation emphasize OWL-S as a foundational tool for semantic interoperability, ensuring precise, reusable, and automated service descriptions [?]. However, since OWL-S was originally grounded in SOAP-based infrastructures, extending its principles to RESTful architectures is necessary. This research contributes by adapting OWL-S to REST semantics—incorporating HTTP verbs, URIs, content negotiation, and resource representation types—thus bridging the gap between traditional SOAP/WSDL-based services and modern Web services. This ontology-driven extension enables automated discovery, quality-aware composition, and accurate invocation of RESTful services, fulfilling SWIM's requirements for semantic-native interoperability in heterogeneous and distributed aviation environments.

3. A Semantic Orchestration Model

The goals of this section are the following.

1. To extend the work presented in [5] and create a semantic model to support the orchestration of service process relationships to automate the composition of services by APIs. Building upon the OWL-S extension developed in this paper, where RESTful services' processes have an ontologically defined taxonomy using 'Atomic' and 'Composite' classifications, this work involved classifying those implemented SOA-unit service processes and building an RDF Knowledge Graph by associating them, creating a RDF Knowledge Graph.

2. Refactor the software proposed in [5] and develop a new functionality to perform automated dynamic service composition, driven by the common web user's own choices. Specifically, this functionality leverages the semantically enriched processes already orchestrated in the model and links them with the REDEMET-API RESTful services, generalizing the composition coded with Python to accommodate service descriptions with different abstraction levels.

The orchestration logic, grounded in OWL-S' ontological compatibility, enables any API to integrate diverse services into cohesive workflows without requiring manual intervention. This enhancement addresses the broader vision of democratizing service composition, allowing non-expert users to trigger complex sets of services by specifying 'Quality of Service Level' ranks for each one. In doing so, this paper's proposal reinforces the practical value of semantic description environments in real-world web architectures and extends the OWL-S adaptation presented in [5] toward multi-provider interoperability for intelligent compositions.

The central point of this section is the *Process* ontology, exactly as it is presented in [5]. This ontology, within the extension of OWL-S proposed in this paper, serves as a foundational component to describe the behavioral dynamics of services on the Web. Its internal structure is systematically organized according to several defining criteria that establish the functional and operational semantics of service processes. These criteria facilitate the semantic search, and the specific Atomic-Composite association facilitates service composition. Central to this ontology are the Inputs, Outputs, Preconditions, and Effects (IOPE), which provide a formal specification of a service's capability, allowing for precise semantic matching during service discovery. Precise search and discovery of services are essential prerequisites for service composition; finding the best services enables the creation of an optimal composition.

Martin [6] comments that Atomic processes represent indivisible actions with direct invocations, while Composite processes, on the other hand, define complex associations through atomic or other composite processes. The *Process* ontology's ontological representation enforces a strict taxonomic distinction whereby every individual of the **Process** class must also be instantiated as a member of **AtomicProcess**. Figure 1 shows that some of them are instances of the **CompositeProcess** class, in full compliance with the foundational specifications of OWL-S. This mandatory classification is formally realized through two canonical inverse properties, as depicted in the modeling presented in [5]. Figure 1 illustrates that:

