INTERNATIONAL STANDARD

ISO 14258

First edition
1998-09-01

Industrial automation systems — Concepts and rules for enterprise models

Systèmes d'automatisation industrielle — Concepts et règles pour modèles d'entreprise

Reference number
ISO 14258:1998(E)

Contents

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet iso@iso.ch

Printed in Switzerland

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.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO 14258 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 5, *Architecture and communications*.

Annex A of this International Standard is for information only.

Introduction

The major objective of this International Standard is to define concepts and rules for enterprise models (see clause 3) with the intent to guide and constrain other standards or implementations that do or will exist on the topic. It accomplishes this by defining the elements to use when producing an enterprise model (see 3.2), concepts for life-cycle phases (see 3.3), and how these models describe hierarchy (see 3.4), structure (see 3.5), and behavior (see 3.6). This International Standard provides guidelines and constraints for enterprise models to anyone attempting to model an enterprise or to model processes (see 3.7).

The users of this International Standard are primarily the standards bodies making more detailed standards about a part of the integration and modeling domain. Systems implementers may also find value in the structure developed in this International Standard so that their developments parallel the concepts outlined herein. If conforming implementation designs have the same technology areas and nomenclature, or are able to map to them readily, the information of one enterprise or process can be more readily shared with information of another enterprise or process (see 3.8).

The rationale for this International Standard is that other well-designed standards in the domain of enterprise integration and modeling are needed to provide a known environment to enterprise designers. Thus, the risk of investing in islands of integration can be significantly reduced. Where an island does exist, then these standards assist the designer to create the translation required for the island to interact with the known environment. A standard for enterprise models should enhance interoperability by establishing the elements that must be present in an enterprise model. These elements will come into play when there is need for one process to communicate with another.

Industrial automation systems — Concepts and rules for enterprise models

1 Scope

This International Standard specifies concepts and rules for computer-understandable models of a manufacturing enterprise to better enable enterprise processes to interoperate.

This International Standard does not define standard enterprise processes, standard enterprises, standard organizational structures, or standard enterprise data. In addition, this International Standard does not specify the enterprise-modeling process, but forms the basis by which enterprise-modeling standards can be developed where they are needed.

2 Definitions

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

2.1 Definitions of enterprise concepts

2.1.1
enterprise
a group of organizations sharing a set of goals and objectives to offer products, services or both

2.1.2
environment
the uncontrollable part of a system which is widened to the extent that a decision-taking procedure cannot be conceived for the control of such a system

2.1.3
factors of production
that which is required to transform, transport, store, and verify raw materials, parts, (sub-) assemblies, and end products

2.1.4
user of standard
one who applies the requirements of this International Standard for whatever purpose

EXAMPLE 1 Enterprise planners, builders, modifiers, and analyzers using the requirements to check completeness of their activity.

EXAMPLE 2 Enterprise-model builders using the requirements to assure consistency between models to enable model interoperability.

EXAMPLE 3 Developers of standards for enterprise representation using the requirements to assure consistency between their standards and this International Standard.

2.2 Definitions of model concepts

2.2.1
abstraction
a shortening in duration or extent with no sacrifice of sense, used to differentiate between a real-world system and a model of the real world

2.2.2
behavior
how an element acts and reacts

2.2.3
constraint
restrictions and limitations on the system which can come from inside or outside the system under consideration

2.2.4
element
a basic system part that has the characteristics of state, behavior, and identification

2.2.5
enterprise model
a representation of what an enterprise intends to accomplish, how it operates and possibly how it is organized, which is used to improve the effectiveness and efficiency of the enterprise

NOTE – An enterprise model is an abstraction that represents the basic elements of an enterprise and their decomposition to any necessary degree. It also specifies the information requirements of these elements, and provides the information needed to define the requirements for integrated information systems.

2.2.6
model
a representation of something else expressed in mathematics, symbols, or words

NOTE – A model is an abstraction that represents one's understanding of a system or situation, and of the relevant elements and relationships. It represents the system elements and the connectivity between the elements.

