ISO/IEC 18012-2

Edition 1.0 2012-07

INTERNATIONAL STANDARD

colour inside

Information technology – Home electronic system (HES) – Guidelines for product interoperability –
Part 2: Taxonomy and application interoperability model

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE T

ICS 35.200

ISBN 978-2-83220-192-3

CONTENTS

INFORMATION TECHNOLOGY –

HOME ELECTRONIC SYSTEM (HES) – GUIDELINES FOR PRODUCT INTEROPERABILITY –

Part 2: Taxonomy and application interoperability model

FOREWORD

1) ISO (International Organization for Standardization) and IEC (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. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. 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.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 18012-2 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 18012 series, under the general title *Home electronic system (HES) – Guidelines for product interoperability*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The widespread development of many national and regional home automation specifications, some standard and some proprietary, necessitates a mechanism for interoperability. Interoperability ensures that products from multiple manufacturers (potentially implemented using different automation systems) can interwork. It is desirable that devices needing to interwork do so seamlessly to provide users with a variety of integrated applications without modification of their protocols used within their specific system cluster. Examples of such applications include lighting control, environmental control, energy management, audio/video equipment control, and home security.

There are two fundamental methods to enable interoperability among applications developed for different communications protocols that use different application layers. These application layers may include different syntax and semantics for command, control, eventing and data.

- Method 1: Multiple gateways

  A communications gateway is intended to interconnect two different communications protocols. Therefore, to provide interoperability among three applications (A, B and C) that each use different protocols, gateways might be specified for:

  a) A ↔ B

  b) A ↔ C

  c) B ↔ C

- Method 2: A generic gateway

  Each application developer adds an interworking function (IWF) specified in this International Standard so that the application can communicate with other applications, regardless of the underlying communications protocol.

  d) A ↔ IWF

  e) B ↔ IWF

  f) C ↔ IWF

Interoperability is achieved via the IWF. For example, for application A and C to communicate: A ↔ IWF ↔ C. The IWF is a software-based generic gateway. This is a much less complex solution than Method 1. Application developers seeking interoperability using Method 1 develop translators (gateways) to each target applications. Developers using Method 2 implement only one IWF translator.

This International Standard provides a common classification and descriptive mechanism so that there is a common way of describing applications in any individual system, and an unambiguous mapping to key implementation items (e.g., data type primitives) to allow for transparent interoperability. Application-level interoperability cannot be achieved without being able to describe applications in a common form. The term “product interoperability” should be considered synonymous with application-level interoperability, since products are developed to implement and/or participate in (distributed) applications. The value of products to the end user derives from the applications which they support or provide.

The taxonomy specified here is based on application domain classification criteria for applications in home systems, as well as a lexicon of objects, events, properties, and primitive actions to effect or otherwise propagate change in the objects (their properties). This International Standard enables the specification and implementation of distributed application functions and services within the context of home electronic systems.

Work on this International Standard began with an in-depth review of the following existing systems, to understand the various application, interaction, and implementation models in use: ISO/IEC 14543-3-x *[network based control of HES Class 1)*, ISO/IEC 14543-4-x *[network enhanced control devices of HES Class 1],* ISO/IEC 29341 *[UPnP Device Architecture UPnP)],* ANSI/CEA-600 and ANSI/CEA-709 (also known as EN-14908). From that analysis, key similarities were identified among the various approaches and implementations. Those similarities are primarily in the high-level application functions that are being implemented, with differences appearing in the details of how the functions are represented. In short, there is a great deal of semantic similarity among various automation system application functions, but significant differences at the syntactic level.

That observation is the premise for the approach used in this International Standard. In order to facilitate interoperability, it is necessary to define an application interoperability model with the following characteristics.

- It allows a rich set of application functions, properties and interactions amongst distributed components contributing to the application to be clearly described in a common format.
- It incorporates a simple but flexible interaction model abstraction to represent all interaction models adopted by various system implementations (command/response, shared variable, message-oriented, etc.).
- It establishes the minimal set of common data type primitives to support unambiguous mapping of logical application data descriptions into implementation-specific binary representations within the interoperability domain.

