INTERNATIONAL STANDARD

ISO 5054-1

First edition
2023-02

Specification for an enterprise canonical model —

Part 1:
Architecture

Reference number
ISO 5054-1:2023(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

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 ISO documents 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 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 of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 154, *Processes, data elements and documents in commerce, industry and administration*.

A list of all parts in the ISO 5054 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document focuses on the need for interoperability at the business process level and the benefits for companies of being able to mix and match different vendor solutions to address their requirements. When companies want to purchase business applications from multiple vendors, they have the difficult task of getting the business applications to work together without the benefit of controlling the integration of those business applications. Additionally, customers are struggling with the larger task of integrating all of their systems into a coherent information technology infrastructure to support their business.

By defining integration standards as an enterprise canonical model, application integration can be optimized for all, instead of having different costly point-to-point solutions between each business applications, businesses and software vendors benefit.

The definition of the enterprise canonical model is a “single source of truth” for a semantic language all enterprise applications can speak, for commerce worldwide, for enterprise applications such as enterprise resource planning (ERP), purchasing, order management, finance, customer management, quality data, and more. The long-term scope is to enable all current and future enterprise application integration in a syntax-neutral manner to keep pace with rapidly changing technology landscape and resource skillsets. Other applications of canonical model include business intelligence and reporting, to normalize and aggregate business data across applications to provide a consistent vocabulary to generate business insights (see Annex B).

The enterprise canonical model reflects many thousands of person hours contributed by the Open Applications Group[1)] (OAGi) member companies.

The contributors represent the people who are building, testing, and implementing their business applications in thousands of companies worldwide.

It is anticipated that future parts of this standard will:

— Provide specification of a physical data model that underlies the model-based approach to data exchange specification. The data model is an adoption of the ISO 15000-5 standard conceptual model, as is implemented in open-source application named Score.

— Discuss the content, including the actual nouns and messages (business object documents - BOD), of the standard. The document contains a discussion of the notion of business context, including its definition and use to specify precisely profiles (i.e. subsets of) nouns. A list of integration scenarios is provided, which are the basis for determining detailed business processes that are essential for the definition of business context. Each scenario is provided with links to specific BODs appearing in the scenarios. Additionally, list of all nouns, which make up the standard, is provided. Finally, differing delivery approaches for (including canonical, standalone, and database) and alternative formats (JSON Schema,[1] XSD, etc.) are described.

— Describe the platform aspect of the standard whereby the user can create his or her own noun or message (BOD). The purpose of the platform package is for its content to be reused by other organizations. The submission describes the library of core component and fields that may be used to create new messages.

— Describe the serialization of ‘profiles’ into XML schema[2][3] definition. Also, the submission describes mappings from the BIE profiles to XML schema.

— Describe the serialization of semantic ‘profiles’ into Javascript Serialized Object Notation (JSON) schema Draft 4, and Open API Specification 3.0 schema object representation. JSON syntax has been increasingly popular with business application and API developers ever since the introduction of mobile and cloud-based technologies due to its more compact payload size for specific instances. Combined with the Representational State Transfer (REST) architectural style, JSON syntax is

1) https://oagi.org/

currently what development resources expect to use for modern application integration and what students are learning in college and vocational schools.

Much of OAGi member research has suggested a hybrid approach for REST JSON, taking the 'best of' various existing and historical efforts to standardize metadata definitions for REST include OData,[4] JSON Schema and Open API Specification. The OAGi RESTful Web API Specification describes this hybrid of OData's URI Conventions,[20] JSON Schema Draft 4 (as published by the Internet Engineering Task Force - IETF) which influenced 'Swagger 2.0', vendor specific approaches such as OpenAPI,[5] and ultimately, the Open API Specification 3.0.

Mappings from the BIE profiles to JSON schema are provided, as well as best practices to incorporate in request/response (HTTP GET)[6][7] and synchronous one-way (HTTP POST, PUT, PATCH, DELETE) methods.

