PD ISO/TR 12310:2015

BSI Standards Publication

Health informatics — Principles and guidelines for the measurement of conformance in the implementation of terminological systems

bsi.

...making excellence a habit.™

National foreword

This Published Document is the UK implementation of ISO/TR 12310:2015.

The UK participation in its preparation was entrusted to Technical Committee IST/35, Health informatics.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

ISBN 978 0 580 83759 3

ICS 35.240.80

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 May 2015.

Amendments issued since publication

Date	Text affected

PD ISO/TR 12310:2015

TECHNICAL REPORT

ISO/TR 12310

First edition
2015-05-15

Health informatics — Principles and guidelines for the measurement of conformance in the implementation of terminological systems

Informatique de santé — Principes et lignes directrices pour le mesurage de la conformité dans la mise en oeuvre des systèmes terminologiques

Reference number
ISO/TR 12310:2015(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

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

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information.

The committee responsible for this document is ISO/TC 215, *Health informatics*.

Introduction

This work item is a Technical Report that will identify and discuss principles and guidelines for the measurement of conformance in the implementation of terminological systems, in particular, as applied to Electronic Health Record (EHR) systems.

This item will leverage the current work under way in Canada and will be developed in liaison with International Health Terminology Standards Development Organization (IHTSDO) and the Vocabulary Committee of HL7 in the spirit of harmonization across organizations with similar interests. Additional terminology organizations, active projects and existing expertise will be sought out for input into this work item.

Conformance is a key step in helping stakeholders determine if implementations of terminology systems have been done in a correct and consistent manner, particularly as implemented in EHRs. Loose declarations regarding terminological systems that cannot be tested with meaningful results do very little to support the end goal of the interoperable EHR. Therefore, the principles and guidelines for establishing and measuring conformance will focus on identifying the degrees of conformance of terminological systems with or without use in messaging standards.

This Technical Report is intended to define what is meant by conformance with respect to terminology systems, particularly as applied to EHR systems, and it is expected to facilitate the formulation of policies and governance practices locally or nationally. This Technical Report is timely as the emerging IHTSDO and progressive implementation of the EHR will lead to the increasing awareness of conformance with respect to terminologies and consistent implementations that allow interoperability by all end-users.

The focus of this Technical Report is to define best practices and a framework for establishing and measuring conformance. The scope of this Technical Report will include the identification of definitions and best practice considerations for what constitutes conformance to terminology systems and the principles for which conformance can be demonstrated.

Health informatics — Principles and guidelines for the measurement of conformance in the implementation of terminological systems

1 Scope

The purpose of this Technical Report is to define a framework of good practices for terminology system maintenance and the principles for which conformance can be demonstrated. The primary focus is the application of terminology system to Electronic Health Record (EHR) systems, although the principles and guidelines can be applied broadly in health informatics

The scope of this Technical Report will include, at a minimum, the following considerations for keeping terminology systems and associated reference material clinically and/or technically relevant and valid:

— governance models and practices;

— high level processes;

— requirements for managing the change.

The scope of this Technical Report will not include a definition of the detailed processes for performing terminology maintenance.

This Technical Report aims to define the framework of good practices for EHRs and systems regarding terminology maintenance within these systems. This Technical Report relates directly to the ability of these records to be safe and legally accurate records of healthcare in the environment of changing technologies related to the use of clinical terminologies to represent meaning within these systems.

2 Objective

This Technical Report identifies considerations for the expression and evaluation of conformance for solutions that make use of terminology. The specific focus of this Technical Report is terminology used in healthcare solutions. However, the principles should apply to solutions implementing terminology across the health industry. “Solutions” is interpreted broadly and includes both software and hardware technical implementations, as well as other specifications that are based on or claim to adhere to all or part of the specification against which conformance is being assessed. Implementation in this Technical Report does not consider procedural or governance requirements.

By using the definitions and recommendations found here-in, standards bodies, implementers, and other parties can better achieve their objectives in the development and use of specifications that make use of terminologies and can better express their terminology capabilities.

This Technical Report is intended to be independent of any particular terminology or terminological approach, though some portions of the guidance provided will only apply to certain types of terminologies.

3 Terms and definitions

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

NOTE Because “terminology” is such a broad term, conformance actually needs to be stated in terms of the various terminology components that are referenced in a specification. These components will also be defined.

3.1
conformance
adherence of a system or specification to the expectations set by another specification

Note 1 to entry: The general definition for conformance has changed over time and been refined for specific standards. In 1991, ISO/IEC 10641 defined conformance testing as "test to evaluate the adherence or non-adherence of a candidate implementation to a standard." ISO/IEC/TR 13233 defined conformance and conformity as "fulfillment by a product, process or service of all relevant specified conformance requirements." In recent years, the term conformity has gained international use and has generally replaced the term conformance in International Standards.

Note 2 to entry: In 1996 ISO/IEC Guide 2 defined the following three major terms used in this field:

— conformity - fulfillment of a product, process, or service of specified requirements;

— conformity assessment - any activity concerned with determining directly or indirectly that relevant requirements are fulfilled;

— conformity testing - conformity evaluation by means of testing.

3.2
code system
managed collection of *concept representations* (3.4) intended for use in persisting or sharing of information

3.3
concept
single mental representation of some real or abstract thing

Note 1 to entry: Concepts should be unique within a *code system* (3.2).

3.4
concept representation
mechanism by which the system can express a *concept* (3.3)

Note 1 to entry: Different representations can serve different purposes. Most *code systems* (3.2) support multiple representations for each *concept* (3.3), sometimes even multiple representations of a given type. In some cases, distinct representations of a *concept* (3.3) may have their own identifier assigned within the *code system* (3.2) for maintenance and internal reference purposes. The types of representations are *code* (3.4.1), *concept id* (3.4.2), and *concept designation* (3.4.3).

3.4.1
code
concept representation (3.4) intended for use when representing a *concept* (3.3) in a computable manner

EXAMPLE Passing into a decision support tool or for use in data exchange.

3.4.2
concept id
concept representation (3.4) that is unique within the *code system* (3.2) and that is used internally by the *code system* (3.2) when referencing *concepts* (3.3)

3.4.3
concept designation
human consumable representation of the *concept* (3.3)

Note 1 to entry: A concept designation may or may not be a string of characters (could be multimedia); generally subject to language variants.

3.4.3.1
concept name
concept designation (3.4.3) that is the unique designation of the *concept* (3.3) in the *code system* (3.2) and intended for human understanding

Note 1 to entry: This is usually text but might also be graphical for some *code systems* (3.2). For example, images of different facial expressions for a *code systems* (3.2) representing pain scales.

3.5
code system partition
result of dividing a code identifier namespace into constituent components in order to delegate responsibility among organizations

Note 1 to entry: Some *code systems* (3.2) divide their "code" identifier namespace and delegate responsibility for different sets of *codes* (3.4.1) to different organizations.

Note 2 to entry: Examples include SNOMED-CT and LOINC. Each delegated organization is then responsible for the development, maintenance, and publication of the content for its delegated namespace of *codes* (3.4.1). This delegation allows organizations to introduce needed *codes* (3.4.1) more quickly than would be possible with a centralized approval mechanism. A code system partition is the set of *codes* (3.4.1) maintained by a single organization in such a delegated scheme.

3.6
code system supplement
informative extension to a *code system* (3.2) involving non definitional information to support implementation

Note 1 to entry: When introducing the use of a *code system* (3.2), the representations, properties, and relationships of the *concepts* (3.3) within that *code system* (3.2) do not always meet the needs of the potential users of that *code system* (3.2). For example, they may require translations of display names or definitions to other languages or using terminology more familiar to their community. They may need additional properties indicating allowed use of the *concepts* (3.3), for example, "Which lab tests are orderable?" These users may choose to "supplement" the *code system* (3.2) with additional information so that it meets their requirements. Because no *codes* (3.4.1) or *concepts* (3.3) are added, interoperability based on the underlying *code system* (3.2) is still maintained. This set of independently published supplemental information for an existing *code system* (3.2) is known as a code system supplement.

3.7
local code system
code system (3.2) used only within the organization that maintains the *code system* (3.2) or in direct communication with that *code system* (3.2)

Note 1 to entry: These *code systems* (3.2) are useful for achieving consistency within an organization but do not achieve interoperability across organizations. Because they are maintained directly by the organization using the *codes* (3.4.1), their maintenance processes are normally very responsive in the addition of new *codes* (3.4.1); however, they are frequently not as robust in the following of good vocabulary processes such as avoiding *code* (3.4.1) re-use, avoiding overlap, etc.

