INTERNATIONAL STANDARD

ISO/IEC 18384-3

First edition
2016-07-01

Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) —

Part 3: Service Oriented Architecture ontology

Technologie de l'information — Architecture de référence pour l'architecture orientée service (SOA RA) —

Partie 3: Ontologie de l'architecture orientée service

Reference number
ISO/IEC 18384-3:2016(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword – Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 38, *Cloud Computing and Distributed Platforms*.

ISO/IEC 18384 consists of the following parts, under the general title *Reference Architecture for Service Oriented Architecture (SOA RA)*:

— *Part 1: Terminology and concepts for SOA*

— *Part 2: Reference Architecture for SOA Solutions*

— *Part 3: Service Oriented Architecture Ontology*

Introduction

Service oriented architecture (SOA) is an architectural style in which business and IT systems are designed in terms of services available at an interface and the outcomes of these services. A service is a logical representation of a set of activities that has specified outcomes, is self-contained, it may be composed of other services but consumers of the service need not be aware of any internal structure.

SOA takes "service" as its basic element to constitute and integrate information systems so that they are suitable for a variety of solution requirements. SOA enables interactions between businesses without needing to specify aspects of any particular business domain. Using the SOA architectural style can improve the efficiency of developing information systems and integrating and reusing IT resources. In addition, using the SOA architectural style can help enable rapid response of information systems to ever-changing business needs.

This International Standard is intended to be a single set of SOA technical principles, specific norms, and standards for the world-wide market to help remove confusion about SOA and improve the standardization and quality of solutions.

This International Standard defines the terminology, technical principles, reference architecture and the ontology for SOA. ISO/IEC 18384 can be used to introduce SOA concepts, as a guide to the development and management of SOA solutions, as well as be referenced by business and industry standards.

This International Standard contains three parts:

1) ISO/IEC 18384-1 which defines the terminology, basic technical principles and concepts for SOA.

2) ISO/IEC 18384-2 which defines the detailed SOA reference architecture layers, including a metamodel, capabilities, architectural building blocks, as well as types of services in SOA solutions.

3) ISO/IEC 18384-3 which defines the core concepts of SOA and their relationships in the Ontology.

The targeted audience of this International Standard includes, but is not limited to, standards organizations, architects, architecture methodologists, system and software designers, business people, SOA service providers, SOA solution and service developers, and SOA service consumers who are interested in adopting and developing SOA.

Users of this International Standard will find it useful to read ISO/IEC 18384-1 for an understanding of SOA basics. ISO/IEC 18384-1 should be read before reading or applying ISO/IEC 18384-2. For those new to the SOA reference architecture in ISO/IEC 18384-2:2016, Clause 4 provides a high level understanding of the reference architecture for SOA solutions. The remaining clauses provide comprehensive details of the architectural building blocks and tradeoffs needed for a SOA Solution. This part of ISO/IEC 18384 contains the SOA Ontology, which is a formalism of the core concepts and terminology of SOA, with mappings to both UML and OWL. The SOA Ontology can be used independent of or in conjunction with ISO/IEC 18384-1 and ISO/IEC 18384-2.

The purpose of this part of ISO/IEC 18384 is to contribute to developing and fostering common understanding of service-oriented architecture (SOA) in order to improve alignment between the business and information technology communities and facilitate SOA adoption.

The SOA Ontology defines the concepts, terminology, and semantics of SOA in both business and technical terms, in order to

— create a foundation for further work in domain-specific areas,

— enable communications between business and technical people,

— enhance the understanding of SOA concepts in the business and technical communities,

— provide a means to state problems and opportunities clearly and unambiguously to promote mutual understanding, and

— provide a starting point for model-driven development of SOA solutions.

Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) —

Part 3:
Service Oriented Architecture ontology

1 Scope

This part of ISO/IEC 18384 defines a formal ontology for service-oriented architecture (SOA), an architectural style that supports service orientation. The terms defined in this ontology are key terms from the vocabulary in ISO/IEC 18384-1.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 18384-1, *Information technology — Reference Architecture for Service Oriented Architecture (SOA RA) — Part 1 Terminology and concepts for SOA*

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 18384-1 and the following apply.