Interoperability in distributed application systems is defined as the ability of two or more distributed components to communicate and to co-operate in predictable ways despite differences in implementation language, execution environment, or model abstractions. Three main levels of interoperability between components in a distributed application can be distinguished, as follows:

- protocol level, where the order of message exchanges and the constraints placed on either participant in the exchange (e.g. synchronous or sequential communications), are defined, alongside the resulting behaviour and possible blocking conditions. Interoperability at the protocol level provides the foundation to support the syntactic layer;
- syntactic level, where the names, interfaces and operation of the components are defined. Interoperation at this level is a necessary condition to support interoperability at the semanctic level;
- semantic level, where the meaning of the possible interactions between the components in the distributed application system is defined, in the sense of a defined/desired effect or output being generated.

Assuming that interoperability exists at protocol and syntactic levels, semantic level interoperability clashes between two application objects belonging to two different specifications but installed in the same premises and expected to co-operate are caused by differences in the HES-lexicon (conceptual schemas) that describe the components. Simply put, they may use different units for physical variable values and different names for objects, their properties and their functions, albeit they may be addressing the same application. Therefore, a mistranslation may occur between the two systems because of incomplete shared information. The possible clashes can be classified into two main groups, described below.

- *Lossless clashes* are those that can be resolved with no loss of information. Some examples in this category include component *naming clashes*, where the same component/information is represented by different labels; *structural clashes*, where information elements are grouped in different ways in different systems and *unit clashes*, where some scalar value (e.g. distance or temperature) is represented with different units of measure.
- *Lossy clashes*, which include interoperability clashes for which any transformation available, in either direction, causes loss in the information being communicated between the two application objects. These clashes are associated with differing levels of granularity, refinement or precision of the representation of the information. Note that a lossy translation between the two application objects may be an acceptable solution, provided that it achieves the desired application behaviour and does not affect the functional safety of the application or the system as a whole. One example of a lossy clash would be a light controller with a dimmer function. This controls a light actuator

(lamp); the light actuator understands only three levels of dimming, whilst the light controller supports up to 8 different levels. Any interoperability mapping between these two devices would require the mapping of the three levels recognised by the light installation to three out of the eight levels of dimming supported by the light controller.