3.8
terminology binding
assertion of what *codes* (3.4.1) are to be used at a particular place within a specification, including an indication of *conformance* (3.1) expectations

4 Purposes for conformance

For the evaluation of conformance to serve a useful purpose, there has to be some sort of benefit. There are a variety of benefits to seeking conformance with a specification that includes a terminology component. This Clause summarizes some of the most frequent objectives.

4.1 Interoperability

One of the most frequent objectives for the enforcement of terminology specifications is to aid interoperability between systems that use those specifications. Two systems that use different codes for the same concept or that do not understand the same set of codes are unlikely to interoperate safely.

4.2 Data analysis

Knowledge bases, decision support engines, clinical studies, and other forms of analysis usually require coded information to be captured in a consistent way. Verifying the conformance of a system or specification can help to confirm that the data collected using that system will be able to be analysed.

4.3 Consistency of user experience

When users in a particular community will be making use of multiple systems, there are significant benefits to learning curve and user acceptance if those systems capture data in a consistent way. This includes the terminologies used to capture data.

4.4 Application functionality

The terminology capabilities of a system or specification are one measure of the sophistication of that system or specification. A system that is capable of capturing detailed post-coordinated coded information is significantly more sophisticated than a system that is only capable of capturing free-text information. As a result, conformance with a terminology specification can be used as a quality measure independent of any particular expectation of use of the data.

4.5 Acceptance filter

Sometimes, an expectation of conformance may be used as a bar for acceptance in some sort of program. This might be for a regulatory purpose, as part of a procurement process, or for other reasons. The expectations might be driven by one or more of the preceding rationales, but it can also just act as a "filter" to reduce the number of qualifying systems.

5 Conformance process

Once a determination has been made that conformance to a terminology specification will be useful, the following are four steps to the conformance process.

a) Ensure the specification that conformance is being evaluated against (base specification) clearly documents terminology expectations.

b) Document the capabilities of the system or specification being evaluated for conformance in terms of the base specification.

c) Evaluate the documented capabilities against the expectations set by the base specification.

d) If dealing with a system, test the system to verify that the documented capabilities reflect the actual capabilities.

These steps are discussed in detail in the following subclauses.

5.1 Documenting expectations

Documenting expectations is the most critical step in any terminology conformance exercise. If the base specification is unclear about requirements, then any evaluation of conformance will be uncertain and the desired objectives of achieving conformance will not be met. This subclause will discuss areas a terminology specification should cover in order to be clear and complete.

5.1.1 Optionality

Many specifications allow a degree of flexibility in exactly how terminology (and other aspects of the specification) needs to be implemented in order to be conformant. This may be due to variations in the environments in which the specification is being implemented, recognition of reasonable diversity of implementer capability requirements or needs, or other reasons. It is therefore essential that terminology specifications clearly differentiate which options are required and which are not. Recommended terms and definitions are listed below.

5.1.1.1 Optional

Optional requirements are those that may be supported but need not be supported by conformant systems. The expectation is that any conformant system that does support the requirement will support the requirement in the manner specified, i.e. the mechanism of implementation is not optional, but the choice of whether to implement or not is optional. Terminology specifications identifying a requirement as optional will often use the term "MAY".

5.1.1.2 Recommended

A "recommended" requirement is an optional requirement where the author of the terminology specification wishes to assert a best practice. While systems that do not implement a recommended requirement would technically be conformant, they might still be considered somewhat deficient or "non-optimal". Terminology specifications identifying a requirement as recommended will often use the term "SHOULD".

5.1.1.3 Required

A "required" portion of a terminology specification shall be implemented by all implementations that wish to be conformant with the base terminology specification. Failure to implement a required component automatically renders the solution non-conformant. Terminology specifications identifying a component as required may use the term "MUST" or "SHALL".

5.1.1.4 Not recommended

In some cases, a terminology specification will identify what sort of content is not allowed rather than what is allowed. This can be stated with varying strengths. Non-recommended content is similar to optional content. The inclusion of the non-recommended content is not non-conformant, but may indicate that the implementation is "non-optimal". Terminology specifications identifying a requirement as not recommended will often use the term "SHOULD NOT".

5.1.1.5 Not permitted

This is the opposite of "required". Rather than being non-conformant if the content is not supported, a system would be non-conformant if it does support non-permitted requirement. Terminology specifications identifying content as not permitted will often use the term "SHALL NOT".

5.1.1.6 Conditional

Any of the above degrees of optionality may be conditional. This means that a given part of a terminology specification might be required, recommended, optional, not recommended, or not permitted based on some condition. The condition might be based on what other aspects of the terminology specification are supported, the environment in which the terminology specification are being implemented, or other factors. Care should be taken when marking components of a terminology specification as "conditional" that the condition is clearly stated and can be consistently evaluated to either true or false for a given implementation. As well, the terminology specification shall be clear as to the type of optionality that applies if the condition is true. For example, is the requirement "conditionally recommended" or "conditionally required"?

6 Terminology artefact conformance considerations

The following subclauses describe the various questions that should be answered when creating terminology specifications that reference different types of terminology artefacts. By ensuring the terminology specification provides clear answers to each of these questions, ambiguity is reduced enabling conformance to be more accurately assessed.

6.1 Code system considerations

This subclause covers the questions that should be addressed when implementing each code system referenced by a terminology specification, as well as by terminology specifications that reference code systems.

6.1.1 What is the code system being referenced?

Ensure that the code system being referred to is clearly identified. For example, "ICD" would not be a useful reference as there is multiple code systems with that label that have been published over time, does the author mean ICD9? ICD10? Even specifying ICD10 might not be sufficient as many jurisdictions have their own versions of the ICD code system that have been supplemented and organized differently. If the code system is a local code system, it is important that the responsible organization be clearly distinguished. The set of "lab order types" for one hospital (or even one department) might not be the same as they are for another, even if they are nearby or even in the same building.

In the event that a terminology specification supplements codes from some non-local code system with additional local codes, a clear distinction shall be made between the third party codes (which come from one code system) and the local codes (which come from a distinct code system). This is because updates and validations may be processed differently against the third party code system than they are against the local codes.

Reference to a unique identifier assigned by a registry of code systems can help avoid ambiguity when referring to code systems.

6.1.2 What version(s) of the code system are supported?

Code systems frequently change over time. New codes may be introduced. Older codes may be deprecated. In some code systems, codes may even be assigned to different code systems over time. Code systems that follow this practice are extremely risky to use and should be avoided, if possible. Other information within the code system may also change, such as concept designations, definitions, relationships, etc. Because of this variability in the content of code systems over time, knowing what version (or versions) of the code system are expected to be used makes a difference in the expected behaviour of a system. Clear descriptions of expected behaviour are essential to conformance, therefore identification of the version or versions of a code system that are supported is essential when defining terminology conformance.

Code system authors may define different mechanisms for identifying versions of the code system. Some code system authors may not recognize the concept of version at all. They simply change the contents of the code system as and when they need to or wish to. Even among those code systems that do assign version identifiers, the consistency with which versioning occurs may vary. For example, some code system authors may still apply changes within a version, making the maintainer-assigned version id insufficient, or at least sub-optimal for identifying exactly what set of code system content is being referred to.

Because of the non-reliability or even non-existence of code system author assigned version identifiers, a date stamp or even a timestamp may be the most effective mechanism for explicitly referring to a particular version of many code systems. If taking this approach, the terminology specification should be clear on exactly what the date means, is it the date the content is changed, the date the content is published, or the date the content is "intended for use"?

Where using a terminology maintenance organization-assigned version identifier, care should be taken to ensure consistency in the representation of the identifier with respect to the capitalization,

whitespace, punctuation, and other features that can occasionally interfere with matching of string-based identifiers.

6.1.3 How are codes represented?

Codes are only useful if they are captured consistently and can be recognized by systems searching and manipulating them. In some cases, the display and persistence of coded values can vary. Terminology specifications shall identify rules around how codes must be expressed for their various uses. Possible issues that can interfere with consistent representation include the following.

— Case sensitivity: Are code representations case-sensitive or case-insensitive? If case-insensitive, is there a convention of always using upper case or lower-case?

— Character set: Does the code system make use of special characters that are only available in limited or specialized character sets?