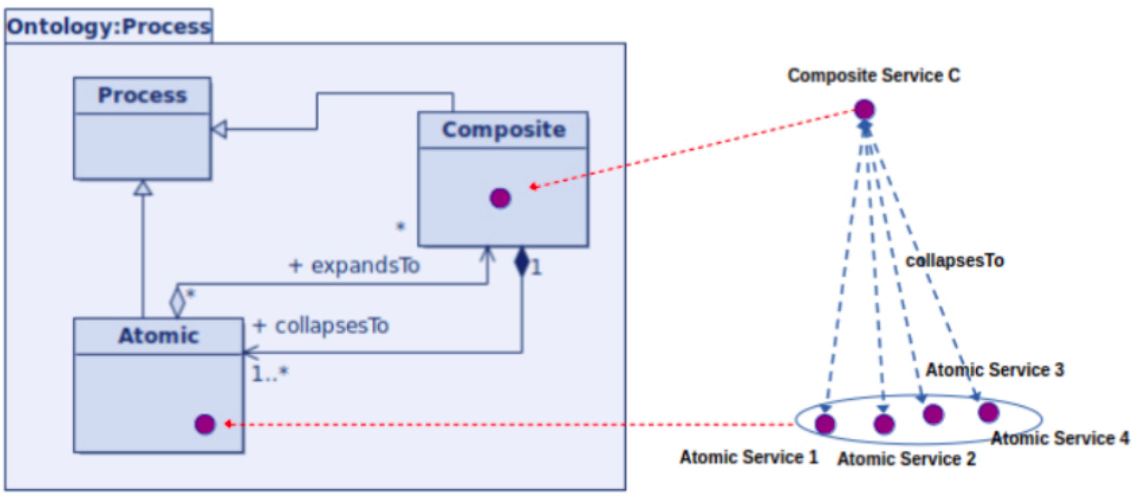

Figure 1: Orchestration's Model of Process entities.

- collapsesTo - A decomposition linkage: **CompositeProcess** → **AtomicProcess**

- expandsTo - An 'Aggregation': **AtomicProcess → CompositeProcess**

This dual-property framework ensures both structural integrity and semantic consistency in process decomposition, thereby enabling rigorous reasoning over service composition orchestrations such as this one. Using these properties, it is possible to select all instances of services already implemented and insert their 'process entities' into the existing OWL classes: **Atomic** and **Composite**, in the *Process* ontology. After that, associations among composite and atomic entities are necessary using object properties already defined in this ontology (e.g., collapsesTo and expandsTo). Why classify them all as atomic and some as composite? By doing this, it is possible to establish two types of composition:

- A composite service composed only of atomic services, and
- A composite service composed of atomic and other composite services.

The formalization of service processes classification and maximization was given considering the OWL-S Compliant Knowledge Graph. According to [5], all instances of service processes within the developed knowledge base have been systematically classified as members of the **AtomicProcess** class, and some were classified in the **CompositeProcess** class. This strict taxonomic enforcement ensures adherence to the foundational semantics of OWL-S and a new characteristic: the ability to create composite services composed of *other composite services.*

To establish machine-interpretable semantic relationships between processes, each instance of a process from the semantic registry built in [5] has been explicitly associated with the canonical object properties collapsesTo and expandsTo, thereby formalizing hierarchical dependencies within service compositions:

- AtomicProcess individuals represent an indivisible service set of operations, with no further decomposition permitted under the axiomatic constraints of the ontology (e.g., even for 'composite services' classified as such).
- CompositeProcess individuals are structured aggregations of subprocesses, recursively defined via the collapsesTo property, which links a composite service to its constituent atomic or nested composite processes.

The resultant RDF knowledge graph encodes these relationships as semantically described triples, conforming to OWL-S specifications while enabling advanced querying capabilities. Through SPARQL queries, the graph can be interrogated to:

- Retrieve atomic services meeting specific functional criteria (e.g., inputs/outputs, preconditions);
- Decompose composite services into their procedural workflows, exposing nested execution logic;
- Validate process compositions by reasoning over property constraints.

The structured representation presented in Figure 1 collaborates with more criteria for the discovery of semantic services and the analysis of automated composition, as the formal semantics of the graph permit inference about the compatibility of processes and behavioral equivalence. It is possible to mention some consequences of applying our model proposed in Figure 1 to the resulting RDF Knowledge Graph:

- Establishment of a hierarchical process topology
- Creation of an explicit definition of the service composition paths through relationships among nodes, represented by Triples
- A layer of descriptions made by semantic machine-readable composition patterns

By formalizing process hierarchies, constraints, and operational logic, the Process Ontology ensures that services can be dynamically discovered, composed, and invoked in a semantically consistent manner.

