DRAFT INTERNATIONAL STANDARD

ISO/IEC/IEEE/DIS 42010

ISO/IEC JTC **1**/SC **7**

Secretariat: **BIS**

Voting begins on:
2020-12-16

Voting terminates on:
2021-03-10

Software, systems and enterprise — Architecture description

ICS: 35.080

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC/IEEE/DIS 42010:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Contents

Page

Foreword

ISO (International Organization for Standardization) and the 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 (JTC), ISO/IEC JTC 1.

IEEE Standards documents are developed within IEEE Societies and Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of IEEE and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards. Use of IEEE Standards documents is wholly voluntary. IEEE documents are made available for use subject to important notices and legal disclaimers (see https://standards.ieee.org/IPR/disclaimers.html for more information).

IEC collaborates closely with IEEE in accordance with conditions determined by agreement between the two organizations. This Dual Logo International Standard was jointly developed by the IEC and IEEE under the terms of that agreement.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by a minimum of 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from ISO or the IEEE Standards Association.

ISO/IEC IEEE 42010 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Software and Systems Engineering Standards Committee of the Computer Society of the IEEE, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This second edition of ISO/IEC IEEE 42010 cancels and replaces ISO/IEC IEEE 42010:2011, which has been technically revised.

The main changes compared to the previous edition are as follows:

— The term used to refer to the subject of an architecture description is changed from “system” to “entity” to be compatible with ISO/IEC IEEE 42020 and ISO/IEC IEEE 42030 standards and to allow for its application in non-system architecture situations.

— Architecture description element, introduced in the 2011 edition (see 4.2.6, 5.7 and A.6 of that edition) is now defined in Clause 3 as a generic description concept allowing representing at least

stakeholders, concerns, perspectives, and aspects identified in an AD, and views, view components, viewpoints, and model kinds included in an AD.

— Architecture aspect and stakeholder perspective concepts are added to accommodate current practice where these ideas are prevalent.

— Correspondences between architecture descriptions is distinguished from correspondence between architecture description elements.

— The term architecture view component is introduced as a separable portion of one or more architecture views. This change is to account for the fact that some parts of a view are model based while others might not be. View component can be derived from an information source, which can sometimes be a model.

— Model based view components are governed by model kinds. Non-model-based view components are governed by legends.

— The concept of architecture viewpoint is updated to accommodate current practice where a viewpoint governs one or more architecture views.

— The figures of this document use an informal entity-relationship diagram notation to facilitate comprehension by readers of this document. The multiplicities of the relationships are explained in the text when necessary.

— Annex E illustrates a few concepts pertaining to architecture life cycles and architecture description life cycles.

— Annex F show an interpretation of how some Architecture Description Frameworks could comply with requirements of this document.

Introduction

The complexity of human-made entities has grown to an unprecedented level. This has led to new opportunities, and also increased challenges for organizations that create and use these entities. Concepts, principles and procedures of architecting are increasingly applied by organizations, teams and individuals, to help manage the complexity faced by stakeholders of these entities.

Examples of entities include the following: enterprise, organization, solution, system (including software systems), subsystem, processes, business, data (as a data item or data structure), application, information technology (as a collection), mission, product, service, software item, hardware item, product line, family of systems, system of systems, collection of systems, collection of applications, etc.

An architecture of an entity, expressed in an architecture description (AD), assists in understanding of the fundamental properties of the entity, pertaining to its structure, behaviour, design and evolution, such as feasibility, utility and maintainability and fundamental concepts for its development, operation, employment and uses.

Architecture descriptions (ADs) are used by the parties that create, use and manage human-made entities to improve communication and cooperation, enabling all parties, organizations teams and individuals to work together in an integrated, coherent fashion. Architecture description frameworks (ADFs) and architecture description languages (ADLs) are used to codify the conventions and common practices of architecting and the description of architectures within different communities and domains of application.

ADs have many uses, such as design, development, documentation, analysis, evaluation, maintenance, risk mitigation, down-stream user specifications, tool specification, communication, planning, guidance, life cycle support, decision support, review, training, design validation, solution trade studies, cost comparison and analysis, by a variety of stakeholders throughout the life cycles of their entities of interest. Annex D describes a variety of uses of an AD.

