HL7 Services-Aware Interoperability Framework (SAIF)

Enterprise Conformance and Compliance Framework

March 12, 2010 Peer Review Draft

Health Level 7
# Table of Contents

1. 1.4. Enterprise Conformance and Compliance Framework
    1.4.1. ECCF Goals ................................................................. 61
    1.4.1.1. Prerequisite Knowledge and Outcomes ....................... 61
    1.4.1.2. Important ECCF Terms .............................................. 62
    1.4.1.3. ECCF System Map .................................................. 62
    1.4.2. ECCF: A Template for Working Interoperability ............... 63
    1.4.2.1. Levels of Working Interoperability ......................... 65
    1.4.3. ECCF Problem Statement ............................................ 65
    1.4.4. ECCF Foundational Concepts ....................................... 67
    1.4.4.1. ECCF Reference Scenario ....................................... 67
    1.4.4.2. ECCF Specification Stack ....................................... 68
    1.4.4.3. Conformance ........................................................... 70
      1.4.4.3.1. Example: Conformance in a Specification Stack ........... 71
    1.4.4.4. Compliance ............................................................ 72
      1.4.4.4.1. Example: Compliance in a Specification Stack ............ 73
    1.4.4.5. Certification .......................................................... 74
      1.4.4.5.1. Compliance Validation ....................................... 75
      1.4.4.5.2. Consistency ...................................................... 76
      1.4.4.5.3. Consistency, Traceability, and Compatibility in a Specification Stack 76
    1.4.4.6. Traceability ............................................................ 77
    1.4.4.7. Compatibility .......................................................... 78
    1.4.4.8. Localization ............................................................ 78
    1.4.4.9. Jurisdiction ............................................................ 79
      1.4.4.9.1. Provenance ....................................................... 79
    1.4.4.10. Jurisdiction ........................................................... 79
    1.4.4.11. Provenance ............................................................ 79
    1.4.4.12. Technology: Viewpoint, Bindings, and Implementations .... 80
    1.4.4.13. Comparison of Conformance and Compliance Lexicons: ECCF vs. HL7 Implementation and Conformance Work Group .................................................. 81
    1.4.5. Completeness of a Specification Stack Instance ................ 81
      1.4.5.1. Applying the ECCF ................................................. 82
      1.4.5.2. Mature and Immature Specification Stack Instances .......... 84
      1.4.5.2.1. Traceability of Conformance Statements in a Mature Specification Stack Instance 85
      1.4.5.2.2. Traceability of Conformance Statements in an Immature Specification Stack Instance 86
      1.4.5.2.3. Conformance Certification: Mature vs. Immature Specification Stacks 86
      1.4.5.3. Incorporating Reference Specifications in a Specification Stack Instance 87
      1.4.5.3.1. HL7 Reference Specifications Example 87
      1.4.5.3.2. Externally-Developed Reference Specifications 88
    1.4.5.4. ECCF Specification Stack Diagrams ............................ 89
    1.4.5.4.1. Specification Stack Template ................................. 89
    1.4.5.4.2. The Full ECCF Specification Stack .......................... 90
    1.4.6. ECCF and Enterprise Architecture Specifications ............... 91
    1.4.6.1. The ECCF as the Skeleton of an EAS .......................... 93
    1.4.7. The Value of an ECCF ................................................. 93
    1.4.8. ECCF Appendixes ....................................................... 94
      1.4.8.1. Harmonization and Mapping between ECCF and HL7 Implementation and Conformance Work Group 94
      1.4.8.1.1. Comparison of ECCF and HL7 Implementation and Conformance Work Group Concepts 97
      1.4.8.1.2. Reference Notes for the SAEAF and HL7 2.8 Comparison 99
    1.4.8.2. ISO Terminology: Conformance and Compliance ............ 99
1.4. Enterprise Conformance and Compliance Framework

This document discusses in detail the motivation for, as well as the structure, content, and use of, the Enterprise Conformance and Compliance Framework (ECCF).

As discussed in the SAEAF Introduction and Overview, the Services-Aware Enterprise Architecture Framework (SAEAF) consists of four core sub-frameworks:

- Information Framework (IF)(8)
- Behavioral Framework (BF)
- Enterprise Conformance and Compliance Framework (ECCF)
- Governance Framework (GF)

When appropriate, this document mentions relevant details of the other core SAEAF sub-frameworks. However, those references are not definitive explanations of the other sub-frameworks, as the larger “SAEAF book” (of which this document is a part) includes details about the other sub-frameworks.