4. Case Study

RESTful aeronautical service development communities encounter systemic interoperability barriers due to representational fragmentation: a divergence in service description models [5]. This causes a huge development effort to build APIs able to use RESTful shared services on the Web.

4.1. Motivation

This study addresses the urgent need to realize SWIM's vision of seamless and standardized air traffic management (ATM) data exchange between the ICAO member states, aiming to improve the ability to use shared RESTful services [7] to create more complex services. SWIM's proposed semantic enrichment of service descriptions transforms interoperability from a legacy concept into a natively standardized communication paradigm, enabling a globally harmonized aviation data interoperability. Building on existing frameworks, this paper aims to bridge these gaps, ensuring deterministic composition made by software through semantically enriched service descriptions.

4.2. Orchestrating and Composing Complex RESTful Services

The article by Rodriguez and Parente de Oliveira [5] presents a rigorous semantic model designed to enable the composition of RESTful services through ontologically grounded orchestration. Drawing on the foundational architecture of OWL-S, the model presents a dual-level abstraction also presented in the original OWL-S: atomic and composite processes. Each SOA-unit service registered in this semantic registry has its process entity semantically annotated and encapsulated within a class called Atomic process, which is connected, in the RDF Knowledge Graph, with entities that describe its capabilities, such as inputs, outputs, preconditions, and effects (IOPE).

Composite processes are structured as orchestrations of atomic processes made to define the logical and temporal relations between service calls [8, 5]. This semantic representation enables reasoning engines to identify, match, and bind services dynamically, thereby automating the composition of services into coherent workflows. To instantiate the model presented in [5], the authors applied it in a case study involving the semantic representation and orchestration of 23 SOA unit services from the original model. The propagation of the orchestration of SOA-unit services will reach 17 RESTful services provided by REDEMET-API [4], and three other RESTful services as part of a provider specifically created in the registry for this paper, the **SWIM Provider**. The SOA-unit services' process entities were first formalized as an atomic process, where the semantic annotations create an atomic foundation for serving several composite processes.

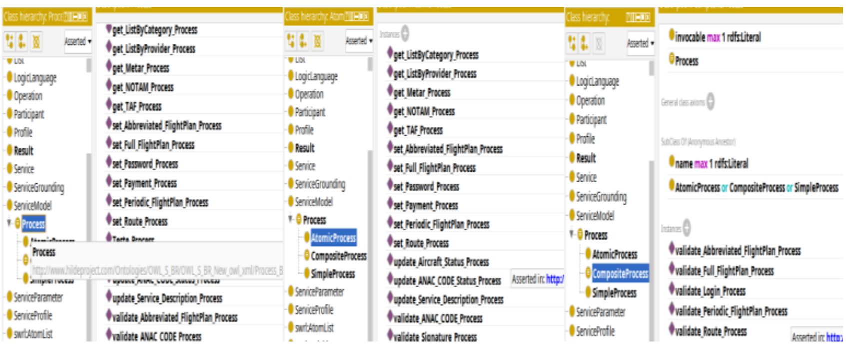

Figure 2: Process entities orchestrated in Protegé.

Figure 1 presents the proposed orchestration. This hierarchical structure enables the expression of increasingly complex service behaviors while preserving semantic clarity. This layered composition model demonstrates the flexibility of OWL-S in describing realistic service orchestration scenarios in the RESTful paradigm. There are no cyclic representations in the RDF Knowledge Graph. The functionality is granted through an accurate set of associations between all composite and atomic processes.

The semantic orchestration presented in this paper was implemented in the *Process* ontology, such as in [5] using Protégé 5.5. The procedure begins with the import of the OWL-S ontology into a new project. The entity of the process of each generic service was created in the **ServiceModel** class as an instance of the `AtomicProcess` class. The HTTP semantics, including the type of the method and the endpoint URI, were encoded through custom data properties or extended using existing ontologies such as *Aerodrome*, *Country*, and *RESTfulExtension*.