This document provides core terms, definitions and relationships for the ADs. The provisions of this document serve to specify desired properties of ADs. This document also gives provisions that specify desired properties of ADFs and ADLs in order to usefully support the development and use of ADs. This document provides a basis on which to consider and compare ADFs and ADLs by providing a common ontology for specifying their content.

This document can be used to establish a coherent practice for developing ADs, ADFs and ADLs within an organization the context of a life cycle of an entity of interest or its architecture. This document can further be used to assess conformance of specifications of ADs, ADFs, ADLs, viewpoints and model kinds with the provisions of this document.

Users of this document are advised to consult Clause 5 to gain appreciation of the conceptual foundations, along with the concepts and principles associated with an AD work product.

Software, systems and enterprise — Architecture description

1 Scope

This document specifies requirements on the structure and expression of architecture descriptions (ADs) for various entities, including software, systems, enterprises, systems of systems, families of systems, products (goods or services), product lines, service lines, technology, and business domains. In this document, the term entity of interest refers to the entity whose architecture is under consideration in the preparation of an architecture description (AD).

This document distinguishes the architecture of an entity of interest from an AD expressing that architecture. ADs, not architectures, are the subject of this document. Whereas an AD is a tangible work product, an architecture is intangible and abstract that can be understood through its concepts, properties and principles.

This document specifies requirements on use of the architectural concepts and their relationships captured in an AD and does not specify requirements for any entity of interest or its environment.

This document specifies requirements on architecture description frameworks (ADFs), architecture description languages (ADLs), architecture viewpoints and model kinds in order to usefully support the development and use of ADs. This document also provides motivations for use of architecture-related terms and concepts in other documents such as guides and standards.

This document specifies conformance to the requirements on ADs, ADFs, ADLs, viewpoints, and model kinds.

This document does not explicitly address completeness or correctness of an AD. Nevertheless, completeness and correctness of an AD can be partially checked, for example, through the consistency of the AD elements established, whether relationships are transitive, and whether AD elements are shown in the respective views. Consistency rules can also be defined with respect to showing the same particular AD element has correspondences with an AD.

This document does not specify the processes, architecting methods, models, notations, techniques or tools by which an AD is created, utilized or managed.

NOTE ISO/IEC IEEE 42020 [17] specifies a set of processes for architecting which can be employed in support of creating one or more ADs. The architecture elaboration process in that standard is especially relevant for creation of ADs.

This document does not specify any format or media for recording ADs. The intent of this document is to enable a range of consistent and coherent approaches to AD including document-centric and model-based techniques.

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 terminological databases for use in standardization at the following addresses:

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

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

— IEEE Standards Dictionary Online: available at https://ieeexplore.ieee.org/xpls/dictionary.jsp

Note 1 to entry Access to the IEEE Standards Dictionary Online requires a free IEEE account. It does not require IEEE membership or any subscription fee.

Definitions for other related terms typically can be found in ISO/IEC IEEE 24765, *System and software engineering – Vocabulary*, available at http://www.computer.org/sevocab.

3.1
architecting
conceiving, defining, expressing, documenting, communicating, certifying proper implementation of, maintaining and improving an architecture (3.2) throughout the life cycle of an entity of interest (3.12)

[SOURCE: ISO/IEC IEEE 42020, modified – The term "entity of interest" replaced "architecture entity".]

3.2
architecture
fundamental concepts or properties related to an entity in its environment (3.13) and governing principles for the realization and evolution of this entity and its related life cycle processes

[SOURCE: ISO/IEC IEEE 42020, modified – The words "related to" replaced "of".]

3.3
architecture aspect
unit of modularization of concerns within an architecture description (3.4), capturing characteristics or features of the entity of interest (3.12)

Note 1 to entry: Aspects enable the architect to analyse, address and structure architecture concerns. In general, there is a many-to-many relation between aspects and concerns. An aspect can pertain either to an entity of interest, to an architecture, or to an environment (such as to a situation or action).

EXAMPLE Functional, structural and informational aspects of an architecture.

Note 2 to entry: See 5.2.5 for more discussion and examples.

Note 3 to entry: The identification of an aspect is often the result of prior knowledge, experience and praxis in the domain to which the aspect applies.

Note 4 to entry: Cross-cutting concerns can be shared among perspectives of stakeholders.

3.4
architecture description
AD
work product used to express an architecture (3.2)

Note 1 to entry: An AD is a tangible representation of information provided to the stakeholders. In other words, it can also be considered as an information item.