1.4.1. ECCF Goals

The major goal of the Enterprise Conformance and Compliance Framework (ECCF) is enabling Working Interoperability between different users, organizations, and systems.

The ECCF is manifest in a structure called the ECCF specification stack (SS). This structure identifies, defines, organizes, and relates a set of artifacts that collectively specify the relevant semantics of a software component specification or other system-of-interest.

In summary, the ECCF SS provides an organizational framework in which inter-related artifacts are sorted by content – for example, business rules, information constructors, behavioral contracts, and level-of-abstraction.

1.4.1.1. Prerequisite Knowledge and Outcomes

This topic covers what you need to know ahead of time and what you will learn after reading about Enterprise Conformance and Compliance Framework (ECCF).

Prerequisite knowledge:

- Knowledge of SAEAF Introduction and Overview.
- Knowledge of Behavioral Framework.
- Parallel exposure to SAEAF Governance, as ECCF is closely linked to Governance.

For example, Standards Developing Organizations (SDOs) can use external standards and specifications in the ECCF specification stack.

Outcomes:

- Understanding the need for a layered ECCF.
- Understanding an applying the ECCF foundational concepts.
- Understanding the specification stack.
- Understanding the relationship of ECCF to Governance.

(8) The Information Framework document will be developed from existing HL7 materials contextualized for use within SAEAF. It will describe the how existing HL7 artifacts such as the Reference Information Model (RIM), data types, Vocabulary Best Practices, Clinical Document Architecture, Clinical Statement Pattern, and HL7 Core Principles are applied in the context of SAEAF.
1.4.1.2. Important ECCF Terms

This topic introduces the key Enterprise Conformance and Compliance Framework (ECCF) terms, such as specification stack subject and conformance statements.

The terminology and phraseology used in discussing various aspects of the ECCF include:

- **Relevant semantics** refers to all aspects of the specification that are stated to enable a component built from the specification to achieve Working Interoperability with another component. (Working Interoperability is a core concept of SAFAF, as depicted in Figure 1 and discussed in detail in the SAFAF Introduction and Overview). Relevant semantics includes both static and dynamic/behavioral semantics, as required by a given specification.

- **Specification stack subject** – or “subject” for short – refers to a particular SS instance and defines the instance’s scope or topic. This discussion uses the term “subject” exclusively to avoid the already overloaded terms “scope” and “topic.” The subject of a specification varies, depending on the granularity, intent, and/or context of the specification.

- **Specification** refers to a collection of artifacts bound to a named instance of an ECCF specification stack, i.e. to a particular subject. Further definition of the term is intentionally left somewhat general at this point in the discussion, because a specification may apply to any one of several subjects including:
  - A single service (business, utility, infrastructure, or other type) within a Services-Oriented Architecture (SOA);
  - A particular message or document; or
  - A general capability of a system or organization (for example, a party engaged in a double-blind clinical trial).

- **Conformance statements** are explicit testable representations of explicit assumptions made by the specification. Conformance statements can be made from a number of perspectives and at varying levels of computational abstraction. They are collected and sorted by type and leveled by layers of abstraction, as explained below.

An implementation of a given specification, as realized through a given technology binding or other operational embodiment (depending on the specific SS subject), asserts one or more pairwise-associated conformance assertions, which can be formally evaluated as True or False.

Related information

- Section 1.4.4. ECCF Foundational Concepts
- Section 1.4.2. ECCF: A Template for Working Interoperability
- Section 1.3.2. Value Proposition: Working Interoperability
- Semantics
- Conformance statement

1.4.1.3. ECCF System Map

The Enterprise Conformance and Compliance Framework (ECCF) system map shows how you can use the ECCF specification stack as a template for achieving Working Interoperability.

---

(9) **Note:** Although the last example is a less common application of the ECCF, it is a viable application and is included to help illustrate the overarching conceptual description of the ECCF.
The ECCF specification stack is a method for defining artifacts for each interoperability paradigm (documents, messages, and services). The artifacts are sorted by content, such as business rules and behavioral contracts.