Subsequently, composite processes were created using the `CompositeProcess` class, and their execution flow was defined via semantic relationships between the processes using ***Object Properties*** already existing in the previous model [5]. This ultimately results in a machine-interpretable semantic representation that captures the logic of SOA-unit composite services and propagates to the constraints and sequencing of the REDEMET-API RESTful services network. This case study implementation effort produced a machine-interpretable semantic infrastructure that not only encapsulates the logic of composite SOA-unit services but also propagates the predefined features about process sequencing across the REDEMET-API meteorological RESTful services network.

The case study practical outcomes are multifold: for the developers community, it offers a reusable and extensible description model for service composition's orchestration; for global civil aviation, the model aligns with SWIM's vision of semantic interoperability and dynamic information sharing; for end users and companies, it enables intelligent service discovery and automated interaction, thus reducing integration costs, enhancing scalability, and paving the way for a new generation of semantically aware RESTful ecosystems.

The orchestration model proposed and implemented in this study provides concrete generic-domain benefits by enabling a semantic composition framework that propagates from generic SOA-units to specific RESTful services. By providing a formal semantic description, developers can automate essential events of the composition pipeline, replacing ad hoc scripting and manual configuration with logically inferred orchestration. By offering semantic transparency and structural formalization, the orchestration mechanism allows heterogeneous provider services to be composed into reliable workflows without human intervention, enhancing situational awareness, operational efficiency, and safety. Beyond aviation, the approach serves as a generic technological contribution to the field of service composition by demonstrating that OWL-S-based semantic models can be successfully adapted to RESTful paradigms.

5. Conclusions

5.1. Contributions

A key contribution is the definition of a novel orchestration format, grounded in OWL-S and enriched with precise object properties (e.g., expandsTo, collapsesTo), which allows developers to express composition relationships among services with reduced syntactic ambiguity. It reduces the cognitive and technical effort required to construct complex service integrations, exactly as evaluated in [5], replacing heuristic-based composition methods with logic-based reasoning. Moreover, by implementing and validating the model over a real-world API—REDEMET, the paper contributes a reusable, standards-aligned methodology for modeling and composing RESTful services in domains requiring high semantic precision, such as civil aviation under the SWIM framework.

Declaration on Generative AI

During the preparation of this work, the author(s) used Deep Seek as a tutor for English Grammar Correction.

References

[1] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, Ph.d. dissertation, University of California, Irvine, Irvine, CA, USA, 2000. URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.

[2] Y. Gamha, A framework for rest services discovery and composition, Service Oriented Computing and Applications 17 (2023) 259–275. doi:10.1007/s11761-023-00376-6.

[3] M. Bennara, M. Mrissa, Y. Amghar, An approach for composing restful linked services on the web, in: Proceedings of the 23rd International Conference on World Wide Web, Seoul, South Korea, 2014. doi:10.1145/2567948.2579222, hal-01212722v2.

[4] REDEMET-API, Brazilian Aeronautical Provider, https://ajuda.decea.mil.br/base-de-conhecimento/api-redemet-o-que-e/, 2025. Accessed: Apr. 16, 2025.

[5] L. A. A. Rodriguez, J. M. P. de Oliveira, A semantic model to describe restful services, IEEE Access 13 (2025) 72402–72426. doi:10.1109/ACCESS.2025.3562503.

[6] D. Martin, Bringing semantics to web services: The owl-s approach, IEEE Intelligent Systems 22 (2007) 72–81. doi:10.1109/MIS.2007.100.

[7] EUROCONTROL, Swim technical infrastructure and governance: Semantic interoperability guidelines, 2019.

[8] OWL-S Coalition, Owl-s: Semantic markup for web services, version 1.2, https://www.daml.org/services/owl-s/1.2/, 2006. Released by the OWL-S Coalition, including members from SRI International, Carnegie Mellon University, BBN Technologies, and University of Maryland.