INTERNATIONAL STANDARD

ISO/IEC/IEEE 31320-1

First edition
2012-09-15

Information technology — Modeling Languages —

Part 1: Syntax and Semantics for IDEF0

Technologies de l'information — Langages de modélisation —

Partie 1: Syntaxe et sémantique pour IDEF0

Reference number
ISO/IEC/IEEE 31320-1:2012(E)

ISO copyright office
Case postale 56
CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
E-mail inmail@iec.ch
Web www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York
NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web www.ieee.org

Published in Switzerland

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.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The 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 the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

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 at least 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 31320-1 was prepared by the Software & Systems Engineering Standards Committee of the IEEE Computer Society (as IEEE 1320.1-1998). It was adopted by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in parallel with its approval by the ISO/IEC national bodies, under the “fast-track procedure” defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE. IEEE is responsible for the maintenance of this document with participation and input from ISO/IEC national bodies.

ISO/IEC/IEEE 31320 consists of the following parts:

— ISO/IEC/IEEE 31320-1, *Information technology — Modeling Languages — Part 1: Syntax and Semantics for IDEF0*

— ISO/IEC/IEEE 31320-2, *Information technology — Modeling Languages — Part 2: Syntax and Semantics for* $IDEF1X_{97}$ ($IDEF_{object}$)

(blank page)

IEEE Std 1320.1-1998(R2004)

IEEE Standard for Functional Modeling Language—Syntax and Semantics for IDEF0

Sponsor

Software Engineering Standards Committee
of the
IEEE Computer Society

Approved 25 June 1998

IEEE-SA Standards Board

Abstract: IDEF0 function modeling is designed to represent the decisions, actions, and activities of an existing or prospective organization or system. IDEF0 graphics and accom-panying texts are presented in an organized and systematic way to gain understanding, support analysis, provide logic for potential changes, specify requirements, and support system-level design and integration activities. IDEF0 may be used to model a wide variety of systems, composed of people, machines, materials, computers, and information of all varieties and structured by the relationships among them, both automated and nonautomated. For new systems, IDEF0 may be used first to define requirements and to specify functions to be carried out by the future system. As the basis of this architecture, IDEF0 may then be used to design an implementation that meets these requirements and performs these functions. For existing systems, IDEF0 can be used to analyze the functions that the system performs and to record the means by which these are done.

Keywords: enterprise, functional modeling language, IDEF0, language, modeling language

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 0-7381-0340-3

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require 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. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Introduction

(This introduction is not part of IEEE Std 1320.1-1998, IEEE Standard for Functional Modeling Language—Syntax and Semantics for IDEF0.)

This standard formally documents the Integration Definition 0 (IDEF0) language for function models in two parts. The body of this standard describes the syntax and semantics of the IDEF0 language that are required to draw the physical diagrams of a *specific* IDEF0 model. Annex B describes the syntax and semantics of the IDEF0 language as an *abstract* formal structure and, therefore, provides the foundation for the specifics found in the body of the standard. The diagrams discussed by the standard are real instantiations in a concrete model of the mathematical formalisms of the IDEF0 Language Abstract Formalization (the “language formalization”).

Background

During the 1970s, the US Air Force Program for Integrated Computer Aided Manufacturing (ICAM) sought to increase manufacturing productivity through systematic application of computer technology. The ICAM program identified the need for better analysis and communication techniques for people involved in improving manufacturing productivity.

As a result, the ICAM program developed a series of modeling methodologies known as the ICAM Definition (IDEF) methods, which include

a) IDEF0, used to produce a function model. A function model is a structured representation of the functions within a system or subject area.

b) IDEF1, used to produce an information model. An information model represents the structure and semantics of information within a system or subject area.

c) IDEF2, used to produce a dynamics model. A dynamics model represents the behavior of a system or subject area as it varies over time.

IDEF0 was derived from a well-established graphical modeling method known as the Structured Analysis and Design Technique (SADT). IDEF0 was developed by the originators of SADT, notably Douglas T. Ross, under the ICAM program. In 1983, the US Air Force Integrated Information Support System program enhanced the IDEF1 information modeling technique to develop IDEF1 extended (IDEF1X), a semantic data modeling method.

Continued IDEF method developments followed to address needs for additional analytic methods. These follow-on developments have been directed toward providing a mutually supportive family of methods that are applicable to a broad range of enterprise improvement and integration strategies (e.g., concurrent engineering, total quality management, business reengineering). Reflecting this general applicability, the IDEF acronym has been recast to refer to an integrated family of Integration Definition methods. Currently, IDEF0 and IDEF1X techniques are widely used in the government, industrial, and commercial sectors, supporting modeling efforts for a wide range of enterprises and application domains. IDEF0 has been widely adopted as the function modeling method of choice in large number of both military and nonmilitary organizations in both North America and Europe.