After you define the artifacts in the specification stack subject, you create conformance statements, which are testable representations of assumptions that the specification makes. An implementation of the specification asserts one or more pairs of conformance assertions as True or False.

### Related information

- Section 1.4.4. ECCF Foundational Concepts
- Section 1.4.2. ECCF: A Template for Working Interoperability
- Section 1.4.4.2. ECCF Specification Stack

### 1.4.2. ECCF: A Template for Working Interoperability

Examining the relevant specification stacks (SS) provides a tractable, scalable approach to assessing the degree of difficulty and specific amount of effort required to enable two trading partners to attain Working Interoperability (WI).

Figures 1 and 2 show examples of Working Interoperability.

*Figure 15. Working Interoperability: Any interoperability paradigm, including messages, documents, or services, can be used in a given WI context.*
The ECCF can be characterized as the infrastructure supporting the SAEAF Stairway to Heaven (see Figure 2).

**Figure 16. SAEAF Stairway to Heaven**

The greater the distance on the Stairway to Heaven, the more difficult the transforms required to achieve WI.

**Reference:** See the SAEAF Introduction and Overview for a detailed discussion of Working Interoperability and for further explanation of the SAEAF Stairway to Heaven.

**Related information**

- Section 1.4.7. The Value of an ECCF
- Section 1.3.2. Value Proposition: Working Interoperability
Section 1.3.2.3. Working Interoperability: Stairway to Heaven

1.4.2.1. Levels of Working Interoperability

The higher up the SAEAF Stairway to Heaven, the easier it is for trading partners to achieve Working Interoperability.

The following figure shows the relationship between the SAEAF Stairway to Heaven and the overall structure of the ECCF specification stack.

*Figure 17. Relationship between the levels of the SAEAF Stairway to Heaven and the rows in the ECCF specification stack.*

The thickness of the arrows indicates the level of explicitness of one trading partner’s specification. Thicker arrows indicate that a trading partner is “higher up” the Stairway to Heaven. As a consequence, the other trading partner is able to work with a more explicit information relationship to the specific WI context, a fact that results in an “easier” path to WI. Specification stack (SS) columns are collapsed for clarity.

Related information

- Section 1.4.4.2. ECCF Specification Stack
- Section 1.3.4.2.3. Model-Driven Architecture
- Section 1.3.4.2.5. Reference Model for Open Distributed Processing

1.4.3. ECCF Problem Statement

Although a robust, scalable, supportable, extensible, and implementable Enterprise Architecture Specification (EAS) requires all of the critical components specified in SAEAF, the ECCF and the specification stack instances supply the “skeleton” from which an EAS can be developed.

Given the following requirements:
• The existence of two intra- or inter-enterprise trading partners wanting to achieve Working Interoperability;

• The existence of a specification of each trading partners’ software/system components that are involved in the Working Interoperability exercise;

• The additional requirement that the specifications have a maximum of flexibility and robustness that enable their reuse in other Working Interoperability contexts, including those involving other trading partners.

The following scenario constitutes the essence of the ECCF problem statement:

• Group A (for example, Standards Developing Organization (SDO) or Organization A) develops Specification X. If Group A is an SDO, it publishes Specification X as Standard X.

• Development organizations C and D want to implement Standard X.

• Group B (for example, SDO or Organization B) has previously developed Specification/Standard Y.

• Group A wants to say, “Specification X uses Specification Y to satisfy requirement R.”

• Both Group A and Group B would like to enable a trusted third party to validate either or both of the following statements with a high degree of certainty:

  1. Specification Y is correctly referenced and used by Specification X.

  2. Specification X is correctly implemented by Implementations I1 (by C) and I2 (by D).

Disambiguating either of these two statements (1 and 2) requires non-ambiguous answers for the following two questions:

• What does correctly referenced and used mean?

• What does correctly implemented mean?

Answers to these two questions lead to the question of how one can explicitly specify correctness criteria. The discussion also raises the question of whether the goal is to attain automated certification of correctness or whether one can restrict the certification process to human or human-assisted.

In the world of HL7, other SDOs, and national-level healthcare IT programs, the maximum amount of automated certification is the desired goal, simply because of the required scale of certification efforts. Such a goal leads to the virtually inescapable conclusion:

If explicit definitions and processes are not established using, for example, a framework like the ECCF SS, implicit assumptions will not be made explicit until implementation time or runtime.