3.1.1
opaque
having no internal structure that is visible to an external observer

3.1.2
ontology
model that represents a domain and is used to reason about the objects in that domain and the relations between them

Note 1 to entry: This part of ISO/IEC 18384 is high level and not meant to be used for formal reasoning.

[SOURCE: ISO/IEC/TR 24800-1:2007, 2.1.9]

3.2 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

ABB	Architecture Building Block
BPMN	Business Process Model and Notation
EA	Enterprise Architecture
ESB	Enterprise Service Bus
IT	Information Technology

OWL	Web Ontology Language
RA	Reference Architecture
RDF	Resource Definition Framework
SLA	Service Level Agreement
SOA	Service Oriented Architecture
UML	Unified Modeling Language

4 Notations

The ontology is represented in the web ontology language (OWL) defined by the World Wide Web Consortium. OWL has three increasingly expressive sub-languages: OWL-Lite, OWL-DL, and OWL-Full (see Reference [10] for a definition of these three dialects of OWL). This ontology uses OWL-DL, the sub-language that provides the greatest expressiveness possible while retaining computational completeness and decidability.

The ontology contains classes and properties corresponding to the concepts of SOA. The formal OWL definitions are supplemented by natural language descriptions of the concepts, with graphic illustrations of the relations between them, and with examples of their use. For purposes of exposition, the ontology also includes UML (see Reference [8]) diagrams that graphically illustrate its classes and properties of the ontology. The natural language and OWL definitions contained in this part of ISO/IEC 18384 constitute the authoritative definition of the ontology; the diagrams are for explanatory purposes only. Some of the natural language terms used to describe the concepts are not formally represented in the ontology; those terms are meant in their natural language sense.

The availability of an OWL expression, a standard RDF format, allows easy loading into tools for architects and developers and allows validation.

This part of ISO/IEC 18384 uses examples to illustrate the ontology. One of these, the car-wash example, is used consistently throughout to illustrate the main concepts (see Annex A for the complete example). Other examples are used ad hoc in individual clauses to illustrate particular points.

5 Conventions

Bold font is used for OWL class, property, and instance names where they appear in clause text.

Italic strings are used for emphasis and to identify the first instance of a word requiring definition.

OWL definitions and syntax are shown in fixed-width font.

An unlabeled arrow in the illustrative UML diagrams means subclass.

The examples in this part of ISO/IEC 18384 are strictly informative and are for illustrative purposes.

6 Conformance

ISO/IEC 18384 contains three parts which have different conformance requirements:

1. terminology and concepts — conformance only to terms and adherence to the semantics in the definitions;

2. reference architecture for SOA solutions — conformance only to semantics of the metamodel and any Layers, ABBs, or capabilities that are used;

3. SOA Ontology — conformance for OWL or non-OWL applications.

Conformance to this part of ISO/IEC 18384 is defined as follows.

There are two kinds of applications that may conform to this ontology. One is the OWL-based ontologies (typically extensions of the SOA ontology); the other is a non-OWL application, such as a meta-model or a piece of software (see Clause 2 for the OWL version that is required).

A conforming OWL application (derived OWL-based ontology)

— shall conform to the OWL standard specified in Clause 2,

— shall include the whole of the ontology contained in Annex C,

— may add other OWL constructs, including class and property definitions, and

— may import other ontologies in addition to the SOA ontology.

This part of ISO/IEC 18384 does not use any OWL 2 (see Reference [15]) constructs; however, conforming applications may choose to use OWL or OWL 2.

A conforming non-OWL application

— shall include a defined and consistent transformation (at least semantic mapping) to a non-trivial subset of the ontology contained in Annex C,

— may add other constructs, including class and property definitions, and

— may import and/or use other ontologies in addition to the SOA ontology.

7 SOA Ontology Overview

7.1 At a Glance

A graphically compressed visualization of the entire ontology is shown in Figure 1.