In 1991, the National Institute of Standards and Technology (NIST) received support from the US Department of Defense, Office of Corporate Information Management (DoD/CIM), to develop Federal Information Processing Standards (FIPS) for modeling techniques for use by the federal government. One product of this effort was FIPS PUB 183, *Integration Definition for Function Modeling (IDEF0)* [B2].[a] This FIPS document was based on the IDEF specification manuals published by the US Air Force in the early 1980s. At the same time, NIST also published FIPS PUB 184, *Integration Definition for Information Modeling (IDEF1X)* [B3], to support data modeling for the federal government.

In 1993, the Institute of Electrical and Electronics Engineers (IEEE) Computer Society initiated a project to establish IDEF standards across both industry and government within the standards framework of the American National Standards Institute (ANSI). IEEE Std 1320.1-1998 for IDEF0 function models, based on FIPS PUB 183 [B2], is a result of that effort.

This standard is explicitly oriented to the presentation of an IDEF0 model on paper pages; development of an IDEF0 standard for other presentation media is deliberately not addressed by this document. However, integrated into this standard is a mathematically founded formalization of an IDEF0 model. This formalization allows users to separate what they conceive in conceptual space, that is, the model itself, from their presentation of that model and from their presentation media. In earlier work, the IDEF0 diagram was not considered as something that should or could be distinguished from the paper page that presents that diagram. However, the current formalization allows (indeed, forces) users to separate the abstract structure of an IDEF0 model from the physical structure of the presentation of that model using sheets of paper. An important conceptual and notational difference between this work and earlier work is the clear distinction between an IDEF0 graphic diagram and the medium, e.g., a page of paper, that is used to present that diagram.

New terminology has been presented to ensure that this distinction can be easily maintained, and a more robust categorization of both the components of an IDEF0 model and of an IDEF0 diagram has been introduced. This terminology ensures that the usage presented in the body of this standard is consistent with the formalization presented in Annex B. In addition, these changes to the IDEF0 vocabulary will facilitate the development of IDEF0 presentations in digital or other media.

The IDEF0 approach

IDEF0 includes both a modeling language and a comprehensive methodology for developing models. This standard addresses only the syntax and semantics of the modeling language itself.

In addition to definition of the IDEF0 language, the IDEF0 methodology also prescribes procedures and techniques for developing and interpreting models, including ones for data gathering, diagram construction, review cycles, and documentation.

IDEF0 function modeling is designed to represent the decisions, actions, and activities of an existing or prospective organization or system. For all its apparent simplicity, the method is surprisingly powerful and effective. Like most modeling methods, the primary component of IDEF0 is a graphical language whose constructs are intended to convey information of a certain sort. IDEF0 graphics and accompanying texts are presented in an organized and systematic way to gain understanding, support analysis, provide logic for potential changes, specify requirements, and support systems-level design and integration activities. IDEF0 may be used to model a wide variety of systems, composed of people, machines, materials, computers, and information of all varieties and structured by the relationships among them, both automated and nonautomated. For new systems, IDEF0 may be used first to define requirements and to specify functions to be carried out by the future system. As the basis of this architecture, IDEF0 may then be used to design an implementation that meets these requirements and performs these functions. For

[a] The numbers in brackets correspond to those of the bibliography in Annex A.

existing systems, IDEF0 can be used to analyze the functions that the system performs and to record the means by which these are done.

The result of applying IDEF0 to a system is a model that consists of a hierarchical series of diagrams, with accompanying explanatory text, illuminating graphical, and defining glossary pages that are cross-referenced to these diagrams. The two primary modeling components of a diagram are functions (represented by named boxes) and the physical and data objects that interrelate those functions (represented by labeled arrows).

As a function modeling language, IDEF0 has these characteristics:

a) It is comprehensive and expressive, capable of graphically representing a wide variety of business, manufacturing, and other types of enterprise operations to any level of detail.

b) It is a coherent and simple language, allowing rigorous and precise expression and promoting consistency of usage and interpretation.

c) It enhances communication among analysts, architects, developers, managers, and users through its ease of learning and its emphasis on hierarchical exposition of detail.

d) It is well-tested and proven through many years of use by the USAir Force and other government agencies and by private industry.

e) It can be generated by a variety of computer-based tools; several commercial products specifically support development and analysis of IDEF0 diagrams and models.