Specification for an enterprise canonical model —

Part 1: Architecture

1 Scope

This document specifies the architecture of an enterprise canonical model. It defines the abstract model, conceptual model and physical model.

This document is applicable to organizations implementing an enterprise-wide interoperability capability. Some implementers can find it useful in more targeted applications (e.g. departmental interoperability, projects).

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
application area
structure that conveys information required by senders, receivers and a messaging infrastructure to effectively identify, annotate, transport, route and correlate BOD instances

EXAMPLE Message instance id, sender id and correlation id.

3.2
business object document
BOD
message that conforms to the BOD architecture

3.3
BOD architecture
specification that defines the common data structure and behaviour definition for all OAGIS messages

Note 1 to entry: OAGIS stands for Open Applications Group Integration Specification.

3.4
BODID
unique identifier (UUID) of the message instance

Note 1 to entry: For UUID, see Reference [8].

3.5
BOD instance
message instance of a *BOD* (3.2)

3.6
BOD definition
definition of a *BOD* (3.2) expressed in a specific format (JSON schema or XML schema definition)

Note 1 to entry: For JSON schema, see Reference [9].

3.7
component
data structure composed of substructures and *fields* (3.10)

Note 1 to entry: See the definition of "core component" in ISO 15000-5.

3.8
data area
component that carries the business-semantic payload being communicated by the *BOD* (3.2)

3.9
data type
set of distinct values, characterized by properties of those values, and by operations on those values

Note 1 to entry: A data type can be date, time, decimal, numeric, etc.

[SOURCE: ISO/IEC 11404:2007, 3.12, modified — The term has been changed from "datatype" to "data type"; Note 1 to entry has been added.]"

3.10
field
lowest level of data elements

Note 1 to entry: These elements contain the data values.

3.11
naming convention
set of rules to build a name

Note 1 to entry: Naming conventions are specified for example in ISO/IEC 11179-5[11] and ISO 15000-5[12].

3.12
noun
component (3.7) that specifies the business-specific data being communicated (e.g. purchase order, sales order, quote, route, and shipment)

Note 1 to entry: Nouns are contained within the data area of a *BOD* (3.2).

3.13
receiver
final receiving system of the *BOD instance* (3.5)

3.14
scenario
graphical (using UML sequence diagrams) and textual description of a business process

3.15
sender
application that created a *BOD instance* (3.5)

3.16
signature
digital signature for a *BOD instance* (3.5)

Note 1 to entry: See Reference [13].

3.17
verb
component (3.7) that specifies the behaviour associated with the *noun* (3.12)

4 Cybersecurity

Cybersecurity is critical to information-system design, implementation and operation. Cybersecurity is therefore applicable to systems that build upon the concepts described in this document. However, the concepts described in the ISO 5054 series are at an abstraction level that does not require security considerations. That said, the following brief statement about related work can serve as a preview of cybersecurity-related concepts.

Cybersecurity issues are solved by risk management frameworks. The purpose of risk management is to keep risk at an acceptable level. According to the NIST cybersecurity framework,[14] "understanding the business context, the resources that support critical functions, and the related cybersecurity risks enable an organization to focus and prioritize its efforts, consistent with its risk management strategy and business needs." A key part of risk management is risk assessment. OAGi is working with NIST to explore the development of a business process management framework to be integrated with an information technology (IT) risk assessment approach to support risk assessment automation and help organizations identify their related cybersecurity risks. To this purpose, a next-generation approach and a prototype tool is in development that implements this approach to conduct IT risk assessment continuously and automatically.

5 Message architectures

5.1 General

To achieve interoperability between disparate systems, companies and value chains, there should be a canonical message architecture that provides a common meaning and approach to interoperable business processes and communication.