3 Concepts and rules

3.1 Purpose of enterprise models (informative)

Enterprise models are used as tools to describe and represent an enterprise in the context of a given purpose. Enterprises are systems that can be analyzed and modeled using systems theory. These models can be constructed to analyze, guide the engineering of, and manage the operation of enterprises.

The rules presented in clause 3 are designed to support these usages and to allow information transfer between enterprise models.

3.2 Systems theory as a basis for enterprise models

Enterprise models conforming to this International Standard shall be constructed to conform to the relevant elements of systems theory. The normative concepts and rules presented in clause 3 describe these relevant elements of systems theory and relate them to model content and characteristics.

3.2.1 Methodologies derived from systems theory (informative)

In literature[1] there are various methodologies derived from general systems theory which emphasize different aspects. The three most frequently used aspects are the structural aspect, behavioral aspect, and hierarchical aspect.

[1] *An Approach to General Systems Theory*, George J. Klir (1969); *An Introduction to General Systems Thinking*, Gerald M. Weinberg (1975).

The structural aspect is based on the principle that elements are not isolated but have multiple interdependencies with other elements of the system. The interdependencies are the explanation why the whole (system) exhibits properties different from the properties of its parts (elements).

The behavioral aspect is based on the identification of variables and their functional or other relationships. If the variables are restricted to input and output variables the system is considered as a black box.

The hierarchical aspect is based on the principle that an element of a system can itself be regarded as a system, which is then called a subsystem. Similarly the system under consideration can be regarded as an element of another system, which is then called a supersystem. This implies the assignment of levels of abstraction to systems. Because of interdependence new properties can emerge at a higher level in the hierarchy.

Steps to lower levels allow one to obtain more detailed description of the system under consideration and how it achieves its purpose. Steps to higher levels allow one to understand the role of the system within its environment.

Each level is describable in terms of structure and behavior. Depending on the desired purpose particular methodologies are applicable. Steps downward expose the inner structure of the subsystem. This could be achieved by observation, by logical conclusion, or by design in the case of systems under development. Steps upward expose the behavior of the system in its environment. Similarly this could be achieved by observation, by logical conclusion, or by making assumptions in the case of systems under development.

3.2.2 Factors of production

Enterprise models shall address what happens to the factors of production (such as people, capital, material, information, energy, and tools) during the phases of the enterprise or product life cycle.

3.2.3 Scope of enterprise models

Enterprise models shall define relevant aspects of the enterprise necessary to

- conceive, design, procure for, and construct an enterprise consisting of any set of related chosen processes,
- manage and operate an enterprise so that it can meet its objectives,
- support an enterprise to design, modify, operate, or dismantle it.

3.2.4 Availability and format of model information

In an operating scenario the information captured by an enterprise model shall be available to humans or machines responsible for successful operations. The information shall be either in a neutral format (preferable) or as specified by the using application.

3.2.5 Semantics and syntax of an enterprise model

Models, as representations of enterprises, shall carry syntax and semantics. The syntax of a model refers to the permissible arrangements of the representations of the elements and to the permissible kinds of relations. The semantics of a model encompass the meanings of the elements and relations with respect to enterprise-model concepts. The syntactic form and semantic content of a model will be different depending, for example, on the purpose of the model and on the boundary and the environment of the enterprise.

3.2.6 Management of constituent parts

Enterprise models shall be designed in such a way as to allow their constituent parts to be managed by an automated configuration-management system.

3.3 Concepts for life-cycle phases (informative)

Products, processes, projects, and enterprises are systems. Systems have a life cycle that can be partitioned into phases such as plan/build, use/operate, and recycle/ dispose.

3.3.1 Issue-solving activities

Three activities are required to solve issues found within each high-level system life-cycle phase (plan/build, use/operate, recycle/dispose). These activities are

— find out what to do (activity W),

— find out how to do it (activity H),

— do it (activity D).

Figure 1 illustrates a mapping between common names for system life-cycle phases and the what, how, and do activities.

The activities W, H, and D may be represented by different types of models. These models shall have the capability to interoperate where it has been determined that these activities need to communicate with each other.