Note 2 to entry: A work product is an artefact produced by a process [SOURCE: ISO/IEC 20246:2017, Software and systems engineering — Work product reviews, 3.19]

Note 3 to entry: The subject of an architecture description is the architecture of an entity of interest

3.5
architecture description element
AD element
part of an architecture description (3.4) that expresses the architecture

Note 1 to entry: AD elements include stakeholders, concerns, perspectives, and aspects identified in an AD, and views, view components, viewpoints, and model kinds included in an AD.

Note 2 to entry: For the purpose of correspondences, an architecture description can be considered as an AD element in another Architecture description.

3.6
architecture description framework
ADF
conventions, principles and practices for the description of architectures (3.2) established within a specific domain of application or community of stakeholders (3.18)

EXAMPLE Annex B of Generalized Enterprise-Referencing Architectures Modelling Framework [10], Reference Model of Open Distributed Processing (RM-ODP) [1], Unified Architecture Framework (UAF) [39], and NATO Architecture Framework (NAF) [34].

Note 1 to entry: Architecture description frameworks promote structured organization, consistency of description, greater potential for reuse, and completeness of architecture views and models.

3.7
architecture description language
ADL
means of expression, with syntax and semantics, consisting of a set of representations, conventions, and associated rules intended to be used to describe an architecture

EXAMPLE Architecture Analysis and Design Language (AADL), ArchiMate, UML, SysML, UAF Profile.

3.8
architecture view
information item, governed by an architecture viewpoint (3.9), comprising part of an architecture description (3.4)

Note 1 to entry: A viewpoint is a frame of reference for the concerns determined by the architect as relevant to the purpose of the architecture description.

EXAMPLE A data view in an AD, as described in Clause 4 of ISO/IEC 25024:2015, concerns data for the entity of interest. A data view could include contextual schema, conceptual, logical and physical data models, data dictionary and documents as view components, and entities, relations and attributes as AD elements.

3.9
architecture viewpoint
conventions for the creation, interpretation and use of an architecture view (3.8) to frame one or more concerns (3.10)

Note 1 to entry: A viewpoint is a frame of reference for the concerns determined by the architect as relevant to the purpose of the architecture description.

Note 2 to entry: The conventions of an architecture viewpoint are included in a specification of this viewpoint. In some communities and architecture frameworks, "view specification" is used to mean the same thing.

3.10
concern
matter of relevance or importance to a stakeholder (3.18) regarding an entity of interest (3.12)

Note 1 to entry: Stated concerns are useful when relevant to the purpose of the architecting effort and refer to specific rather than categorical difficulties, problems, or requirements.

Note 2 to entry: Concern expression takes many forms, including among others: as questions about entity features or characteristics, as a keyword label for many related matters, and as expected quality attributes of the entity.

Note 3 to entry: See 5.2.3 for more discussion and examples.

3.11
correspondence
expression of a relationship among architecture description elements (3.5) or among architecture descriptions (3.4)

EXAMPLE Correspondences are used to express a wide range of relationships, such as equivalence, composition, refinement, consistency, traceability, dependency, constraint, satisfaction, and obligation.

3.12
entity of interest
subject of an architecture description (3.4)

EXAMPLE Enterprise, organization, solution, system (including software systems), subsystem, processes, business, data (as a data item or data structure), application, information technology (as a collection), mission, product, service, process, software item, hardware item, product line, family of systems, system of systems, collection of systems, collection of applications, etc.

3.13
environment
aggregate of surrounding things, conditions, contexts of, or influences upon an entity of interest

Note 1 to entry: The environment of an entity of interest includes external entities that can have various influences upon the entity of interest, such as developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences as well as external physical effects such as electromagnetic radiation, charged particles, gravitational effects, and electric and magnetic fields.

Note 2 to entry: A label attached as a qualifier to the word *environment* identifies a particular context within that environment, such as development environment, test environment, and operational environment. It would be more correct to refer to these as development context, test context, operational context, etc. A context can be to help understand an entity or its architecture, including the derivation of an architecture.

3.14
frame
to formulate or construct in a particular style or language

3.15
information item
separately identifiable body of information that is produced, stored, and delivered for human and machine use

[SOURCE: ISO/IEC IEEE 15289:2019, modified – The words “human and machine use” replaced “human use”]

3.16
model kind
category of model distinguished by its key characteristics and modelling conventions

EXAMPLE Functional models, activity models, structural models, use case diagrams, geopolitical models and economic models.