Messages built upon such a messaging architecture are then used for defining system interactions that establish common processes to enable interoperability. These interactions or scenarios provide a step-by-step guide that is used to perform specific business tasks. Complex scenarios created by assembling basic scenarios with additional messaging steps can then be created to fulfil business functions.

5.2 Abstract message architecture

5.2.1 General

OAGIS (Open Applications Group Integration Specification) supports different realizations of an abstract message architecture (or meta model) shown in Figure 1.

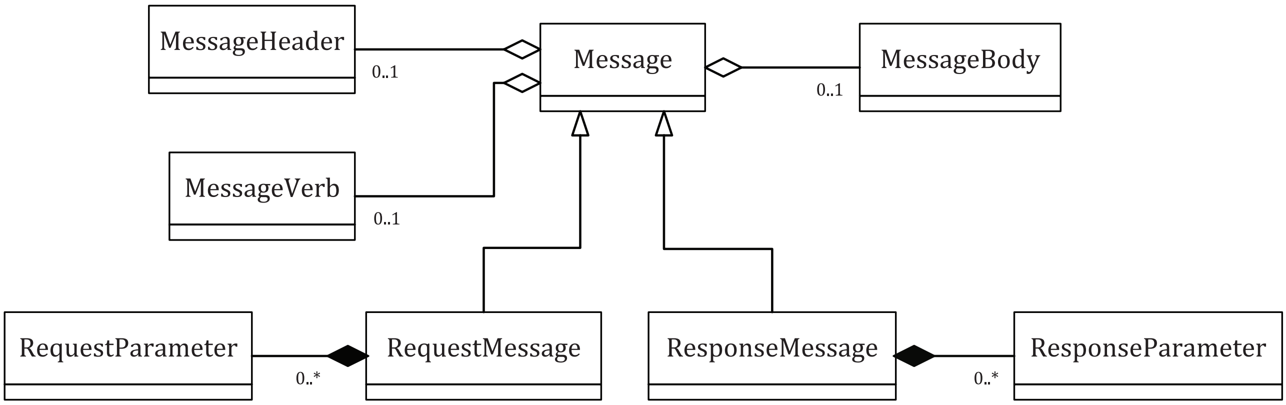

Figure 1 — Abstract message architecture

A `Message` is a definition (or specification) of conveyance of information from a sender to a receiver; it may include a `MessageHeader`, a `MessageVerb` and a `MessageBody`. The `MessageHeader` contains information about the message instance being sent, such as its creation datetime, its origin and destination, message and correlation identifiers, etc. The `MessageVerb` indicates the action or behaviour to be performed by the message receiver. The `MessageBody` contains the business data being managed by the message and may contain metadata related to the `MessageBody` instance, such as the number of records included. There are two types of Messages: `RequestMessage` and `ResponseMessage`. In addition to the message properties already described, a `RequestMessage` may have `RequestParameters`, such as selection and filter criteria.

Subclauses 5.2.2 and 5.2.3 show how OAGIS realizes this generalized message architecture.

5.2.2 Conceptual BOD message architecture

Figure 2 shows the first realization of the abstract message architecture. This realization is known as the BOD message architecture.

The business object document (BOD) realizes the message. The BOD `ApplicationArea` and `DataArea` realize the `MessageHeader` and `MessageBody`, respectively. Only the noun or component of the `DataArea` realizes the `MessageBody`. The verb of the BOD `DataArea` realizes the `MessageVerb`. The BOD request verbs also realize any `RequestParameters` of the `MessageRequest`. Similarly, BOD response verbs realize any `ResponseParameters` of the `MessageResponse`.