	"What" activities	**"How" activities**	**"Do" activities**
Plan and build phase (e.g. before sell/buy title transfer)	• Develop goals • Define strategy • Define product needs	• Develop requirements • Define concept • Design product • Plan to produce product • Plan to support product	• Procure parts • Produce product • Test product • Ship product
Use and operate phase (e.g. after sell/buy title transfer)	• Define support needs • Define use	• Define use requirements • Define support requirements	• Use the product • Support product
Dispose and recycle phase (e.g. after product is no longer useful)	• Define recycle/ dispose needs	• Define recycle/ dispose requirements	• Recycle product • Dispose product

Figure 1 — Mapping between system life-cycle phases and system activities W, H, and D

3.3.2 Life cycle of systems

Different life-cycle phases may have different models. These models shall have the capability to interoperate where it has been determined that processes need to communicate with each other.

Feeding model information forward and backward in life-cycle activities enables value-added iteration of enterprise processes that improves product quality.

3.3.3 Recursion

Activities W, H, and D are recursive and, thus, decomposable. Therefore, each activity can be divided into subactivities, and these subactivities will consist of another set of W, H, and D activities (see Figure 2).

These subactivities may be represented by different types of models. These models shall be able to interoperate where it has been determined that these subactivities need to communicate with each other.

EXAMPLE — In a manufacturing enterprise, the activity "Produce" can be, in turn, separated into lower-level activities W, H, and D. Activity W is user-needs driven and comprises any activities finally resulting in a request for what is to be produced.

Activity H is technology-requirements driven and comprises any activities finally resulting in how the product/system has to be produced in terms of a release statement. Activity D is task driven and comprises any activities finally resulting in the shipment of the product.

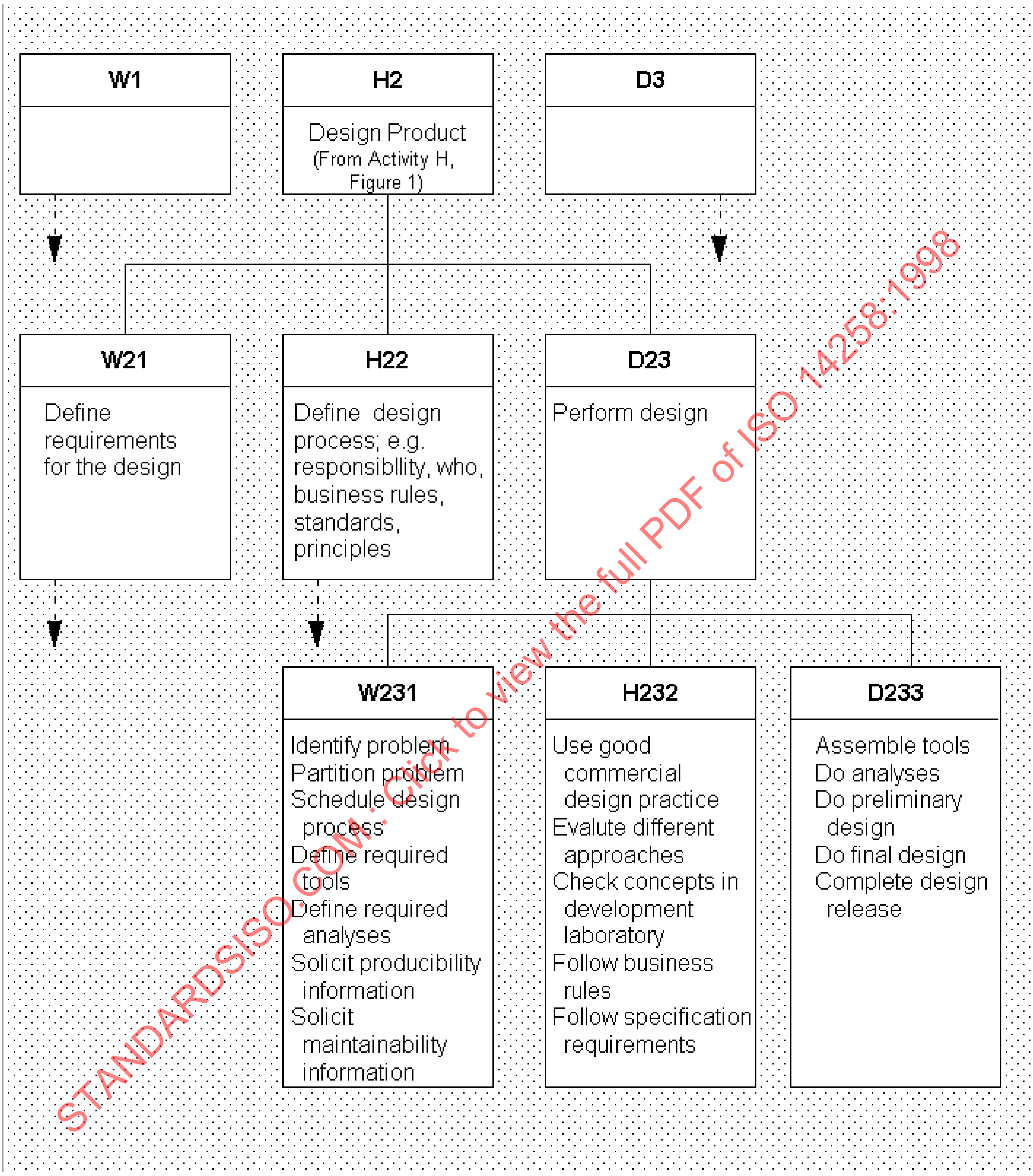
W1
H2
Design Product
(From Activity H, Figure 1)
D3
W21
Define requirements for the design
H22
Define design process; e.g. responsibllity, who, business rules, standards, principles
D23
Perform design
W231
Identify problem
Partition problem
Schedule design process
Define required tools
Define required analyses
Solicit producibility information
Solicit maintainability information
H232
Use good commercial design practice
Evalute different approaches
Check concepts in development laboratory
Follow business rules
Follow specification requirements
D233
Assemble tools
Do analyses
Do preliminary design
Do final design
Complete design release