— Whitespace: Is whitespace significant in determining matches for codes? This can impact how codes are stored.

— Punctuation and special characters: Are the codes typically displayed with separator punctuation and, if so, is this punctuation significant in distinguishing between codes? Is there a convention that the punctuation must be present or absent when persisting or displaying to users?

— Leading zeroes: For "numeric" codes, are leading zeroes present on some coded values, and if so, are they significant? E.g. is "001" equivalent to "1"? Is there a convention for whether leading zeroes should be present or absent, and if present, how many digits should be present?

6.1.4 What are the version migration expectations, if any?

In the event that the terminology specification does not limit itself to the support of a single version or explicit set of versions, expectations need to be established for the timeliness of support for new versions once they are made available. For example, if the code system authority publishes a new version of a code system on a given day, is an implementation non-conformant if it fails to process codes appropriately based on that new version an hour later? What about three years later?

Failure to set expectations on the timeliness in which implementations of a terminology specification support new versions of the code systems referenced by the terminology specification will obviously affect consistency of implementation and make conformance evaluation challenging. This issue can be reduced by code systems that publish changes with "future-dated" effective dates. This allows the changes to be distributed, implemented, and tested prior to the change actually being enabled.

Even when expectations are clear, in the absence of a "future-dating" code system, implementation realities will often make a synchronized "instantaneous" transition of all implementations of a terminology specification unfeasible. Terminology specifications that deal with interoperability should therefore ensure that policies support interoperability even where the communicating systems do not receive the new version of a code system at the same time. For example,

— supporting the receipt of "deprecated" codes throughout the allowed migration window, and

— either tolerating the receipt of unrecognized codes or (even better) automatically checking for an update to the code system upon receipt of an unrecognized code.

6.1.5 What is the authoritative source that will be used for processing updates to a given terminology?

Some code systems have a single authoritative distribution system, be it a terminology service, website or other publication mechanism. However, many other code systems and updates to those code systems may be distributed from a variety of sources. For example, ISO code systems may be distributed by authorized national standards bodies. SNOMED CT updates may be distributed through their respective country affiliates, etc.

Not all of these various distribution mechanisms will necessarily be synchronous. That means that two different systems each looking at the most "current" version of a code system from their respective distribution source may be looking at different versions of a code system. In some cases, the distribution of code systems may even accidentally introduce discrepancies, meaning there could be unintentional variations between the same versions of a code system from different distribution sources.

To ensure maximum consistency between implementations, it is therefore recommended that terminology specifications referencing a code system also indicate the authoritative source for that code system from the perspective of conformance with that system.

6.1.6 What status must usable code system elements have?

Many code systems assign a "status" to elements (codes, representations, relationships, etc.) within the code system. This allows a code system to include content that is still in the design stages, content intended to be in active use and "old" content that should no longer be used but is published for historical purposes. Some code systems may also associate dates with the terminology element identifying a past or future effective date for a particular status. For example, "This code will become active on June 1, 2010".

Where a code system supports the concept of element status, a terminology specification should clearly indicate exactly what statuses are allowed to be used. A default assumption would be that only "active" concepts are permitted. However, in some cases, "pending" concepts or deprecated concepts might be allowed. Some terminology specifications might even place a time limit. For example, "concepts deprecated not more than one year prior to time of capture".

6.1.7 Which codes and concepts are usable for what purpose?

Some code systems include concepts and concept designations that are constrained for particular uses. For example, a code system may define "abstract" concepts that are useful for analysis and navigation, but are considered too high-level or general to be appropriate for data capture or exchange. Others may have "internal" concepts that help define the semantics of a code but are not intended to be used on their own. Examples might be the prefix "k" for "kilo" in a "units of measure" code system or the different axis used to express the semantics of an observation type in the LOINC code system.

Strict conformance with a code system would require that these limitations be respected. However, in some cases, a code system may be used outside the scope of its initially intended use and the limitations on what concepts can be selected and exchanged may not be appropriate to that new use. Terminology specifications should therefore declare if they intend to override the allowed uses for concepts within a terminology specification (and which concepts are overridden, if the override is not global). Of course, this is predicated on code systems making clear what the intended use of concepts and representations within that code system. In the absence of any guidance, codes in a code system should be deemed appropriate for any possible use.

6.1.8 What representations are allowed for what purposes?

Some code systems are extremely simple. They just include a flat list of codes. These codes serve to both uniquely identify a concept, to represent the concept during data capture, and to express the concept to users. However, other code systems will provide distinct representations for each of these uses, frequently referred to as "concept ids", "codes", and "descriptions". For some code systems, there may even be multiple choices among these. For example, "long" vs. "short" display names or "formal" vs. "informal" descriptions.

In situations where more than one representation might be used for a given purpose, terminology specifications referencing a code system should clarify which representation or representations they permit for the uses with which they are concerned.

6.1.9 What code system relationships must be understood and navigated?

Many code systems are simple flat enumerations with no relationships asserted between codes. However, other code systems may provide hierarchies showing concept specialization or containment. Sophisticated

code systems, such as SNOMED CT, include numerous relationships between concepts. These relationships might be used to define semantics, to guide decision support, and/or to aid in user navigation.

Depending on the intended use of a terminology specification, support for and the ability to navigate these different relationships may or may not be important. Terminology specifications should therefore make clear exactly what capabilities they require around the understanding of and ability to navigate different relationship types within a given code system or even a given reference set drawing from a code system.

6.1.10 Are the semantics clearly defines?

Use of code systems is only useful where concepts are represented appropriately within the code system. That means that the meaning ascribed by a system to a given code system representation is in alignment with the semantics ascribed to that representation. This, in turn, requires that the code system be clear in the semantics of a given representation. This might be done through descriptive language definitions and/or the positioning of the concept within a web of other relationships and properties.

Quality code systems should ensure that all concepts have unambiguous semantics such that it is possible to evaluate whether a given real-world concept is encompassed by a particular code system concept. If semantic definitions are ambiguous, then verifying conformance (or at least achieving the desired benefits of it) will be difficult due to multiple inconsistent interpretations of a concept.

6.1.11 What are the expectations for post-coordination?

Most code systems follow a methodology known as “pre-coordination”. This means that every representable concept has explicit representations assigned to it that can be used when capturing, exchanging, or otherwise manipulating that concept. However, a limited number of more sophisticated code systems may also support a methodology called “post-coordination”. In post-coordination, new concepts can be represented by taking the existing pre-coordinated codes and combining them using a grammar defined by the code system. For example, the Uniform Codes for Units of Measure (UCUM) code system does not define a pre-coordinated code for “mg/L”. However, it defines the code “m”, “g”, and “L” and defines a grammar for combining them into a representation for the “mg/L” concept.

Where post-coordination is supported, the syntax and rules for post-coordination syntax shall be clearly defined. In situations where the syntax allows variation, it should also be clear what variations are expected to be supported by a given terminology specification. For example, if the syntax allows expressing both codes and display names, are the display names required, prohibited, or optional? In some cases, there might be an expectation that post-coordination would not be supported at all in which case, that should also be clear in the terminology specification.

6.1.12 What partitions are included?

Most code systems are maintained by a single organization and published as a single set of codes. However, some code systems, including SNOMED-CT and LOINC, delegate the development and maintenance of codes in isolated namespaces to other groups, such as national affiliates or other large organizations. These namespaces or partitions are created in such a way as to avoid collisions between concept identifiers defined by different groups and allow for greater responsiveness in supporting the needs of local implementers.

NOTE The difference between a “partition” and a distinct, local, supplementary code system is whether a namespace (collection of unique identifiers) within the base code system is delegated to a single source responsible for managing the definition and semantics of the concepts with those identifiers. Note that in some environments, multiple layers of delegation can be possible, e.g. International standards body delegates a partition to a national body that then delegates subsets of that partition to regional groups, which may perform further delegation to individual projects.

Because each of these partitions is independently maintained and published, there is no single source from which an implementer can receive a “complete” copy of the code system. In addition, there is rarely an expectation that a single implementation would support concepts from all of the various regional partitions. When a terminology specification references a “partitioned” code system, it therefore needs to make clear exactly which partition(s) are expected to be supported. In addition, because partitions

can be versioned independently, all questions relating to the version of the code system also apply to versions of the partitions of a code system.

The use of partitions might even exacerbate versioning issues as code systems that support partitioning often have a process for migrating concepts initially proposed in delegated namespaces into the "main" code system. This can result in concept identifiers being deprecated in the partitions and replaced by concepts identified in the main code system, with the attendant migration and translation issues for implementers.