As a system engineering technique, IDEF0 may be used for performing and managing needs analysis, benefits analysis, requirements definition, functional analysis, systems design, maintenance, and baselines for continuous improvement. IDEF0 models provide a “blueprint” of functions and their interfaces that must be captured and understood in order to make systems engineering decisions that are logical, affordable, integratable, and achievable. When used in a systematic way, IDEF0 provides a systems engineering approach to

a) Performing systems analysis and design at all levels, including the entire enterprise, a system, or a subject area;

b) Producing reference documentation concurrent with development to serve as a basis for integrating new systems or improving existing systems;

c) Allowing collaborative team consensus to be achieved by shared understanding;

d) Managing large and complex projects using qualitative measures of progress; and

e) Providing a reference architecture for enterprise analysis, information engineering, and resource management.

Typographic conventions

A word that has a special meaning for IDEF0 is italicized the first time that it is used in its specific sense for IDEF0. There will be an entry in Clause 2 for each italicized word.

Figure conventions

The figures in this document have been prepared using the IDEF standard diagram form (SDF). The SDF and its use are documented by both FIPS PUB 183 [B2] and FIPS PUB 184 [B3]. FIPS PUB 183 [B2] does not standardize the SDF; instead, the SDF is covered in an informative annex. In keeping with this precedent, this standard also refrains from addressing the SDF as an element of IDEF0 language standardization.

The IDEF SDF is designed for use and presentation on conventional 8½-by-11-inch paper sheets. Therefore, to preserve this conventional size, the figures for this document have been collected at the end of the document rather than being interspersed with the text.

Contents

Annex C (informative) Examples of IDEF0 usage and style 54

IEEE Standard for Functional Modeling Language— Syntax and Semantics for IDEF0

1. Overview

This standard provides requirements for the construction of semantically and syntactically correct Integration Definition 0 (IDEF0) models and diagrams. A model or diagram conforms to this standard if it adheres to all mandatory provisions (marked by the verbs "shall" or "is") of this standard.

Annex C of this document collects the figures that are referenced in the body of the standard and in Annex B.

1.1 Scope

The body of the standard describes the modeling language (syntax and semantics) that supports the IDEF0 method for developing graphical representations of a system or subject area. The clauses that follow govern the physical construction of IDEF0 models that represent functions, functional relationships, and the physical and data objects required by those relationships.

This part of the document is divided into 10 clauses. Clause 1 provides an overview of this part of the standard. Clause 2 defines key terms. Clause 3 discusses the concept of an IDEF0 model. Clause 4 defines the syntax of the IDEF0 language. Clause 5 defines the semantics of the language. Clause 6 describes the different types of IDEF0 diagrams. Clause 7 presents the different types of IDEF0 model pages. Clause 8 provides details on the various features of an IDEF0 diagram. Clause 9 defines IDEF0 reference expressions. Finally, Clause 10 defines IDEF0 diagram feature references.

Documentation of best commercial practices and guides to recommended usage are beyond the scope of this document.

1.2 Purpose

The objectives of this part of the standard are to prescribe the construction and components of an IDEF0 model and to define the correct syntax and semantics for construction of an IDEF0 diagram within an IDEF0 model.

2. Definitions, acronyms, and abbreviations

This clause contains definitions of important IDEF0 terms used in this standard. Terms used in a definition that are defined elsewhere in this clause are italicized.

2.1 Definitions

2.1.1 A-0 context diagram: The only *context diagram* that is a required for a valid *IDEF0 model*, the *A-0* diagram contains one *box*, which represents the top-level *function* being modeled, the *inputs*, *controls*, *outputs*, and *mechanisms* attached to this *box*, the full *model name*, the *model name abbreviation*, the model's *purpose statement*, and the model's *viewpoint statement*.

2.1.2 A-0 diagram: See: *A-0 context diagram*.

2.1.3 activation: One occurrence of a *function*'s transformation of some subset of its *inputs* into some subset of its *outputs*.

2.1.4 activation constraint: A *function*'s requirement for the presence of a nonempty *object set* in a particular *arrow role* as a precondition for some *activation* of the *function*.

2.1.5 activity: See: *function*.

2.1.6 ancestral box: A *box* related to a specific *diagram* by a *hierarchically consecutive* sequence of one or more parent/child relationships.

2.1.7 ancestral diagram: A *diagram* that contains an *ancestral box*.