Figure 2 — Decompose “Design Product” activity to show recursiveness of activities W, H, and D

3.3.4 Iteration

Activities W, H and D are iterative; therefore, there is no fixed sequence of these activities, but it is possible to return to previous activities to repeat them with updated input (see Figure 3).

Each performance of each activity may result in a different model. Every one of these different models shall be subject to both change and version management.

3.3.5 Naming

The names presented in the system life-cycle phases column of Figure 1 are merely example names. The particular product, process, project, or enterprise will have its own names for these phases to describe more appropriately what the life-cycle phase is, and the enterprise model should reflect the given names.

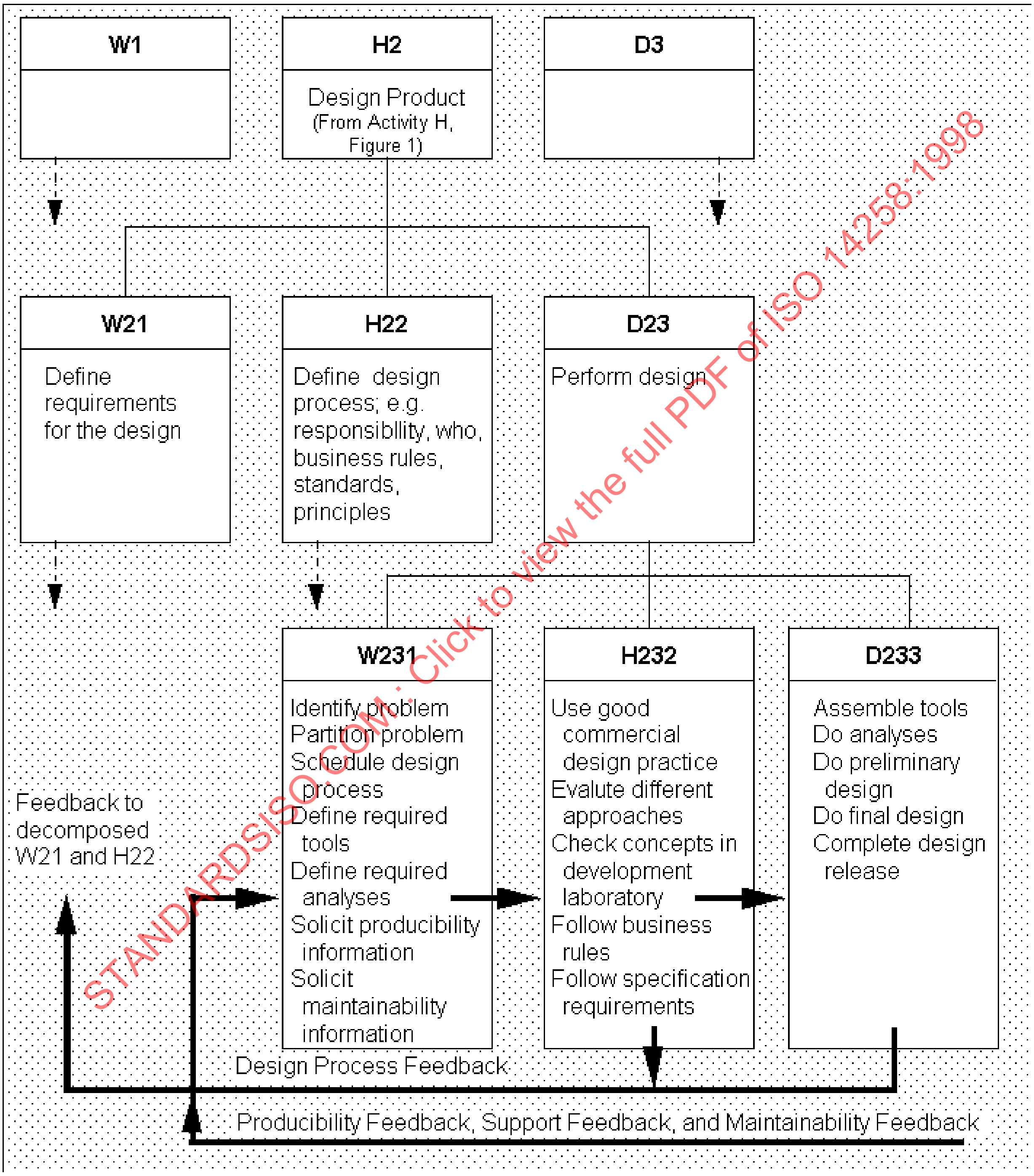
W1
H2
Design Product (From Activity H, Figure 1)
D3
W21
Define requirements for the design
H22
Define design process; e.g. responsibllity, who, business rules, standards, principles
D23
Perform design
W231
Identify problem
Partition problem
Schedule design process
Define required tools
Define required analyses
Solicit producibility information
Solicit maintainability information
H232
Use good commercial design practice
Evalute different approaches
Check concepts in development laboratory
Follow business rules
Follow specification requirements
D233
Assemble tools
Do analyses
Do preliminary design
Do final design
Complete design release
Feedback to decomposed W21 and H22
Design Process Feedback
Producibility Feedback, Support Feedback, and Maintainability Feedback

Figure 3 — Iterating the "Design Product" activity to show feedback used for process improvement

3.4 Hierarchy

3.4.1 Concepts of hierarchy (informative)

Hierarchy is a principle by which real world items and abstractions can be ranked and ordered.

There are two kinds of hierarchies: part-of hierarchies and kind-of hierarchies. Part-of hierarchies represent the composition of elements or the decomposition of systems. Kind-of hierarchies represent levels of abstraction that are ordered by generalization and specialization.

3.4.2 Usages of hierarchy

Kind-of hierarchies shall be used within models to define building blocks for entities to be modeled. Part-of hierachies shall be used to link models of different scope and detailing level.

3.5 Structure

3.5.1 Concepts of structure (informative)

The structural aspect is based on the principle that elements are not isolated but have multiple interdependencies with other elements of the system. The interdependencies are the explanation why the system exhibits properties different from the properties of its elements.

Graphics are often used in the representation of structures. Examples are trees, nets, stars, and loops. The elements are represented as nodes, the relations are represented as edges. Types of edges are undirected, unidirectional, and bidirectional edges.