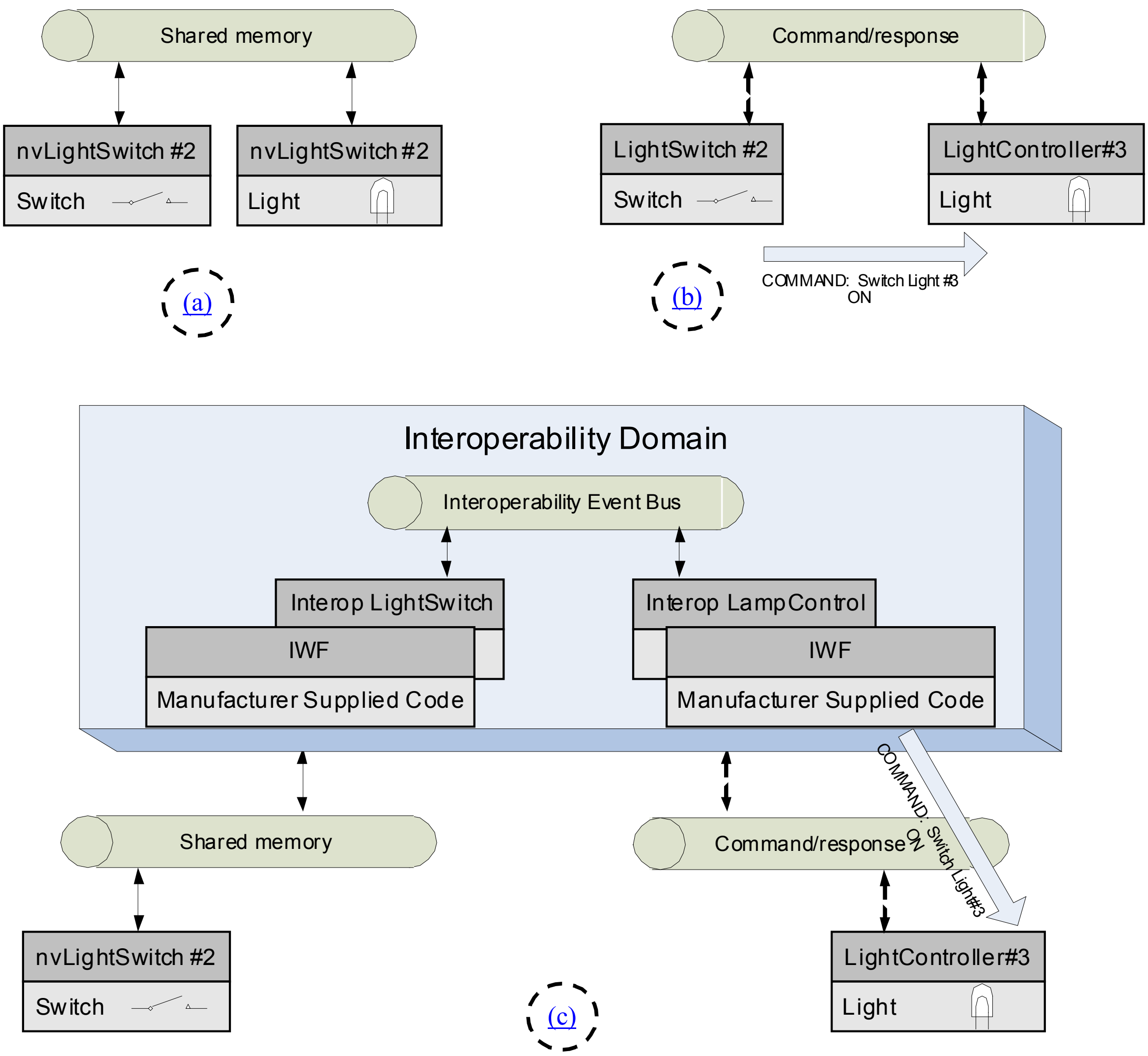

Figure 1 – Lighting application in (a) a shared memory system, (b) a command/response system and (c) an interoperating system

For example, one automation system might implement a shared variable space for communication between devices. In a simple lighting control example shown in Figure 1 (a), a user might turn a wall switch on, causing a shared variable in the switch device connected to the home system network to change from “0” (off) to “1” (on). A lighting controller component in the system might be subscribed to that shared variable, causing the automation system to notify the lighting controller of the change in the variable’s value (state). The controller could then take actions as defined by the configuration and programming of the lighting control application (in this case, switch the connected light on or off as requested).

In a complementary example, based on a command and control-based automation system, the wall switch might cause an “ON” command to be sent across the network in a message to the lighting controller component, which would then react appropriately (as per the description above).

In both examples, the behaviour of the lighting control application is the same, but the method used to implement it was quite different. This is an example of two systems having the same semantics (meaning and behaviour), but different syntax (implementation and codification or form).

This International Standard is based on the separation of the concept of action primitives from their actual implementation. An action primitive is a basic application action or device function that cannot be executed in part; it is either executed completely or not at all. A distributed application, or home electronic system aggregate function, is then thought as being provided through the execution of a sequence of such action primitives step by step, across the system. An example of an action primitive will be “Set temperature to 21 °C in Thermostat#21” (*Thermostat#21* is the name of an object.). To clarify the separation between action primitives and their implementation, let consider an example HES System-A, where devices/objects use <get>/<put> functions to implement local or remote reading and writing of variable values. This will not constitute “action primitives” in the context of this International Standard, but rather a programming mechanism used to invoke the application action primitives. These application actions are caused by <put>-ing (setting) the same variable to different values, which means that it is the variable and the value that it contains at a given time that define the application action, not specifically how the value is set (which, in this example, is through the use of a “PUT” message in System-A. This is captured later on in this International Standard by the introduction of eventing; objects (devices) notify each other with the values of their parameters, and each device (object) makes its decisions and takes actions (which contribute/constitute application actions) based on these values. During this processing each device may change these values; changes should be notified to all the interested parties. These *same actions* can be invoked in System-B by performing a (different) remote or local function call (i.e. not using <PUT>, but some form of a remote procedure call interaction). In this case it is possible for both systems to implement the same lexicon, and have the same application actions, but maintain (i.e. do not need to change) their own interaction mechanism and the corresponding protocols and syntax.