When those definitions and processes are implicit, their explicitness is often determined in inconsistent ways. For example, they might be determined through decisions made by developers and database designers who are not qualified to make those decisions. As a result, the original assumption is interpreted in ways that produce inconsistencies between software interfaces purporting to be focused on similar problems. As a result, achieving Working Interoperability is often an expensive, time-consuming, one-off effort.

The ECCF focuses on making assumptions explicit in a traceable, layered manner, as explained in the remainder of this document.

References: The "ECCF Foundational Concepts" topic defines each of the concepts that collectively define the organizational and content semantics of the ECCF. The "Completeness of a Specification
Stack Instance" topic follows with a discussion of the implications of the “maturity” (completeness) 
of a SS in a given WI context.

As evidenced by the moniker ECCF, the core concepts of conformance and compliance are central to 
this discussion. The ECCF and Enterprise Architecture Specifications (EAS) topic discusses in detail 
the relationship between the ECCF and an EAS.

Related information

- Section 1.4.4. ECCF Foundational Concepts
- Section 1.4.2. ECCF: A Template for Working Interoperability
- Section 1.4.6. ECCF and Enterprise Architecture Specifications
- Section 1.4.4.2. ECCF Specification Stack
- Section 1.4.5. Completeness of a Specification Stack Instance

1.4.4. ECCF Foundational Concepts

As an understanding of the Enterprise Conformance and Compliance Framework (ECCF) 
foundational concepts is essential to use effectively the ECCF in the context of enabling Working 
Interoperability.

The following are the ECCF foundational concepts:

- Specification stack

- Conformance: statements, assertions, and certification

- Compliance

- Consistency

- Traceability

- Compatibility

- Localization

- Jurisdiction (discussed in greater detail in Governance Framework documentation)

- Provenance (discussed in greater detail in Governance Framework documentation)

Each of these concepts defines one or more aspects of the content and content interrelationships 
within an instance of the ECCF specification stack. A definition for each of these concepts is provided 
in the subsections below.

The ISO has addressed several of the concepts in the standards that specifically pertain to the broad 
topic that they call conformity. This section concludes with a comparison of the ECCF foundational 
concepts with terms defined by the Implementation and Conformance Work Group (IC WG) of HL7. 
The goal of the comparison is to attain harmonization to a single set of terms for use within the HL7 
community.

1.4.4.1. ECCF Reference Scenario

The following scenario provides a working reference example of the ECCF’s foundational concepts.

In particular, it focuses on the most salient concepts and constructs that are required to enable 
scalable, tractable Working Interoperability:

- Conformance statements: SDO S1 working in Jurisdiction A defines Specification A.

- Conformance statements: SDO S2 working in Jurisdiction B defines Specification B.
• **Compliance assertion:** SDO S2 applies a constraint pattern to Specification A for use within Specification B.

• **Conformance assertions:** Developers implement Specification A.

• **Conformance assertions:** Developers implement Specification B.

• **Conformance certification:** Third party evaluates Specification A and/or Specification B for correctness.

• **Compliance validation:** Specification A correctly references and uses Specification B.

References: Section 1.4.8.1 provides notes on the mapping of existing HL7 terminology to the ECCF.

Section 1.4.8.2 includes a discussion of the mapping of the ECCF foundational concepts to ISO. This section also discusses ISO terminology and standards relevant to the ECCF’s use of the terms *conformance* and *compliance*.

Related information

► Section 1.4.8.1. Harmonization and Mapping between ECCF and HL7 Implementation and Conformance Work Group

► Section 1.4.8.2. ISO Terminology: Conformance and Compliance

► Constraint pattern

### 1.4.4.2. ECCF Specification Stack

The specification stack (SS) is a 3-row by 4-column matrix that represents a collection of artifacts, which collectively and unambiguously defines the subject of the collection.

Understanding of a given SS’s subject enables potential users to judge whether it fits a particular need for either application or reuse within a new SS instance for a related subject.

The following figure shows a prototypic ECCF specification stack (SS) template with example artifacts in each cell.