Structure carries both a formal and a semantic aspect. The formal aspect refers to the layout of the graph and to the kind of edges. The semantic aspect is based on mappings of elements and relations to enterprise related notions. These mappings can be built very differently depending on the purpose of the mapping as well as on the boundary and the environment of the enterprise under consideration.

Enterprises perform activities on objects by using other objects. Examples of objects are products, resources, and orders. Examples of the activities performed on objects are generate, move, store, modify, and delete.

There are two structuring approaches commonly used for the mapping of elements and relations to enterprise related notions. The first approach is one in which activities correspond to elements and objects correspond to relations. An example of the first approach is a value-adding process where the output objects (considered as relations) of one activity (considered as an element) are the input objects of another action (considered as an element). The second approach is one in which activities correspond to relations and objects correspond to elements. An example of the second approach is the structure of a process plan where two objects (considered as elements) are linked by an activity (considered as a relation).

3.5.2 Compatibility of structuring approaches

The type of structuring shall be unambiguous to whatever facility is interpreting the models, either human or machine. The enterprise modeler shall ensure that models obtained by the two structuring approaches described in 3.5.1 are able to interoperate.

3.6 Behavior

3.6.1 Concepts of behavior (informative)

An enterprise is a social hybrid system, determined by properties of humans and machines. Humans (modeled as objects or resources) in the enterprise have a different behavior (e.g. learning and problem solving) from machines (e.g. acting and reacting) and therefore need a different kind of information.

Enterprises are dynamic and undergo continuous changes due to factors such as changing market conditions, technology, and knowledge. In recent years there has been a shift in how the operations of an enterprise are viewed. Rather than viewing the enterprise as being hierarchical in both structure and control, a distributed view has evolved where enterprise units communicate and cooperate in both problem solving and action. In turn, this requires far greater integration of functions within the enterprise and between enterprises than has been achieved in the past.

3.6.1.1 Time representation

If an individual element of the enterprise system has to be traced then properties of time need to be modeled to describe short-term changes. If the property time is introduced in terms of duration, it provides the base to do further analyses (e.g. process time). There are two kinds of behavior description relative to time: static and dynamic.

3.6.1.2 Static representation

Static representation of behavior is the description of the relations between variables of the system. For example, a business process is a logical sequence of relations between enterprise elements. For this static description it is not necessary to model the property time because it is the potentially allowed sequence of relations. Time related information (e.g. duration, concurrency) are missing.

3.6.1.3 Dynamic representation

Dynamic representation of behavior provides information about dynamic properties of elements, events and concurrency as well as about the time dependence of elements, attributes and relations. If the behavior description should be used to simulate the enterprise, the model must be populated with, for example, starting conditions, capacity, and workload.

3.6.1.4 Short-term and long-term behavioral change

An observer can categorize the change in behavior of a system as either short term or long term. There is no universally recognized distinction between short term and long term, but it is useful to make the distinction at the time of analysis. Criteria used in making the distinction include time duration of events, and rate of changes in behavior.

EXAMPLE — A drilling machine with a certain drill bit can illustrate behavioral change concepts. The behavior of the drill bit is of interest to the production-control system as it affects availability of the drilling machine for production purposes. The production-control system categorizes changes in the behavior of the drill bit as either immediate (short term) or continual (long term). An immediate behavioral change occurs when the drill bit breaks, resulting in a disruption in production until it is replaced with a new bit. A continual behavioral change occurs as the drill bit wears normally. Under normal conditions, the production-control system would plan for a drill bit change at a prescribed time or when the product approaches the acceptable minimum. Thus, the distinction made between short-term and long-term change allows the production-control system to specify different behaviors to replace the drill bit.

3.6.1.5 Sequentiality

Sequentiality is the necessary basis to describe behavior. Sequential cycles can be considered as similar states being traversed at different times. Measuring sequences in terms of time enables discrimination between similar cycles that progress at different rates.

3.6.2 Representation of behavior

Enterprise models shall have the capability to describe behavior; that is, to represent sequentiality (time), events, actions, condition, states, state changes, start states, end states, sequencing relationship between actions, and description of transformation functions.