3.17
specification
information item that identifies, in a complete, precise and verifiable manner, the requirements, design, behaviour, or other expected characteristics of a system, service, or process

[SOURCE: ISO/IEC IEEE 15289:2019, 3.1.26]

3.18
stakeholder
role, position, individual, organization or classes thereof, having an interest, right, share, or claim, in an entity or its architecture (3.2)

EXAMPLE End users, operators, acquirers, owners, suppliers, architects, developers, builders, maintainers, regulators, taxpayers, certifying agencies, and markets.

3.19
stakeholder perspective
way of thinking about an entity, especially as it relates to concerns (3.10)

EXAMPLE The labels given to the middle three rows (i.e. owner, designer and builder) of the Zachman framework [56] correspond to stakeholder perspectives. The rows in the Unified Architecture Framework [39] and NATO Architecture Framework [34] grids correspond to stakeholder perspectives (although they are called "domains" and "subjects of concerns," respectively in those frameworks). See 5.2.4 for more examples.

Note 1 to entry: The way one thinks about an entity can be influenced by one's beliefs, training, experience, knowledge, personality, character traits, culture, peer pressure, role or stance etc.

3.20
view component
architecture view component
separable portion of one or more architecture views (3.8) that is governed by the applicable model kind (3.16) or legend

EXAMPLE An architecture view component describing a firewall can be used in several views of an architecture description to explain functional flows, behaviour and security features of an entity.

4 Conformance

The requirements in this document are contained in Clauses 6, 7 and 8. There are five situations in which claims of conformance with the provisions of this document can be made.

1) When conformance is claimed for an architecture description, the claim shall demonstrate that the specification of the architecture description meets the requirements listed in Clause 6.

2) When conformance is claimed for an architecture description framework, the claim shall demonstrate that the specification of the architecture description framework meets the requirements listed in 7.1.

3) When conformance is claimed for an architecture description language, the claim shall demonstrate that the specification of the architecture description language meets the requirements listed in 7.2.

4) When conformance is claimed for an architecture viewpoint, the claim shall demonstrate that the specification of the architecture viewpoint meets the requirements listed in 8.1.

5) When conformance is claimed for a model kind, the claim shall demonstrate that the specification of the model kind meets the requirements listed in 8.2.

Requirements of this document are marked by the use of the verb "shall." Recommendations are marked by the use of the verb "should." Permissions are marked by the use of the verb "may." In the event of a conflict between normative figures and text, the text takes precedence. Please report any apparent conflicts.

NOTE This document is designed such that "tailoring" is neither required nor permitted for its use when claims of conformance are made.

5 Conceptual foundations

5.1 Introduction

This clause introduces the conceptual foundations of architecture description expressed in a set of conceptual models (see 5.2) and its application to ADs, ADFs (see 5.4.2) and ADLs (see 5.4.3). The concepts introduced in this clause are used in Clauses 6 through 8 to express requirements.

NOTE Annex A provides further discussion of the terms and concepts used in this document and presents examples of their use in an historical context.

5.2 Conceptual models of an architecture description

5.2.1 Context of an architecture description

The term *entity of interest* is used in this document to refer to the subject of an architecture description. The term is intended to encompass, but is not limited to, entities within the following *fields of application*, reflecting the intended scope of this document as specified in Clause 1.

— software, including software products and services per ISO/IEC IEEE 12207 [4];

— systems as discussed in ISO/IEC IEEE 15288 [5];

— enterprises as described in ISO 15704 [10], i.e. human undertakings or ventures that have mission, goals and objectives to offer products or services, or to achieve a desired project outcome or business outcome.

This document takes no position on what constitutes an entity within those fields of application or elsewhere. An entity can be a concrete entity or an abstract entity. An AD as specified in this document is suitable for not only entities in the fields of applications listed above, but also for entities in fields such as natural systems or conceptual systems.

Each entity of interest is situated in an environment. The environment determines the totality of influences upon the entity of interest and the totality of influences of the entity of interest upon that environment, including its interactions with the environment and other entities, throughout the life cycle of that entity of interest. The environment of an entity of interest can influence other entities or can be influenced by other entities.