**Note:** The specific SS instance for a given subject may have different artifacts. The example artifacts listed are representative of those used in specifying business-capability-level WI specifications. Thus, an SS instance has a scope defined by a coherent collection of functionality and content within a particular Working Interoperability (WI) context.

**Note:** Although the RM-ODP viewpoints are, by definition, non-hierarchical and non-orthogonal, the collections of artifacts within a single SS cell can be, and usually are, arranged in a hierarchy.

Structurally, the rows of the SS are levels of abstraction that map to the Object Management Group’s (OMG) Model-Driven Architecture (MDA) framework:

- Computationally Independent model (CIM)
- Platform-Independent model (PIM)
- Platform-Specific model (PSM)

The columns represent four of the five Reference Model for Open Distributed Processing (RM-ODP) viewpoints:

- Business/Enterprise
- Informational
- Computational
- Engineering
The fifth viewpoint is Technology, which represents a technology binding of an implementation to a specification. The discussion explains this viewpoint in more detail.

In summary, the 3-row by 4-column matrix forms the structure of the ECCF. The cells of an instance of an SS contain a layered set of related artifacts with content that is specific to the SS subject.

Figure 18. A prototype ECCF specification stack (SS) with example artifacts for a specific software component with a well-defined subject.

In the context of Working Interoperability, SS-specific conformance statements are especially important. These statements are contained in and expressed by the various artifacts of an SS instance.

As mentioned in the discussion of the ECCF problem statement, a given technology implementation of, or technology binding to a particular SS instance makes pair-wise conformance assertions that can be validated and certified. An independent third-party organization devoted to conformance certification activities usually certifies those conformance assertions. The organization certifies assertions as true or false, because both conformance statements and conformance assertions are Boolean-valued.

One or more constraint patterns often generate specific SS artifacts for a given cell, therefore associated with a given SS subject. These patterns are applied to existing artifacts from a previously defined specification that is now contextualized within the specific specification stack. The ECCF requires that these constrained specification artifacts be sorted by RM-ODP viewpoints and Model-Driven Architecture (MDA) levels of abstraction, thereby enabling the constrained artifacts to be used in a manner that is identical to the SS instance’s primary artifacts. There is no inherent difference between a new SS artifact and a reused/constrained artifact from an external SS instance.

References: See Externally-Developed Reference Specifications for a more detailed discussion on the inclusion of external specifications within an SS instance.

Related information

- Specification stack
- Section 1.4.5. Completeness of a Specification Stack Instance
1.4.4.3. Conformance

This topic describes the ECCF concept of conformance.

The ECCF has adopted the following definition of conformance and its associated specific terms from RM-ODP, as developed by the Australian National E-Heath Transition Authority (NEHTA):

Conformance relates a specific implementation of a specification irrespective of whether or not the specification is a "standard." Conformance is a quantitative assessment of how completely and accurately a given implementation fulfills the requirements stated in the specification. Conformance is evaluated at specific foci in the implementation. Conformance points as identified in the specification-of-interest. Evaluation is a Boolean statement, that is, it is either true or false.

As seen from the above definition, conformance denotes the validity of a given software component relative to a stated set of requirements. It confirms the correctness of a specific implementation or technology binding of a specific specification stack (SS) instance. The specification must express its requirements in the form of conformance statements. A specific implementation then makes pair-wise conformance assertions that may be evaluated—either through human or (preferably) automated testing—to be true or false.

Specifically, if a given conformance assertion is verified as true, the software component is said to be conformant, conforming to, or in conformance with pairs of conformance statements. Clearly, this means that the notion of conformance is certified at a granular level equal to that of the conformance statement.

Based on the preceding discussion, the ECCF clearly enables—but certainly does not guarantee—automated conformance evaluation and certification of a given specification instance that is bound to an implemented technology. Automated testing is certainly possible, however, if the conformance statements of the specification stack instance are correctly expressed at the Platform-Specific Model (PSM) level of the SS.

Note: The relevancy of PSM-level statements substantially increases when some of the conformance statements are traceable derivatives from conformance statements at the Platform-Independent Model (PIM) and Computationally Independent Model (CIM) levels of the SS instance.

The more mature the SS instance, the greater the conformance value.

Note: The term mature refers to the “completeness” of a given SS instance; i.e., the degree to which the maximum number of possible explicit assumptions and conformance statements have been made across the cells of the SS. This is represented by the collection of SS cells that are populated with the appropriate artifacts.