Properties of sequentiality shall be modeled to describe short-term changes in the event that an individual element must be traced. Enterprise models used to analyze enterprise performance or to simulate certain processes shall have the capability to represent effects of sequential phenomena and the time duration of each sequence step. Enterprise models shall be capable of representing time duration, dynamic performance of processes, and sequential phenomena after specific units of time.

3.7 Relating the real world to enterprise models through views

3.7.1 Purposes of models (informative)

Models are descriptions of essential and relevant parts of the area of concern. They do not duplicate reality, but are limited approximations of the subset of reality under consideration. The appropriate level of detail for a model is indicated by its intended use, i.e., the purpose of the model. A full description of any model includes statements of its purpose, assumptions, and constraints.

Models can have very different purposes. They can be used to structure an area of concern to:

- illustrate or clarify thinking about an area;
- define structure, logic, and behavior of a system;
- support the problem-solving process by analyzing different options or solutions;
- help design, build, or operate a system.

3.7.2 Real world (informative)

The real world, in the context of this International Standard, is restricted to the enterprise which includes people, capital, equipment, processes, policies, and external links to its environment. The term enterprise represents any group of processes that one wishes to analyze. The portion of the real world which is selected by the modeler for a particular purpose can be either a whole enterprise or any subset of an enterprise.

3.7.3 Observers (informative)

An observer perceives the real world and analyzes it by attributing a certain meaning to what is observed. Observations pass through an observer’s mental filter which has been formed, and is continually modified, by experience, personality, politics, society, and the situation.

An enterprise modeler is an observer whose purpose is to create an enterprise model. The modeler shall define the purpose for the model.

A user of a model is also an observer. His perception is determined by the necessity to do something within the area of concern using the model as an aid to understand the area of concern in order to accomplish his task. For example, an implementer may use a model to improve enterprise operations.

There are many possible uses for a model. The user may wish only to work with a portion of the real world; that is, the user defines which attributes of the real world are to be included within the model. The user is free to define and select whichever portion of an enterprise is required to suit his purpose.

3.7.4 Views

Two things are important in order to understand a real-world system: the structure and the behavior of the system. The information view and the function view are of primary importance in representing the structure and behavior of a real-world system. This International Standard does not define the following:

- how the information of these views has to be structured in the model (only the existence of these representations is stated);
- what these views should include or not include;
- if other basic views (such as resource view or process capability view) are needed.

This International Standard states only that there is a minimum set of modeler views needed to present sufficient material to ensure the completeness, consistency, and integratability of enterprise models.

3.7.4.1 Information view

The information view is an orderly structured compilation, description, and representation of system information. This information describes the system elements, their structure, relationships, the information to be supplied for function execution, and information describing function results.

3.7.4.2 Function view

The function view is a description and representation of activities and processes in the enterprise. The function view describes the processing of elements and the concatenation of single processing steps to process chains, reflecting their logical connection and interdependence. The function view emphasizes the representation of system behavior, mutual dependencies, and influences of elements during function execution in the enterprise.

3.7.5 Rules for model views

A full, integratable description of any model shall include statements and descriptions of its purpose and constraints (including modeler assumptions). This shall be done by including a minimum set of modeler views that ensure adequate completeness and consistency, and provide the potential for integrating multiple models of a same enterprise. The number and the type of modeler views to be included in that set depend on the methodology used and the purpose and level of the models.

Views that are considered necessary are those that present a useful combination of activities, information, control, resources, and process capabilities.

3.8 Requirements for standards on model interoperability

3.8.1 Concepts of model integration (informative)

An enterprise model consists of structured data and rules about the information that applications use to do their work. An enterprise is large and complex, meaning that many models for many purposes are required to describe what is happening at any one time. Because of the diversity of the applications in an enterprise, the information contained in the models is encoded in many formats depending upon the needs of information generators and information users. Models can be in forms such as organigrams, spreadsheets, engineering drawings, process-flow data, and computer-aided engineering, computer-aided design, and computer-aided manufacturing files. These are the so-called islands of information or automation.