Figure 1 depicts key concepts pertaining to entity of interest and their architectures as a context for understanding ADs.

NOTE The figures and text in the remainder of 5.2 constitute a set of conceptual models of an AD. These figures use an informal entity-relationship diagram notation to facilitate comprehension by readers of this document. In the figures rounded rectangles represent information objects, and arrows represent relationships between objects with the annotation read in the arrow direction. The figures illustrate the key concepts described throughout Clause 5.

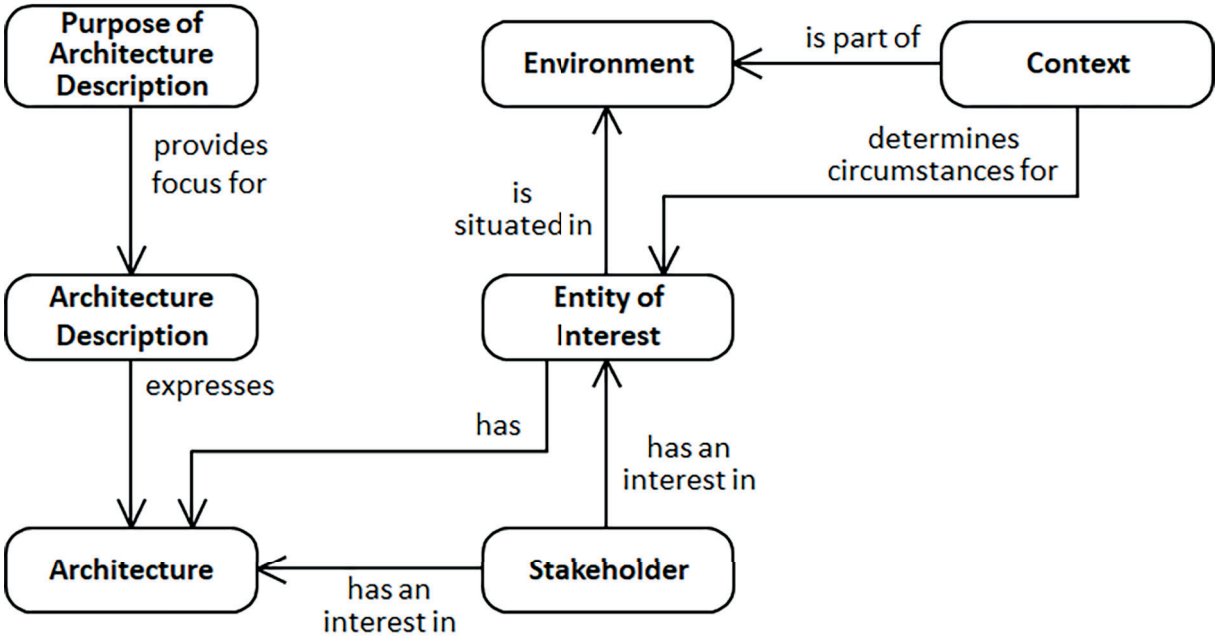

Figure 1 — Context of architecture description

5.2.2 Architectures and architecture descriptions

The architecture of an entity of interest comprises the fundamental concepts or properties of that entity considered in its environment. The architecture of an entity of interest can pertain to any or all of the entity's:

— constituent elements;

— interactions or interrelationships among its elements;

— interactions or interrelationships with other entities in its environment or the environment itself;

— behaviour and structure; and

— principles governing its design, use, operation and evolution.

An AD is an expression of an architecture. An AD comprises AD elements (see 5.2.10).

ADs are work products resulting from architecting efforts. As a work product, an AD is devised for the specific purpose for which the architecting effort is undertaken, which is distinct from the purpose of the entity of interest.

The architecture of an entity of interest can be understood through one or more distinct ADs, each created for a purpose relative to the architecture and stakeholder needs. Different ADs can, for example, be based on different stakeholders (or stakeholder perspectives), time periods (sometimes termed *epochs*), or specific contexts or usage within the environment.

NOTE ISO/IEC IEEE 42020 [16] specifies a set of processes for architecting which can be employed in support of creating one or more ADs.

5.2.3 Stakeholders and concerns

Stakeholders are parties with interests in an entity, its architecture, or its architecture description. Among the stakeholders are those parties that have influence or control over or are impacted by an entity or its architecture. A stakeholder's interests are typically expressed as concerns about an