6.1.13 What code system supplements are supported?

Code systems are often used in languages and circumstances that are substantially different from the purpose for which they were initially created. The underlying structure and set of concepts defined by the code system are still appropriate for the re-use and re-using allows greater interoperability than would be achieved by starting from scratch. However, the published code system might be lacking certain essential pieces of information for use in the new context. Listed below are examples.

— The codes might require display names or definitions that have been translated to the language in the jurisdiction where the code system will be used.

— Relationships such as subsumption hierarchy that were only implicit in the original publication of the code system may need to be made explicit to support analysis or decision-support.

— Additional properties may need to be associated with concepts giving guidance on use within the context.

— Properties defined in supplements may be used to construct the definitions for reference sets. The additional concept designations may be used for exposing the concept to users. Therefore knowing what supplements may or shall be supported is important.

6.2 Reference sets: sets of codes that are allowed

The introduction of reference sets into a terminology specification means that there is an attempt to restrict the set of coded representations that are permitted for use in a particular area. This adds an additional set of questions that need to be clearly answered in the terminology specification.

6.2.1 What code systems are drawn from?

Reference sets draw codes from one or more code systems. Therefore, all questions related to code systems discussed in the previous subclause shall also be defined for each reference set specification. This also includes identifying the code system version(s), partition(s), and/or supplement(s) that are included in defining the allowed set of codes.

Note that the same specification may draw from the same code systems as part of multiple distinct reference sets and may have distinct expectations for versions and other information for the same code system for each reference set. For example, the reference set for diagnosis codes might be constrained to a single version of SNOMED CT, while the reference set for procedure codes might allow multiple versions.

6.2.2 How is the reference set defined?

Reference sets can be defined in two ways: extensionally (by enumeration) and intentionally (by means of an expression). Enumerated reference sets are straightforward when evaluating conformance. If the code is in the enumerated list, then it is allowed. If it is not in the enumerated list, then it is not allowed.

Matters are more complex for intentional reference sets. The expression used to define the allowed codes shall itself be unambiguous and should ideally be able to be algorithmically evaluated. Ambiguity can lead to conformance evaluation issues. For example, a reference set defined as "All SNOMED CT procedure codes related to diabetes" is extremely ambiguous. What is meant by "related"? What types

of diabetes? How close must the relationship be? Is it limited to clinical procedures or are administrative procedures included as well?

In some cases, a reference set may be built from other reference sets, including reference sets defined in other specifications possibly by other authors. All of the considerations for unambiguous definition shall therefore be evaluated against these "incorporated" reference sets as well.

6.2.3 Which representations are allowed from a code system?

When dealing with intentional reference sets, the expression will often identify which concepts are to be included or excluded. However, there may be multiple representations associated with the concept in some code systems. For example, a numeric concept id vs. an alphabetic code or a short display name vs. a longer more formal display name. Reference sets need to identify which representation (or set of representations) are actually permitted.

6.2.4 What constraints are there on post-coordinated concepts?

In a post coordinating code system, selecting the set of "base" concepts is not sufficient. Concepts can have additional details conveyed and those details can have additional details. For example, given a particular diagnosis, how sudden was the onset? How severe is the condition? What kind of body part does it affect? Is that body part the left or right?

To fully define a post-coordinating reference set, constraints shall be defined on each of the possible post-coordinating axes available from the allowed concepts. Which axes shall be specified? Which are optional? Which are not permitted? Where multiple repetitions might be allowed, constraints on the minimum and maximum number of repetitions should also be documented. Moreover, this shall be done at each possible level of post coordination.

6.2.5 What is post-coordinated vs. Pre-coordinated?

Even when all allowed concepts and the constraints on post-coordination axes are specified, there can still be imprecision. In a post-coordinating code system, there may be more than one way to express the same concept. There may be both a pre-coordinated and a post-coordinated way of expressing a concept, or even a mixed representation where part of the concept is pre-coordinated and additional aspects are post-coordinated. In many systems, it will be desirable to restrict the representation of a given concept to only one form to allow for easy comparison. If so, the terminology specification will need to define which representation is preferred.

In addition, when communicating post-coordinated concepts, the order in which the various post-coordinated constructs can be expressed can vary. Again, consistency is often desirable in which case, the preferred or normalized ordering of post-coordinated concepts shall also be clearly documented.

6.2.6 What happens if the concept exists more than once in the reference set?

In multi-code system reference sets, there is the potential for the same concept to exist in more than one code system as for both to be permitted within the reference set. This is generally not recommended, as even if the concepts in both code systems are close to identical, there are often subtle differences in the semantics that can make deciding between the two codes challenging. Where this does occur, the terminology specification should therefore provide guidance on which of the code system representations should be selected as the preferred representation.

6.3 Reference set bindings

While some terminology specifications may define a particular set of codes independent of a particular usage, most terminology specifications then link the various reference sets they have defined to the locations within the specification where those reference sets are expected to be used. This is often referred to as binding. The same reference set may be referenced in more than one place and the constraints on how it is referenced may vary in each location.

6.3.1 Is the binding to the reference set static or dynamic?

Like code systems, the definition of a reference set can change over time. Codes can be added to and removed from the enumeration for extensional reference sets. The expressions can be updated and refined for intentional reference sets. Furthermore, reference sets that were once enumerated may be changed to be defined by expression or vice versa.

When linking a particular data model, user interface, decision logic or other element to a reference set, it needs to be clear whether the link is to a specific version of the reference set, or whether those who implement the terminology specification are expected to keep up with revisions to the reference set definition over time.

Similar to the earlier discussion on code system versions, in situations where the reference is independent of a specific version, terminology specifications shall document expectations of update frequency. What is the allowed time gab between the time a revised reference set definition being published and the time conformant systems are expected to use the revised definition? Strategies such as having reference set versions become effective at a future date or ensuring support for a transition period where different versions may be in use.

6.3.2 What reference set representation capabilities are supported?

Another consideration when a binding is not to a specific reference set version is the ability of the systems to successfully process future versions of the reference set that can be defined using intentional expressions. Specifically, what constraints exist on the types of expressions that might be used to define reference set content? This can be relevant even if the reference set is presently extensional, as future versions could still be expression-based.

Terminology specifications shall therefore define the constraints on what types of expression are permitted (including the possibility of constraining to enumeration only). An alternative is to reference an external specification or terminology service that defines the boundaries on reference set expressions.

6.3.3 Is the reference set extensible?

A binding identifies a set of codes to be used in a particular context. However, guidance is also required on what happens if there is not an appropriate code available within the bound reference set. There are two options. One is to prohibit the element from being captured/transmitted/used/etc., if no appropriate code is available. This means that the binding is “non-extensible”.

The other approach is to allow free text and/or local codes to be sent in place of one of the approved codes in circumstances where none of the approved codes is appropriate. In this case, the binding is considered “extensible”.

6.3.4 What expectation is there to support all codes within the bound reference set?

The reference set bound to a particular use in the terminology specification defines the set of codes that are allowed. However, it alone does not define whether implementers are expected to support all of those codes. For example, is a lab application compliant if it only allows selection of a subset the lab test types supported by a terminology specification? What about a physician office system that only allows the selection of two out of the thousands of diagnosis types supported by a terminology specification?

The specific meaning of “supporting” a particular code depends on the intended use of the terminology specification. It might mean allowing users to select the code, exposing the code to users, being able to transmit or receive the code, processing the code when performing search or decision support functions, etc. Additional discussion on “support” will be found in Clause 7. However, regardless of the intended use of a terminology specification, distinguishing which codes must be supported and which are “optional” is an essential part of being able to validate conformance.

When it comes to "support of codes", bindings fall into one of the following three categories.

a) All codes are optional. Implementers and derived specifications are expected to use the specified codes if they need them, but can choose any subset they consider relevant.

b) All codes are required. Implementers and derived specifications are expected to support all of the codes within the bound reference set.

c) A subset of the codes is required. Implementers are expected to support all codes within a specified subset of the reference set. All remaining codes are considered optional and may be supported or not, as deemed appropriate by the implementer or author of the derived specification.

In the last scenario, a second reference set shall be specified that identifies the subset that shall be supported. Obviously, this second reference set shall be a proper subset of the base reference set and shall remain a proper subset even as either of the two reference sets and their underlying code systems evolves over time.