What users want from these models, however, is portability. They want to be able to reuse models across applications and not be dependent on specific application and tool configuration. The solution appears to be unified or integrated models that would provide a single information template for any application. Users want to implement technology that will improve their capability to provide for information from currently differing models to be usable by applications and platforms enterprise wide. It seems that a set of standard models would help promote this. The problem to accomplish this is that there are as many different ways of presenting a model as there are reasons for wanting to model the domain of interest.

A namespace is a binding space for generically interfacing and managing transactions that covers all platforms, applications, and models. The namespace requires a definition about the manner in which models are interrelated and the extent to which their contents are predetermined; for example if the grammar of the model is predetermined and constrained, or if it is to be taken as it exists or is found in the enterprise(s).

3.8.2 Forms of interoperability (informative)

There are three ways that models can be related to each other: they can be integrated, unified, or federated.

With integrated models there is a standard model form. Diverse models are interpreted against the common template. Standard or reference models must be as rich as the constituent models. All models can be stored in standard form with information filtered or translated by the applications, as in the case of IRDS[2]. Alternatively, standard models can be agreed to by constituent model owners, as in STEP[3]. There are enormous difficulties associated with standardizing large numbers of models. Therefore, the ability to deal with something less than integrated models is most certainly necessary.

The unified approach assumes that there exists a template which provides a common meta-level structure across constituent models, providing a means for establishing semantic equivalence. This template is then the basis for a

[2] Information resource dictionary system, described in ISO/IEC 10027 et al.

[3] Product data representation and exchange, described in ISO 10303.

metamodel. The metamodel is not in an executable form as it is in integrated situations. Using the metamodel, any model can be translated into any other. Loss of some semantics is possible. Normalized semantics is established by owners of constituent models. SUMM[4] is an example of a unified-model template.

The federated model scenario exists if one assumes that no agent successfully or globally can impose requirements for semantic equivalence across all models of an enterprise. Models must be taken as encountered. The template is at the meta level and, as in the unified situation, the template is not executable. Some degree of integration or unification would help the communication process. Interoperability requires that models be dynamically accommodated rather than having a predetermined metamodel. This would be furthered with some sort of predetermined terminology system. Definitions are a particularly difficult problem for process interoperability when the namespace contains federated models.

In reality, many applications will be dealing with models as encountered, or in the form in use at the time the model was created. This will be more common in interenterprise communications. This is often referred to as legacy information. For that reason, in a standard that states rules for enterprise models, where model interoperability is important, the assumption is that the federated situation exists and that the rules presented in the standard shall be rich enough to accommodate the encountered models, whatever the state.

Interoperability requires processes to communicate. For example, a querying process would address the responding process. The responding process would present its standard-model elements, which include its capabilities to communicate, to the querying process whose abilities to communicate are also in standard form. There would be a mediation to find the best fit from among the set of capabilities.

The degree of successful communication then will depend on the skills of the humans observing the process and the quality of the software employed to effect the communication.

Therefore, users must analyze and predetermine, if interoperability is to be accomplished, which processes must communicate and put in place procedures and tools so that the necessary communication can occur. The more that users wish their process to communicate with processes inside and outside the enterprise, the more pressure there will be to arrange the models in a standard way and to present their process capabilities on the medium on which interprocess communication is exchanged.

3.8.3 Need for standards to support interoperability

For the reasons described in 3.8.1, this International Standard asserts that

— the globally integrated models situation is not the subject of an International Standard because integration scenarios are created by business decision within a sector of a company for a particular purpose at a particular time using available technology;

— the federated situation is the most probable scenario of interoperability wherein most models will not be in a standardized or common form and it is not economically feasible to put them in such a form,

— the unified situation is a preferred path to achieve improved interoperability.

Both federated and unified approaches need standards to support their use. Depending on whether there is a federated or unified situation, the elements to be defined in a standard for enterprise models are different. To achieve unification, the constructs for models are the elements that shall be standardized. To achieve federation, the interfaces are the elements that shall be standardized.

For each chosen element to be standardized, the standard shall define

— the form(s) in which the element is represented,

— the relations between the element and other model elements,

— the capabilities of the element,

— the dynamics of the interaction involved.

[4] "Semantic Unified MetaModel," described in working drafts of ISO/IEC JTC 1/SC 21/WG 3.