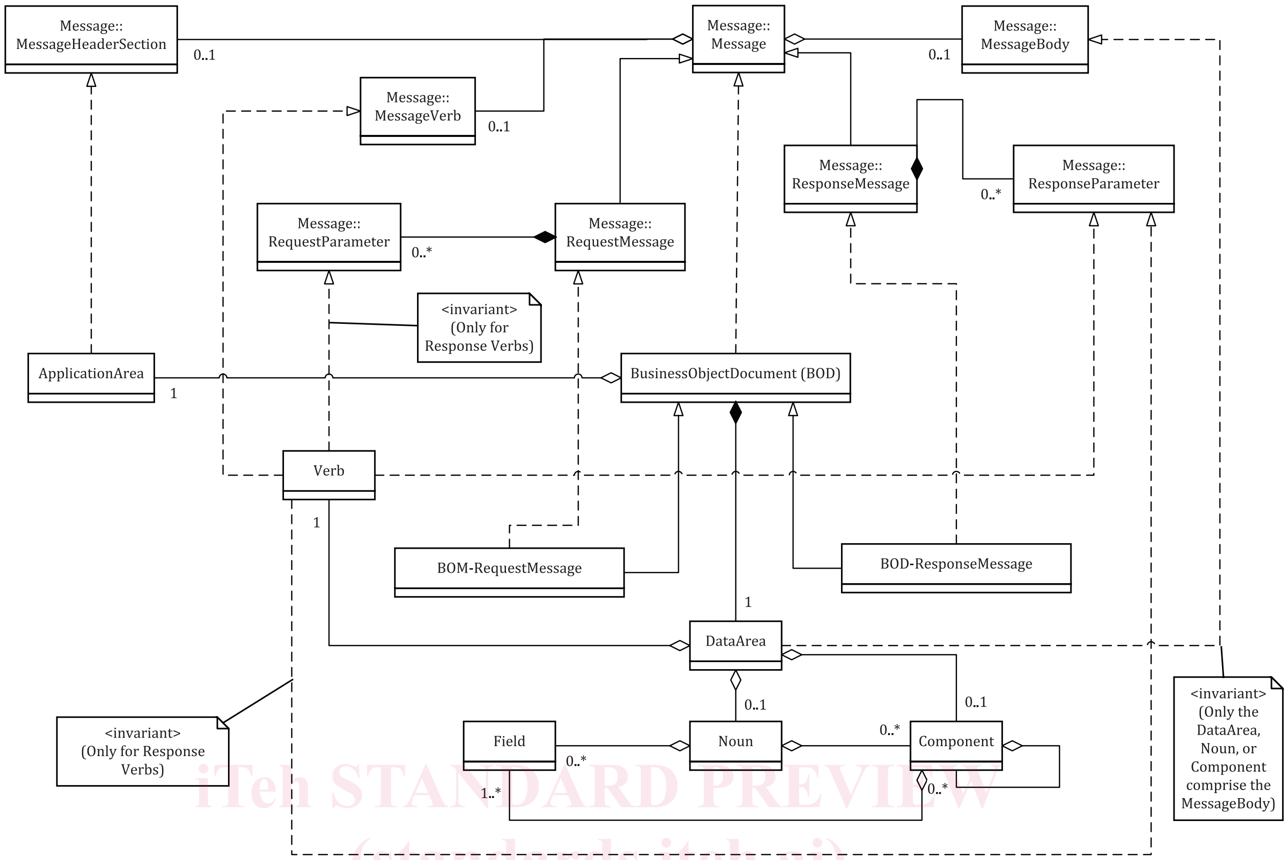

Figure 2 — BOD message architecture

5.2.3 Conceptual RESTful web message architecture

Figure 3 shows the second realization of the abstract message architecture. This realization is known as the RESTful web message architecture. A RESTful web API, as defined herein, is an application programming interface (API) that is offered by a service exposed on the World Wide Web and conforms to the REST (representational entity state transfer) architectural style; it uses the web's primary transfer protocol, the Hypertext Transfer Protocol (HTTP)[15] as the application communication protocol. Generally, an API comprises a set of named operations where each operation includes a request message and an optional response message. RESTful web API operations leverage the HTTP-Message to define their request and response messages (see Annex A). This HTTP-Message architecture[16] serves as the basis for the RESTful web message architecture.