The concepts illustrated in Figure 1 are described in the body.

This part of ISO/IEC 18384 starts by explaining the most basic foundational concept of elements and systems followed by explaining the elements of SOA human actor and task and then service concepts and descriptions and contracts for services and building on that to explain compositions of services. Finally, this part of ISO/IEC 18384 wraps up with Policies and Events which are relevant to all of the elements of SOA.

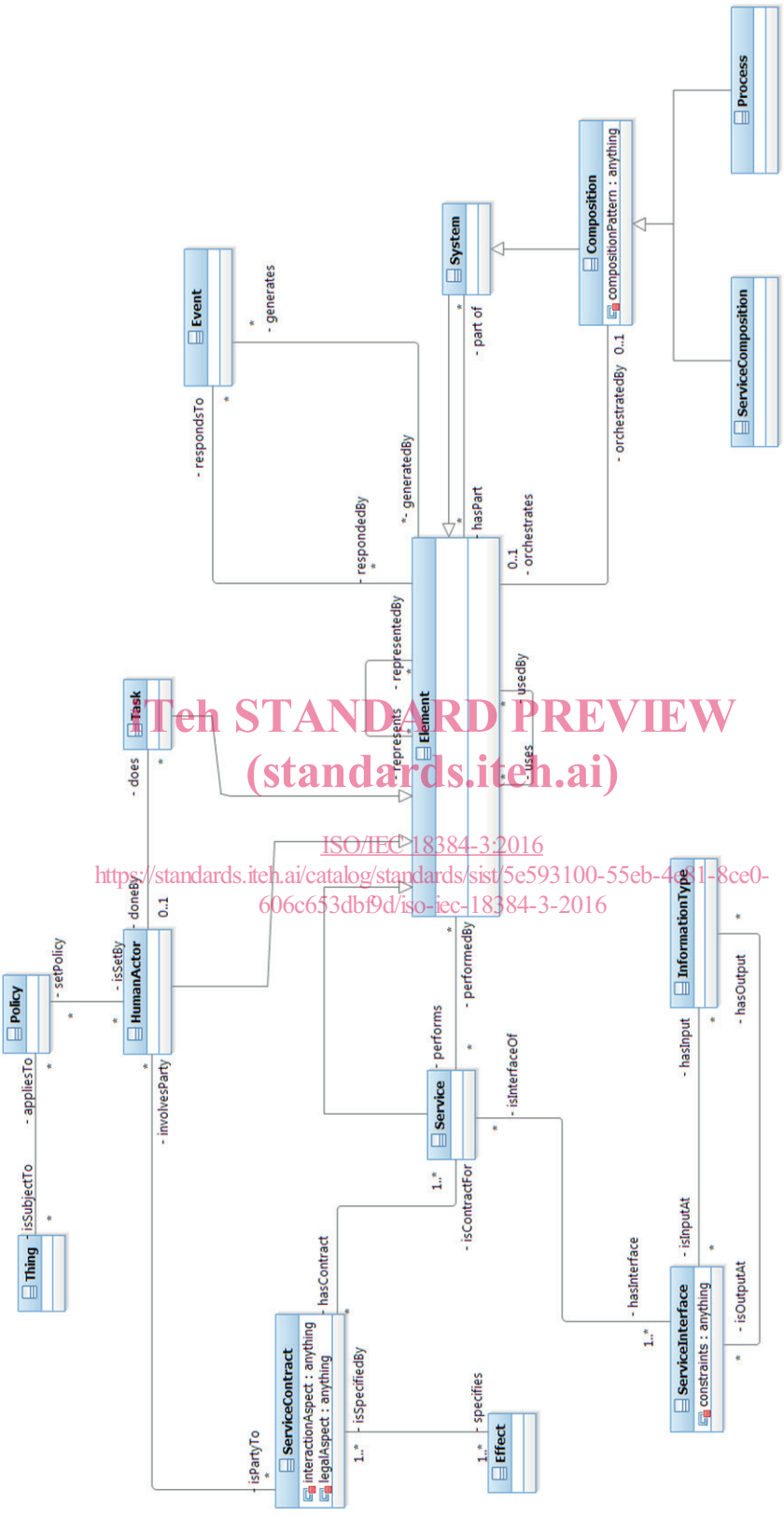

Figure 1 — SOA Ontology — Graphical Overview

7.2 Intended Use

This Clause describes caveats and assumptions for how this ontology should be interpreted.