6.4 What is the reference set bound to?

Often, bindings are created to a single data element "This attribute in this class uses reference set X" or "This field in this message uses reference set Y". This provides a great deal of control of exactly what set of codes can be used in a given location. However, it does not necessarily provide consistency of what codes are used in multiple data elements within a single specification or even across specifications.

The alternative binding approach is to define a named concept space or concept domain that can be referenced by multiple data elements. The reference set is then separately bound to that concept space or concept domain and automatically applies to all elements within any specifications covered by the binding that reference the bound concept space or domain. For example, an organization might have a number of specifications that contain data elements that need to be drawn from drug codes. Rather than binding each element individually to a drug code reference set, the organization could instead have all of their specifications reference a "Drug Code" concept domain. They can then bind the reference set once to the Drug Code domain and affect all of their specifications. If they want to change the set of permitted drug codes, they now only need to change one binding at the concept domain level rather than revising each of the specifications that reference the Drug Code domain.

6.5 When and where does the binding apply?

In many cases, when a binding is defined, it applies for all uses of the specification in which it is defined. However, in some cases, multiple bindings may be defined, each intended to be used in different circumstances. For example, a specification might be created for healthcare claims at a national or international level with recognition that achieving a single reference set binding for "billing codes" is not achievable. Therefore, multiple distinct bindings can be created, each applying within a particular country or even jurisdiction within a country. The country or jurisdiction thus provides context to the specification and determines which binding applies.

In some cases, the context may be much narrower and may even depend on other data elements within a specification. For example, a sophisticated specification for capturing surgical reports might have a data element to capture the diagnosis that led to the surgery. The reference set that applied to that data element could change based on the value that was selected for the type of surgery. The set of possible diagnosis that would apply for one procedure would differ from those for another procedure.

7 Terminology usage conformance considerations

In addition to expectations of behaviour directly associated with terminology artefacts, there can be additional expectations around the "use" of those terminology artefacts. Because conformance is about comparing actual behaviour with expected behaviour, it is essential that all aspects of behaviour be fully defined. This Clause defines some additional behavioural considerations based on how the terminology artefacts are intended to be used.

Note that just because these behavioural aspects can be defined does not necessarily mean that they will or should be included in all terminology specifications. Many standards and specifications will choose to allow a degree of flexibility to encourage diversity and creativity in implementations and to allow for market differentiation.

7.1 Data capture

These considerations apply for terminologies expected to be used for data capture by humans.

7.1.1 Are the code system contents expected to be exposed directly?

Are the codes, display names, or other information defined by the code system expected to be exposed directly, or can the implementation expose the content using maps to application-specific terms or some other interface convention?

7.1.2 What aspects may be or must be exposed to users?

Are users expected to be shown the codes, the concept designations, the descriptions, or both? Are they provided access to definitions or other properties of the code system when making their selections?

7.1.3 Are the available codes to be displayed in a particular order?

Order of presentation has an impact on likelihood of selection. Where this matters (for cost or other reasons), the terminology specification shall indicate what constraints there in order: whether it is a complete ordering, the identification of a few key concepts that are ordered, or an algorithm defines the ordering.

7.1.4 Are there constraints on how the codes are to be navigated?

Codes can be presented to users in a variety of ways: radio boxes, drop-down boxes, tables, tree-views, type-ahead searches, or more sophisticated search interfaces. Each approach has different effects on ease of use and how likely a particular concept is to be selected. Terminology specifications could get as specific as defining the search algorithms to be used when retrieving candidate codes based on user-specified keywords.

7.1.5 Are deprecated, retired, or pending codes expected to be presented differently?

If codes, representations, or relationships with statuses other than “active” are to be exposed to users, is there an expectation that those vocabulary artefacts will in some way be distinguished to the user? This could involve grouping them separately, colour-coding, marking with icons, or some other convention.

7.1.6 Shall the reference set or version be captured?

For data that is being captured for use in statistical or other analysis, it can sometimes be relevant to know not only the code that was captured, but also the complete set of codes that were available at the time of capture. The idea behind this is that as new codes are introduced over time, the “most appropriate” code for a given concept will also change. If the initial value set consists of “red”, “green”, or “blue”, then a somewhat orangey colour will likely be categorized as “red”. If orange were later added to the reference set, then future occurrences would make use of that code. The capture of the reference set or reference set version allows such considerations to be included in future analysis.

7.1.7 Can external knowledge be applied to the selection of codes?

Not all codes can apply in all circumstances. For example, a diagnosis of “pregnancy” is rare in most human males.

With post-coordinated code systems, it is even possible to create post-coordinated expressions that are nonsensical. For example, viral pneumonia due to bacterial agent. Some systems therefore attempt to improve data quality by prohibiting the capture of data believed to be nonsensical.

The introduction of external knowledge in the selection of codes has costs and benefits. The obvious benefits are the improvement in the quality of data captured and the potential reduction of data selection effort by the individual doing data capture. However, if the external rules are not carefully considered, they may prohibit "Reasonable" choices and thus either skew or prevent the capture of accurate data. For example, a prohibition against male pregnancy may encounter issues where the male has undergone a physical (or even just a social) gender change that leaves them with an intact uterus. As medicine and science evolve, combinations that were once considered impossible might need to be captured and might even become common.

Terminology specifications should therefore identify whether there should be any restrictions on data capture beyond the logic expressed in the reference set binding.

7.2 Data exchange

These considerations apply for terminologies expected to be shared between applications, either as part of an electronic transmission or as part of an electronic persistence and sharing terminology specification. Most of these questions relate to the data structures or datatypes used to express coded data when exchanging data.

7.2.1 Identification of code systems

It is possible for the same code symbol representation to exist in different code systems and to represent completely distinct concepts in each of those code systems. Therefore, if a data element is bound to a reference set that draws from more than one code system or the binding allows exceptions from local code systems, then the code system shall be identified in instances to allow the codes to be properly distinguished. Even if the reference set currently bound does not contain any duplicate codes, the underlying code systems can evolve in the future to cause such duplicate codes to exist and thereby cause confusion.

In order for data exchange to be successful, the systems involved in the exchange shall both identify the code system in the same way. This might be done using a shared "code system of code systems" (e.g. "001" = SNOMED CT, "002" = ICD10, etc.). Alternatively, a registry identifier from a code system registry might be used. The key requirement is that the terminology specification requires the use of a single mechanism for identifying code systems and that there is no ambiguity about which code system is being represented.

7.2.2 Identification of code system versions

In some "poorly behaved" code systems, the meaning of a concept can change between versions of the code system. In these circumstances, "code" + "code system" is not sufficient to uniquely identify the concept being represented. In these circumstances, the version of the code system shall also be shared. The terminology specification shall both identify the requirement for sharing the code system version and define how the version is to be identified. For example, by date or timestamp or by some official version number associated with the code system.

7.2.3 Syntax for post-coordination

In post-coordinating code systems, the "code" is not a simple string, but rather a whole expression that combines multiple codes together. In many cases, the syntax for serializing this expression will be defined by the code system. However, in some cases, syntax will need to be defined or constraints will need to be placed on the syntax to allow for consistent communication.

7.2.4 Presence and representation of translations between code systems

Frequently, the concept required for communication will not be captured in the code system it is required to be expressed in for exchange. This can occur for the following number of reasons.

— The system might need to communicate using a variety of terminology specifications, each requiring the use of different code systems. Users, on the other hand, only want to encode a given concept once.

— The data might have been captured before the system became compliant with the terminology specification requiring a different code system

— The data might have been captured by a different system and the current system must deal with what it has.

In these circumstances, the initially captured code shall be translated (see 7.2.4) to the desired code system. In some cases, based on the availability of maps between the code system in which the concept was originally captured and the "desired" code system, multiple translations can be required.

EXAMPLE Local code system to SNOMED CT to ICD10.

Every time translation occurs, there can be a loss of semantic information (two independent code systems rarely represent exactly the same concept). A system that understands both the original code system and the code system specified by the interoperability standard would be better using the original code system, as it will most accurately convey the semantics of the concept.

As well, the quality of translation maps varies. Sometimes, translation maps may contain errors or "non-optimal" translations. Newer versions of the target code system may allow for better translations. As a result, new versions of translation maps may be introduced over time.

To preserve the original semantic information and to allow the data to be re-translated in the future using newer translation maps, some terminology specifications may require that all translations be sent along with the target code, possibly also including information about the order in which the translation occurred.