2.1.8 arrow: A directed line, composed of one or more connected *arrow segments* in a single *diagram* from a single source (*box* or *diagram boundary*) to a single use (*box* or *diagram boundary*). See also: *arrow segment*, *boundary arrow*, *internal arrow*.

2.1.9 arrow label: A noun or noun phrase associated with an *arrow segment* to signify the *arrow meaning* of the *arrow segment*. Specifically, an arrow label identifies the *object type set* that is represented by an *arrow segment*.

2.1.10 arrow meaning: The *object types* (e.g., a physical thing, a data element) of an *object type set*, regardless of how these *object types* may be collected, aggregated, grouped, bundled, or otherwise joined within the *object type set*.

2.1.11 arrow reference: See: *ICOM code*.

2.1.12 arrow role: The relationship between an *object type set* represented by an *arrow segment* and the *activity* represented by the *box* to which the *arrow segment* is attached. There are four arrow roles: *input*, *control*, *output*, and *mechanism*.

2.1.13 arrow segment: A directed line that originates at a *box* side, arrow *junction* (*branch* or *join*), or *diagram boundary* and terminates at the next *box* side, arrow *junction* (*branch* or *join*), or *diagram boundary* that occurs in the path of the line.

2.1.14 boundary arrow: An *arrow* with one end (source or use) not connected to any *box* in a *diagram*. Contrast: *internal arrow*.

2.1.15 boundary ICOM code: An *ICOM code* that maps an untunneled *boundary arrow* in a *child diagram* to an *arrow* attached to the *parent box* that is detailed by that *diagram*.

2.1.16 box: A rectangle containing a *box name*, a *box number*, and possibly a *box detail reference* andrepresenting a *function* in a *diagram*.

2.1.17 box detail reference: A square enclosure encompassing a *box number,* which indicates that the *box* is decomposed or detailed by a *child diagram*.

2.1.18 box ICOM code: An *ICOM code* that maps a tunneled *boundary arrow* to an *arrow* attached to some *ancestral box*.

2.1.19 box name: The verb or verb phrase placed inside a *box* that names the modeled *function*. A *box*takes as its box name the *function name* of the *function* represented by the *box*. See: *function name*.

2.1.20 box number: A single digit (***0, 1, 2, …, 9***) placed in the lower right corner of a *box* to uniquelyidentify that *box* in a *diagram*. The only *box* that may be numbered ***0*** is the *box* that represents the ***A0*** function in ***A-0*** and ***A-1*** *context diagrams*.

2.1.21 branch: A *junction* at which a *root arrow segment* (going from source to use) divides into two or more *arrow segments*. May denote *unbundling* of *arrow meaning*, i.e., the separation of *object types* from an *object type set*. Also refers to an *arrow segment* into which a *root arrow segment* has been divided.

2.1.22 bundle: (A) (As a verb) To combine separate *arrow meanings* into a composite *arrow meaning*, expressed by joining *arrow segments*, i.e., the inclusion of multiple *object types* into an *object type set*. (B) (As a noun) An *arrow segment* that collects multiple meanings into a single construct or abstraction, i.e., an *arrow segment* that represents an *object type set* that includes more than one *object type*.

2.1.23 call arrow: An *arrow* that enables the sharing of detail between *IDEF0 models* (linking them together) or within an *IDEF0 model*. The tail of a call arrow is attached to the bottom side of a *box*. One or more *page references* are attached to a call arrow.

2.1.24 call reference: A *page reference* attached to a *call arrow.*

2.1.25 called diagram: A *decomposition diagram* invoked by a *calling box* and identified by a *page reference* attached to a *call arrow*.

2.1.26 calling box: A *box* that is detailed by a *decomposition diagram* that is not the *box*'s *child diagram*. A *call arrow* is attached to the bottom of a calling box.

2.1.27 child box: A *box* in a *child diagram*.

2.1.28 child diagram: A *decomposition diagram* related to a specific *box* by exactly one child/parent relationship.

2.1.29 complete ICOM code: A *diagram feature reference* in which *dot notation* joins an *ICOM code* to a *diagram reference*.

2.1.30 context: The immediate environment in which a *function* (or set of *functions* in a *diagram*) operates.

2.1.31 context diagram: A *diagram* that presents the *context* of the *top-level function* of an *IDEF0 model*, whose *diagram number* is ***A-n***, where ***0≤n≤9***. The one-box *A-0 context diagram* is a required context diagram; those with *diagram numbers* ***A-1, A-2***, …, ***A-9*** are *optional context diagrams*.