— This ontology is intended for high level representation of concepts and is not intended for formal reasoning.

— This part of ISO/IEC 18384 is designed for use by business people, architects and systems and software designers to enable communications between business and technical people.

— This part of ISO/IEC 18384 focuses on a minimal set of SOA terms, modelling those terms in detail.

— This part of ISO/IEC 18384 explains relationships to other important concepts, but not at the same level of detail as the SOA terms. For example, policy is modelled, but not in great detail.

— This part of ISO/IEC 18384 restricts itself to OWL constructs, not using those introduced in OWL 2 (see Reference [15]), because the OWL constructs are sufficient for the scope of this part of ISO/IEC 18384. It is consistent with OWL 2 and does not preclude others from using it with OWL 2.

— This part of ISO/IEC 18384 elaborates on the SOA terms and relationships in ISO/IEC 18384-1 and ISO/IEC 18384-2. A separate metamodel in ISO/IEC 18384-2 provides the basis for the modeling in ISO/IEC 18384-2 and is used to describe and understand the reference architecture.

— This part of ISO/IEC 18384 defines the concepts, terminology, and semantics of SOA in both business and technical terms, in order to create a foundation for further work in domain-specific areas.

— This part of ISO/IEC 18384 provides a means to state problems and opportunities clearly and unambiguously to promote mutual understanding.

— This part of ISO/IEC 18384 may provide a starting point for model-driven development of SOA solutions.

7.3 Applications

The SOA ontology was developed in order to aid understanding and can simply be read.

It can also be used as a starting point for model-driven development, by applying it to particular usage domains and applications.

The ontology is applied to a particular usage domain by adding SOA OWL class instances of things in that domain. This is sometimes referred to as “populating the ontology.” In addition, an application can add definitions of new classes and properties, can import other ontologies, and can import the ontology OWL representation into other ontologies.

The ontology defines the relations between terms, but does not prescribe exactly how they should be applied. For explanations of what ontologies are and why they are needed, see References [11] and [14]. The examples provided in this part of ISO/IEC 18384 are describing one way in which the ontology could be applied in practical situations. Different applications of the ontology to the same situations would nevertheless be possible. The precise instantiation of the ontology in particular practical situations is a matter for users of the ontology, as long as the concepts and constraints defined by the ontology are correctly applied, the instantiation is valid.

8 System and Element

8.1 Overview

System and *element* are two of the concepts of this ontology. Both are concepts that are often used by practitioners, including the notion that systems have elements and that systems can be hierarchically combined (systems of systems). What differs from domain to domain is the specific nature of systems and elements, for instance, an electrical system has very different kinds of elements than an SOA system.

In the ontology, only elements and systems within the SOA domain are considered. Some SOA sub-domains use the term *component* rather than the term element. This is not contradictory, as any component of an SOA system is also an element of that (composite) system.

This Clause describes the following classes of the ontology:

Element

System

In addition, it defines the following properties:

uses and **usedBy**

represents and **representedBy**

8.2 The Element Class

```
<owl:Class rdf:about="#Element">
</owl:Class>
```

An *element* is an entity that is opaque and indivisible at a given level of abstraction. The element has a clearly defined boundary. The concept of element is captured by the **Element** OWL class, which is illustrated in Figure 2.