7.2.5 Presence of original text

In "extensible" reference set bindings, there is the potential to capture a free text description when the appropriate concept is not available. However, some terminology specifications may support or even require that "original text" be sent even when a coded value is specified. In this case, the original text serves as the most accurate representation of the concept as selected by the person originally capturing the data element. In situations of post-coding, this might be the user's initial textual description. In situations where the user selects a code directly, this would be the text they viewed when the code was selected. In either case, it represents what the user saw and expressed when they captured the data element.

This information might later be used to re-encode the concept in other code systems or even in the original code system. In some cases, it may also help satisfy legal requirements to retain and exchange the information actually seen by the authoring person.

7.2.6 Terminology specifications

Terminology specifications should therefore clearly declare their expectations for support of original text with coded elements, as well as the presence of concept designations and other code system information

When exchanging coded information with other systems, there may be variations in how much, if any, information the receiving system has about the code system in which coded information is represented. Systems that have a current local copy of the code system can manage with just enough information to look up the code in their local copy. Descriptions, definitions, hierarchical relationships, and other information can then be looked up in the local copy for use in rendering and other processing.

However, if a receiving system does not (or might not) have a current local copy, there may also be a need for a terminology service or other mechanism for the receiving system to look up information about a given code or a need to transmit additional information about the code within the instance.

Such information frequently includes concept designations, but could include other information such as definitions, translations and other relationships.

7.3 Data analysis and searching

These considerations apply for terminologies expected to be used in data analysis and information retrieval.

7.3.1 Are subsumed codes included?

One of the most important relationships in many code systems is that of subsumption, i.e. a "parent" code subsumes the semantics of a "child" code. That means that anything that is true for the parent is also true for all children. For example, if the code for "appendectomy" is a specialization of the code for "gastric procedure", then all statements that are true for "gastric procedure" will also be true for "appendectomy".

Subsumption can significantly affect the results of searching and other types of analysis. This includes effects on the results (how many codes are returned or included), as well as on the performance of the system undertaking the analysis. For example, a search for all patient records matching a specific diagnosis code is quite different from a search for all records matching that code or any of its potential specializations.

Terminology specifications should therefore make clear whether, and if at all, exactly where they expect subsumed codes to be included as part of an analysis.

7.3.2 What mathematical support is expected?

Some code systems include mathematical properties. This might be simple less than/greater than relationships with other codes as is found in ordinals. For example, "low" < "moderate" < "high". Others are intended to be used in more complex mathematical relationships, such as addition or multiplication. For example, the components of an APGAR score being summed to determine the overall score.

As with subsumption, introducing mathematical relationships can increase complexity of operations. Terminology specifications should therefore document whether and how they expect mathematical relationships to be supported as part of analysis and/or search capabilities.

7.3.3 How are post-coordinated results handled?

With most code systems, a query to retrieve all concepts matching a particular set of criteria will result in a finite set. However, with post-coordinated code systems, the result set is potentially infinite. For example, the UCUM units for distance potentially include m*s/y, m*s^2/y^2, etc., with no upper limit on the exponents in the numerator or denominator. While these might not be sensible units, there is no easy way for an analysis engine to make that determination.

With many vocabulary operations, subsumption can be handled by finding the enumeration of all possible specialization codes and running the operation on each of them, unioning the results. However, with post-coordinated code systems, enumeration will not always be possible, meaning that other options may be required.

As well, post-coordination means that in addition to dealing with subsumed codes, decision and search logic shall also deal with post-coordinated equivalent concepts where different choices have been made about what portions to pre-coordinate or post-coordinate.

Terminology specifications should therefore make clear what their expectations are for support of post-coordinated expressions when performing analysis functions.

7.3.4 How is cross code-system analysis managed?

Where the data set includes data from multiple code systems, searching and analysis can be more challenging. If the code systems do not contain relationships that span concepts between the two code

systems testing for subsumption and equivalence can be difficult. For example, a search for "Type 1 Diabetes" using the ICD10 code against records containing a mixture of ICD10 and SNOMED CT codes will only match on the records coded in ICD10, not those encoded in SNOMED CT because of the lack of any linkage between the two code systems.

NOTE In the future, there might be linkages available between SNOMED CT and ICD10 so this particular example might not always be valid. However, the principle holds regardless.

Similarly, testing for subsumption between a generic drug formulation code in a WHO maintained code system against codes from a national regulatory dispensable drug code system would only work if the national regulatory code system provides linkages to the WHO code system.

Where searching or analysis across code systems is required, terminology specifications should document what behaviour they expect. This includes documentation of where cross-linkage information can be found, as well as what warnings or errors or other behaviour is desired when analysis of all records is not possible due to the absence of the necessary linkages.

7.3.5 Unknown data

When dealing with encoded data, it is common for some records not to have an appropriately coded value. This might be due to the absence of any information or the element might have been captured in a local code system unfamiliar to the analysis engine or even as original text. There are two considerations for application behaviour when this occurs.

The first is "Does the analysis or search capability includes the ability to search on original text?" For example, doing key-word searches, sounds-like searches, wildcard searches, etc. In the absence of original text, the display names and even, potentially, the definitions and aliases for the coded value might be included as part of the search process. Inclusion of this functionality ensures that non-coded data will be included in any search or analysis. However, text based searching is more error prone that code-based searching. It risks not returning all possible matches due to inability to match on synonyms and wording variations between the text recorded in the record and the text searched for. It also has a tendency to return a large number of false positives, particularly when performing contains wildcard and sounds-like searches.

The second issue is how non-coded (or at least not coded in the code system in use) records will be treated as part of the analysis. The following are three possibilities.

— **Exclude non-coded records:** with this approach, all records without an appropriate coded value is excluded from the search or the analysis. The system might issue a warning along with the results indicating that some records were excluded from the search due to the lack of an appropriate coded value. Obviously, this approach risks excluding relevant data from the result set or analysis.

— **Include non-coded records**: with this approach, all non-coded data elements are considered "potential" matches for a code-based search and are included in the result set. The system might include a warning that "potential match" records have also been included. This approach ensures that records are not missed due to the absence of coded data. However, depending on the prevalence of non-coded data elements in the records being examined, it has the potential to dramatically increase the size of the result set and return a large number of false positives. In some cases, the inclusion of false positives can have privacy implications.

— **Return two record sets:** in this case, both the "definite" coded matches are returned, as well as the non-coded "potential" matches. However, they are returned as distinct result-sets. This ensures that the data returned from the search or used for analysis is complete. However, it avoids polluting the "known" match data records with those that require additional human analysis to determine whether they are matches or not.

To ensure consistency, terminology specifications should document their expectations for the handling of non-coded records.

8 Sharing and persisting conformance expectations

The previous subclauses have discussed the types of conformance expectations that should be documented. However, in addition to "what" should be documented, it is also important to look at "how" that information should best be presented.

Documentation of conformance expectations is likely to be used in the following two significant ways.

— It will be used directly by developers for import into their applications. For example, importing code systems, reference set definitions, and other structures that will be needed for operation of their system.

— It will be used to assess the conformance of systems that attempt to make use of the terminology specification.

Neither of these use-cases is best served by "document" representations of conformance expectations. Many developers have experienced the frustration of attempting to copy and paste tables out of a PDF rendered document. Such extractions are prone to error and act as a barrier to automated verification of conformance.

Terminology conformance expectations should therefore be published in a computation-friendly format. Examples might include XML, spreadsheets, comma-separated variable (CSV), or similar syntaxes. Some standards organizations have developed formal syntaxes for the representation of their terminology artefacts, such as Model Interchange Format (MIF) for HL7 and Terminology Query Language (TQL) for Open EHR.

NOTE While these formats might have initially been developed for use by these organizations, the syntaxes can be used by specifications using different standard technologies.

Computation-friendly representation is particularly important for those data elements that will be the focus of direct software manipulation and evaluation. This includes code system, reference set, and reference set binding definitions. Other content, particularly the implementation guidance identified in the Clause 6 can reasonably be handled as free-text in an implementation guide because it affects developer decisions, not automated computation functionality.

9 Asserting conformance

Terminology specifications have little use unless they are used and are recognized as being used. This means that systems and more refined specifications need to assert conformance to the base specification. Fortunately, there is usually significant market incentive to declare conformance to a specification (sometimes, too much incentive; refer to Clause 11) If a specification is deemed useful by customers of products, then software vendors and smaller, more refined specifications will do their best to declare conformance with it.