Typically, HES use different interaction modes, such as (distributed) shared memory/variable mode, command/response mode, remote procedure call mode, publish/subscribe mode (eventing) and variations of these. Using each of these gives the resulting home system certain characteristics, such as the ability to acknowledge correct execution of a remote operation (such as update of a state variable value, or the activation of a particular control). How to support a different set of operations using a common interaction model is well known in distributed control system design and implementation. However, it is beyond the scope of this International Standard to provide a detailed proof of the equivalence of such methods when translating between different interaction models, as shown in Figure 1.

The interoperable system is generic, and as such it should cater to different interactivity models (as described in the example above). It is assumed that any adaptation necessary from a system-specific interaction model to the eventing interaction adopted by the interoperability model is included in the implementation of the interworking functions provided by the manufacturers (or third party providers) interfacing *into* the interoperability model.

This interoperability standard addresses interoperability of products manufactured by manufacturers. These products can be system components in the context of a thermostat being a component of a heating system, or stand-alone as in a “device” that can collect information from such a thermostat device to be used in an application not related to heating control. The interoperability model in this International Standard addresses the requirements of two developer communities: the component developers (e.g., manufacturers), who develop individual devices and systems, and the solution developers (e.g., integrators or field installers), who provide one (or more) applications that use services provided by these components. The component developers (and manufacturers) will have a clear objective to provide a mapping onto to ensure that the services provided by their components are used outside the original single-specification system design. The solution developers will be able to choose and use from a wider range of components or even create new applications using services from components that are already installed and configured, provided they conform to this International Standard. The two communities overlap at least partially. For example, a company that develops automation devices would typically provide software routines that link their products to specific application objects. Such objects could be an application model the company defined and published or could be a set of standardised objects published as part of a standardised application interoperability model (AIM). In the former case, the object definitions would be considered to be standardised only upon their publication in an AIM registry of interoperable objects.

This International Standard comprises the following clauses:

- Clauses 1 through 4 are the scope, normative references, terms, definitions and abbreviations, and conformance clauses respectively;
- Clause 5 describes the application interoperability model;
- Clause 6 describes the interaction model in terms of an asynchronous event-bus model;
- Clause 7 describes the taxonomy used for inter-system and intra-system interoperability;
- Clause 8 describes the object schema framework;
- Clause 9 describes the application binding map schema framework;
- Annex A describes an example of an interoperable application specification;
- Annex B contains the base object schemas;
- Annex C contains examples of base object schema extensions;
- Annex D contains miscellaneous notes on interoperability and related taxonomy terms.

INFORMATION TECHNOLOGY –

HOME ELECTRONIC SYSTEM (HES) – GUIDELINES FOR PRODUCT INTEROPERABILITY –

Part 2: Taxonomy and application interoperability model

1 Scope

This part of ISO/IEC 18012 specifies a taxonomy and application interoperability model for the interoperability of products in the area of home systems. It also specifies an interoperability framework to allow products from multiple manufacturers to work together in order to provide a specific application. This standard describes an application process that exists above the OSI reference model (ISO/IEC 7498-1) stack, with sufficient detail needed to establish interoperable applications in this domain.

This International Standard is applicable to

- single implementation home electronic system networks, connected devices and applications. Such a system is created when all the home system components comply with a single standard or manufacturer specification,
- multiple implementation home electronic system networks, connected devices and applications. Such a system is created when different home system components comply with different HES standards or manufacturer specifications,
- automatically configured devices,
- manually configured devices,
- manually configured groups/clusters of devices.

This International Standard applies to application objects in operation within networks, between networks and to components located at the junction of dissimilar networks. Two (or more) dissimilar networks that conform to this International Standard, when linked by some communication system, are expected to behave as if both networks were logically the same network from an application perspective.

Interoperability considerations regarding general management processes in HES are described in ISO/IEC 18012-1. This part of ISO/IEC 18012 addresses only the management aspects related to the operation mode of interoperable HESs and does not cover the management processes of individual constituent networks.

2 Normative references

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

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 18012-1, *Information technology – Home electronic system (HES) – Guidelines for product interoperability – Part 1: Introduction*

IEC 60050-714:1992, *International Electrotechnical Vocabulary – Chapter 714: Switching and signalling in telecommunications*

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*