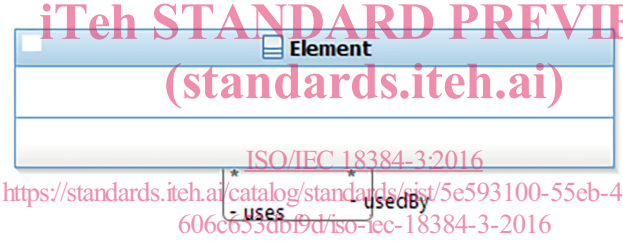

Figure 2 — The Element Class

In the context of the SOA ontology, only functional elements that belong to the SOA domain are considered in detail. There are other kinds of Elements than members of the four named subclasses (System, HumanActor, Task, and Service) described later in this ontology. Examples of such other kinds of Elements are things like software components or technology components (such as Enterprise Service Bus (ESB) implementations, etc.).

8.3 The uses and usedBy Properties

```
<owl:ObjectProperty rdf:about="#uses">
  <rdfs:domain rdf:resource="#Element"/>
  <rdfs:range rdf:resource="#Element"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#usedBy">
  <owl:inverseOf>
   <owl:ObjectProperty rdf:about="#uses"/>
  </owl:inverseOf>
</owl:ObjectProperty>
```

Elements may use other elements in various ways. In general, the notion of some element using another element is applied by practitioners for all of models, executables, and physical objects. What differs from domain to domain is the way in which such use is perceived.

An element uses another element if it interacts with it in some fashion. Interacts here is interpreted very broadly ranging through, for example, an element simply being a member of (used by) some system (see later for a formal definition of the **System** class), an element interacting with (using) another element (such as a service; see later for a formal definition of the **Service** class) in an *ad hoc* fashion, or even a strongly coupled dependency in a composition (see later for a formal definition of the **Composition** class). The **uses** property, and its inverse **usedBy**, capture the abstract notion of an element using another. These properties capture not just transient relations. Instantiations of the property can include "uses at this instant", "has used", and "may in future use".

For the purposes of this ontology, the multitude of different possible semantics of a *uses* relationship is not enumerated and formally defined .The semantic interpretations are left to a particular sub-domain, application or even design approach.

8.4 Element — Organizational Example

Using an organizational example, typical instances of **Element** are organizational units and people. Whether to perceive a given part of an organization as an organizational unit or as the set of people within that organizational unit is an important choice of abstraction level.

Inside the boundary of the organizational unit, as the organizational unit can in fact use the people that are members of it. Note that the same person can in fact be a member of (be used by) multiple organizational units.

Outside the boundary the internal structure of an organizational unit remains opaque to an external observer, as the enterprise wants to be able to change the people within the organizational unit without having to change the definition of the organizational unit itself.

This simple example expresses that some elements have an internal structure. In fact, from an internal perspective they are an organized collection of other simpler things (captured by the System class defined in 8.5).

8.5 The System Class

```
<owl:Class rdf:about="#System">
  <owl:disjointWith>
   <owl:Class rdf:about="#Task"/>
  </owl:disjointWith>
  <owl:disjointWith>
   <owl:Class rdf:about="#Service"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
   <owl:Class rdf:about="#Element"/>
  </rdfs:subClassOf>
</owl:Class>
```

A *system* is an organized collection of other things. Specifically, things in a system collection are instances of **Element**, each such instance being used by the system. The concept of *system* is captured by the **System** OWL class, which is illustrated in Figure 3.

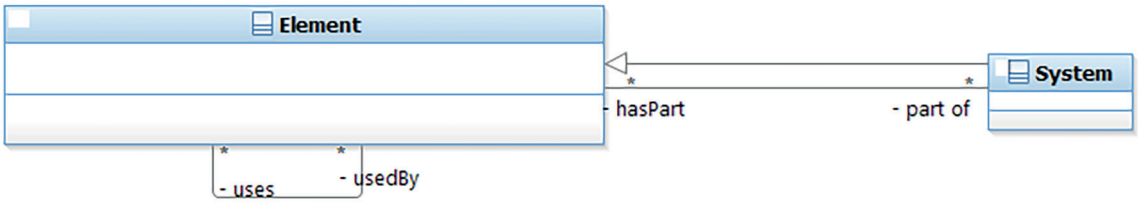

Figure 3 — The System Class