However, declaring conformance with a terminology specification is not necessarily a simple thing. Software vendors (and often their customers) would often be happy just making a simple statement such as "my product is compliant with Standard X". Many systems do make statements similar to this. Some even make statements such as "my product is compliant with Standard Organization Y". In both cases, such statements are rarely either accurate or sufficient in detail to truly express the capabilities of the system with respect to the standard.

To be able to properly assess and validate conformance, it is necessary to have a clear and complete statement of how the system conforms, such as statement is called a "Conformance Statement". This Clause will examine what aspects should be present in good conformance statement.

9.1 Conformance and non-conformance

The first thing to recognize is that a Conformance Statement is often as much about non-conformance as it is about conformance. A system or terminology specification need not be fully conformant to construct and publish a Conformance Statement. It is perfectly legitimate to say "My product fully complies with Terminology specification XYZ except for the following limitations: …" For some customers, the listed

limitations might not be important. Even if they are, the rest of the product might meet requirements sufficiently well that it is worth acquiring the product and working around the limitations. In some cases, the "limitations" may actually reflect deficiencies or unrealistic expectations on the part of the terminology specification being conformed to.

A Conformance Statement can be defined as follows: a document that expresses the capabilities of a system in terms of the requirements and expectations defined in a terminology specification the system is claiming conformance to. "Express capabilities … in terms of" means that the Conformance Statement can express both areas where the system fully meets the requirements, as well as those where it does not.

9.2 Why conformance statements?

Conformance statements are necessary to ensure a complete representation of what the system can do in terms of a terminology specification. Terminology specifications are often complex. Most have at least some degree of optionality. Many have areas of ambiguity. Even where neither of these issues occurs, systems will frequently fail to meet one or more of the stated requirements in some manner. A Conformance Statement is a manner of placing all this on the table.

— What optional portions of the terminology specification are supported and which are not?

— In areas where a terminology specification might be ambiguous, how has this particular implementation interpreted the terminology specification?

— In what areas is this terminology specification in alignment with the expectations documented by the terminology specification?

9.3 What is conforming?

The first essential part of a Conformance Statement is the identification of the system that is conforming. In today's software environment, systems may be made up of numerous components, sometimes supplied by different vendors. The Conformance Statement should make it clear exactly what components of software or which pieces of a derived specification are claimed to be "in alignment" with the terminology specification being conformed to.

Software is not static. It evolves. New versions of products are released. New extensions are produced. The Conformance Statement needs to be explicit about which version or versions of the specified system or collection of systems is being asserted to be conformant with the target terminology specification.

Finally, even single versions of a given system will often be configurable such that different installations will demonstrate different behaviour. A conformance statement will therefore be specific to a particular configuration or will need to include some optionality to reflect the variability resulting from configuration parameters.

A complete Conformance Profile should therefore indicate the following:

— names of all components required to achieve the behaviour documented by the profile;

— version or versions of each of those components that are asserted to be covered by the profile;

— any configuration settings that are necessary to achieve the behaviour described by the profile.

9.4 What is being conformed to?

After identifying the system that is conforming, it is also necessary to identify the terminology specification being conformed to. In most cases, this will be to a single terminology specification (which may itself be dependent on other terminology specifications). Limiting a conformance statement to a single terminology specification will significantly simplify the statement. If the statement references multiple terminology specifications, then every assertion within the statement will need to identify which terminology specification it is dealing with. Where multiple terminology specifications shall be referenced, it may be simpler to create multiple "sub"-conformance statements within the overall statement.

The terminology specification(s) being conformed to shall be explicitly identified, usually by name and version. In some cases, a system may claim conformance to individual terminology artefacts (i.e. code systems or reference sets), in which case the artefacts and version(s) need to be explicitly referenced.

In situations where the referenced specification itself refers to other specifications (which may refer to yet other specifications, etc.), the conformance statement should ensure that these "incorporated" specifications or specification sections are also explicit in terms of the versions to which conformance is being asserted.

Finally, many terminology specifications are made up of multiple components. Not all terminology specification components will necessarily apply to all systems. Therefore, the Conformance Statement should clearly identify which components of the terminology specification are being conformed to (and by implication, which components are not).

9.5 Assumptions

In some cases, the terminology specification referenced might not have been explicit about some aspect of its expected behaviour. For example, a reference to an external coding system that does not declare whether the reference is to the "current version" of the code system or to the version that was in place when the terminology specification was published. In these situations, the Conformance Statement shall declare what assumption has been made. If assumptions are not declared, then appropriate testing cannot be performed and impact of differences in assumptions made by distinct systems cannot be assessed.

Assumptions should be documented in any place where the referenced terminology specification has not captured expectations to the degree recommended within this Technical Report. All other assertions within the conformance statement will be based on the assumptions stated. For example, constraints of optionality will be based on optionality resulting from stated assumptions.

9.6 Support for optional elements

One of the key aspects of Conformance Statement is an indication of which optional components in the referenced terminology specification are supported and which are not. This should occur for every optional element. Possibilities are listed below.

— **Stays optional:** optional element is optional in the derived specification as well or is optional within the implemented system managed through configuration parameters.

— **Supported:** optional element is treated as though it had been flagged as "required", i.e. the element is fully supported within the system or terminology specification.

— **Not Permitted:** optional element is treated as though it were not part of the terminology specification at all. For example, a "non-permitted" code would not be selectable in the user interface or would be rejected as an error if received.

— **Ignored:** optional element will not raise an error. However, it also will not be "processed" like a supported code. For example, the code would not raise an error if received and would not be processed in decision logic or displayed. This type of support is primarily relevant when defining Conformance Statements for systems that are receivers of an exchange or are performing analysis. "Ignoring" content still allows for a degree of interoperability by ensuring that exchanges will not fail even in situations where codes are not supported.

For example, a conformance statement against a reference set binding might do any or all of the following:

— reduce the reference set to exclude some of the optional codes (making those codes 'not permitted);

— increase the size of the "must support" reference subset to include additional optional codes making those codes "supported";

— define an additional sub-reference set of some of the optional codes indicating those codes will be ignored;

— leave optional codes as optional.

Obviously, a particular data element can only fall into one of the four categories. This means that the subset of ignored codes shall not have any overlap with the required codes subset and both shall be proper subsets of the "allowed" codes.

Reference set bindings can be even more complex if the binding has been defined as "extendable". In this case, the allowed set of "optional" codes also includes all possible local codes that are outside the base reference set. A conformance profile might extend the base reference set to explicitly include additional codes that would otherwise have been part of this implicit set. As well, either the "required" set or the "ignored" set could be marked as "extendable", meaning that either all local codes must be supported or all local codes will be ignored. (If neither the required nor the ignored subsets were "extensible", then all local codes would be considered "optional".

Note that "should" and "may" criteria fall under the banner of "optional" elements and shall be categorized as supported or not.

9.7 Variations

The final component of a conformance profile is defining those areas where the system or derived terminology specification is not in alignment with the behaviour expected in the terminology specification being conformed to. This is an extremely wide-open area. Examples might include the following:

— only supporting a subset of the languages allowed by a code system;

— using a code with a different semantic than is suggested by the code's definition;

— using different codes than are permitted by a reference set;

— not allowing navigation by one of the expected relationship types;

— using a different string for identifying a given code system.

Some of these variations might have significant impact on implementation and others might not affect most implementations at all. Most variations are likely to result in all or part of a system being deemed "non-conformant". The criteria for exactly what makes a variation non-conformant will be discussed in more detail in the next subclause.

9.8 Completeness

Specifying the above information will create a technically sufficient Conformance Statement, i.e. it will clearly state what the system or derived specification does. However, it is often useful to capture rationale for the decisions made. For example, why are some codes required while others are not permitted or ignored? Rationale is particularly relevant when dealing with "should" requirements. Examples of rationale might include the following:

— perceived errors in the base specification;

— not yet available, planned for future release;

— not relevant for business environment.

The rationale is useful for the author of the terminology specification because it helps them justify their non-compliance. It is useful for potential users of the terminology specification or system because it gives them insight into the decision by the author of that system or terminology specification and helps them to evaluate whether the non-compliance is "important" to them. Finally, it is useful for the authors of the base terminology specification should they receive a copy of the conformance statement because it may indicate areas of their terminology specification that are flawed, too complex, or are infrequently being taken advantage of. This will in turn allow them to improve future versions of their terminology specifications.

10 Evaluating conformance statements

Once a Conformance Statement has been produced for a given system or derived specification, the next step is to evaluate that statement and determine whether the system is indeed "conformant" or if not fully conformant, which aspects of the terminology specification they can claim to be fully conformant with, i.e. "In what areas is this system meeting the expectations set out in the terminology specification?"

The rules for determining conformance based on element support can be summarized as follows:

a) all "must" and "shall" requirements are met;

b) all "required" elements are supported;

c) all "conditional" elements are satisfied for the conditions stated;

d) where repetitions are supported (e.g. qualifiers in post-coordinated expressions), the terminology specification can handle the maximum number of allowed repetitions and will always satisfy the minimum number of repetitions.

Any violations of these assertions mean that the terminology specification or system is non-compliant in that area.

The evaluation process is usually self-performed by the author of the conformance statement. As a convenience, the conformance evaluation should be published or provided as part of the conformance statement to save everyone who reads the statement from having to undertake the analysis. However, anyone who wishes to confirm the evaluation can do so provided that they have a copy of both the conformance statement and the base specification.

11 Verifying conformance

The final step of the conformance process is to verify that a system described by a conformance statement actually operates as the statement asserts. Between totally trusting parties, this step can be considered optional, although to account for the potential of error in the creation of a profile is still wise. For systems that wish to be "certified" as conformant (have their conformance statement verified by a trusted third party), this step will be mandatory. Verification can theoretically also occur against derived specifications, although this is more of a manual exercise confirming the assertions of the conformance profile against the text of the derived specification.

Because verification is essentially a testing process and is no more unique to terminology than to any other aspect of a system, this terminology specification will not provide details on testing strategies. Other terminology specifications provide more complete guidance in this area. However, this Clause will provide some considerations for the terminology verification process.

11.1 What is verified?

In theory, the verification process should exhaustively confirm the consistency of a system's behaviour with every aspect of the conformance statement. For practical reasons, only a small subset of the functionality of a system can be verified, usually using a set of representative tests to check various boundary conditions. The rigor and complexity of the tests will depend on the complexity of the underlying specification and the amount of effort the verifier wants to invest.

Note that the system's behaviour is not verified against the base specification, but rather against the conformance statement. This means that the verification process only checks those aspects the system actually claims to be conformant with and makes allowances for those parts of the system that have been acknowledged non-conformant. In some cases, such as certification, verification might be limited to those systems that claim 100 % conformance with the complete set of terminology specification capability or allowed subsets.

11.2 Who verifies?

Verification can be performed by anyone. Frequently, the initial verification will be performed by the author of the derived specification or the developer or vendor of a particular system. It is in their interest to ensure that any statement they publish is accurate; particularly as such statements might be relied on as part of legal contracts. Verification might also be done by potential or actual purchasers of a system or by third parties at the request of such a purchaser. Third parties may also offer services as a "trusted verifier" or "certifier" who develops the infrastructure necessary to test multiple systems against a particular terminology specification and can offer a degree of confidence to purchasers that a system does what it says it can do.

11.3 Can verification be automated?

Any sort of testing process is resource-intensive. The ability to automate greatly reduces the costs associated with testing. Unfortunately, the answer is not an unqualified "yes". Some aspects of vocabulary conformance verification can be automated, while others either cannot or are much more difficult to automate. Much of this difficulty stems from the variations in system behaviour that are not defined as part of the terminology specification.

Any aspect of vocabulary conformance verification that involves either population of or conformation of data exposed by a user interface will be more challenging to automate. Automated user interface testing tools is certainly possible, but these systems usually require manual customization for the interface of each system they intend to test. The effort to perform the configuration is often more than the cost of executing the test a single time. Thus, automation is only economic if the testing will be performed on the same system (and same system version) multiple times. This is rarely the case in most conformance verification or certification scenarios.

Verification of exchange scenarios when testing the ability to "consume" information is easier. Testing emitting and logic functionality can be automated if there are exposed programmatic or service interfaces with which to manipulate the system.

At minimum, automation can be used to generate test scenarios when given a computable terminology specification, allowing robust verification of the terminology artefacts referenced within the terminology specification. This can be done by parsing of the reference set expressions and generating tests for both valid and invalid codes.

12 Other considerations

This Clause covers additional issues that may be relevant to terminology conformance but are not strictly part of the conformance assertion and verification process.

12.1 Comparing conformance statements

In addition to documenting, evaluating, and testing the conformance of a system, conformance statements can also be used to evaluate interoperability with other systems. Regardless of how conformant or non-conformant a system might be, if the conformance statements for two systems are in alignment, they will interoperate. For example, if both systems use the same custom code system in place of the one recommended by the terminology specification, they would both be non-conformant but will still interoperate.

To evaluate interoperability, check whether the system that will be emitting the information is a proper constraint on the system that will be consuming the information.

a) The set of codes supported by the emitting application should be a proper subset of the set of codes that are supported by the consuming system.

    1) For semantic interoperability, all "required" codes by the sender shall be "required" codes for the receiver.

2) For merely "non-breaking" interoperability, all "required" codes by the sender shall be either "required" or "ignored" codes for the receiver.

b) If the sender allows "extensibility", the receiver shall also support (or ignore) extensibility

c) The receiver shall support at least as many repetitions of things such as post-coordination qualifiers as the sender does

d) The sender shall send the minimum number of repetitions expected by the receiver.

e) The lengths of codes, original text, and other items produced by the sender cannot be any longer than those of the receiver.

f) All data elements (display names, original text, translations, etc.) sent by the sender shall be either accepted or ignored by the receiver.

12.2 Conformance with conflicting terminology specifications

Often, application developers are faced with a requirement to support multiple terminology specifications rather than just one. They may have customers with different needs, need to operate under multiple jurisdictions, or interoperate with both legacy and "new" ways of communicating information. This can include supporting both current and older versions of a terminology specification. Unfortunately, these multiple terminology specifications will often introduce conflicting requirements.

One possible solution is to create multiple versions of the system, one for each target specification. However, this gets unmanageable quickly and is expensive. Another option is to build one system that provides all of the required capacities for the various specifications and use configuration options to control which behaviour is invoked when. Unfortunately, this adds significant complexity to the application, as well as complexities for ongoing support as each application has many different behaviours.

One option that terminology systems provide is the ability to use translations. This means that an application might use a single terminology internally in its logic and for exposing concepts to users. It then maps to the various terminology specifications' code systems before sharing data or invoking decision logic. Where translations are permitted, it may even be possible to populate all the various representations required by the different specifications in a single element. This approach allows a system to be compliant with multiple inconsistent specifications simultaneously.

For this approach to be most effective, the "internal" code system needs to be as granular as the most granular of the terminology specifications being mapped to. As well, maps shall exist (and be actively maintained for non-versioned vocabularies) to all of the target terminologies.

The mapping approach has some caveats. It only works in circumstances where the terminology specification does not dictate exactly what the user interface displays to the user. As well, the terminology specification shall provide support for communication of translations and shall not have specific rules about where the translation for a particular target code system shall be located within the set of translations.

12.3 What IP considerations are associated with the code system?

While not specifically an aspect of conformance, one consideration in the conformance of a system is its ability to conform to the intellectual property constraints associated with the vocabulary artefacts, particularly those of the code systems used. Obviously, if a system cannot comply with the licensing and other requirements of the vocabulary artefacts used by a terminology specification, it will not (legitimately) be able to comply with the terminology specification.

12.4 Terminology services

Terminology services are not specifically a part of terminology conformance. They would behave and be validated in much the same way as any other software service terminology specification. However, implementation of a terminology-based system may also require implementation of and support for

terminology services specifications. This is particularly important when dealing with terminology specifications that require systems to keep up with revisions to code systems and/or reference set definitions. Terminology services provide a means to retrieve these updates in real-time or near real-time.

For a terminology service to meet these requirements, it shall support the expression of the specification's code systems to the level of rigor required (e.g. languages, properties, relationships, post-coordination, multiple display names, etc.). Similarly, it shall support the specification's reference set representations, including complexity they might have in the future.

Bibliography

[1] ISO/IEC 10641, *Information technology — Computer graphics and image processing — Conformance testing of implementations of graphics standards*

[2] ISO/IEC/TR 13233, *Information technology — Interpretation of accreditation requirements in ISO/IEC Guide 25*

ICS 35.240.80
Price based on 27 pages

This page deliberately left blank

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.

...making excellence a habit.™