A discussion of conformance using statements that exist only at the CIM or PIM levels of the ECCF is often useful from the perspective of Working Interoperability. The existence of conformance statements at these levels indicates the presence and associated rigor of explicit assumptions, even if a PSM has not yet been specified. Once an implementation has been built, however, conformance statements made at the CIM or PIM levels can usually be fully evaluated only through a combination of human-only or human-and-automated testing. (The one-to-many relationships established going from the CIM to the PIM to the PSM levels causes this need for human input.)

Reference: See Section 1.4.8.2 for a discussion of recent ISO terminology regarding conformance, compliance, and certification.
**1.4.4.3.1. Example: Conformance in a Specification Stack**

The following figure is an example of a testable and verifiable conformance statement to which a given software component would claim conformance by a conformance assertion.

Operation X <MUST> bind to the RIM attribute Observation.value with an ISO 21090 data type of <CD> and a value set specified in LOINC as ...

*Figure 19. Conformance in a specification stack. Ideally, conformance certification is done through automated means, which implies that conformance statements must be expressed in a computable way.*

Blue arrows show conformance statements for a given SS.

Red arrows show conformance assertions for a given technology binding.

Gray stripes represent localization.

CIM = Computationally Independent Model (Conceptual)

PIM = Platform-Independent Model (Logical)

PSM = Platform-Specific Model (Implementable)
Conformance statements are included in the artifacts collected within a specification stack instance. Artifacts are distributed across the various viewpoints (columns) and through the levels of abstraction for the models (rows). An implementation (also called a technology binding) using a technology platform makes pair-wise conformance assertions against the conformance statements, which enables conformance certification to the specification.

1.4.4.4. Compliance

This topic discusses the ECCF concept of compliance. Several of the remaining foundational concepts are used to describe horizontal navigation (usually within a given SS cell) and vertical intra- or inter-cellular navigation in a specification stack (SS). The most important of these concepts is compliance.

Like the term conformance, the definition of compliance comes from the NEHTA project:

One standard or specification is compliant with another standard or specification if all propositions true in the initial standard are also true in the complying standard.

The term compliance is also used to state expectations as to how certain specifications need to satisfy possible legislative or regulatory constraints or requirements.

It is possible to develop new specifications with no compliance to existing standards or specifications. However, this is not the desired outcome. Existing standards or specifications should be referenced and adopted wherever possible to allow maximal potential for interoperability. Where no standard is chosen, there is little chance of two independent specifications sharing common approaches and thus enabling the use of common infrastructure.

Note: This definition defines compliance only for reusing and recontextualizing a previously developed standard/specification.

Compliance is also used in the context of artifacts derived from other artifacts by traversal of successive levels of abstraction or other hierarchy travels. The following formal definition(10) of compliance captures both the reuse and evolution aspects of the term:

A target/derived artifact (or artifact component, e.g., conformance statement) is compliant with its associated source artifact IF all conformant implementations of the target are also conformant with the source.

Thus, compliance is a measure of the correctness of a transformation. In the context of reuse, the transformation is a constraint pattern. This is because extensions to the source artifact normally introduce the requirement for additional conformance statements and are therefore not pure compliance issues. In contrast, in the context of evolution, the transformation is most often an increase in granularity or a change in representational vocabulary, such as Computationally Independent Model (CIM) -> Platform-Independent Model (PIM) -> Platform-Specific Model (PSM).

The following is a restatement of that definition:

Given an existing source artifact (specification, standard, etc.), a target artifact is said to be compliant with the source artifact if it has been derived from the source.

(10) From RM-ODP standard.
Thus, the statement "HL7 XML is compliant with W3C XML" can be restated as:

*All valid HL7 XML message instances are also conformant to W3C XML.*

Here is another example with a slightly different perspective:

*Web Services security specifications must be compliant with Web Service messaging (SOAP) if a security application is to support communication using SOAP.*

**References:** See also the discussion of consistency, traceability, compatibility, and provenance.

See Section 1.4.8.2 for a discussion of the ISO’s recent adoption of the general term *conformity assessment*. This term includes both formal notions of conformance certification and compliance validation.

**Related information**

► Section 1.4.4.4.1. Example: Compliance in a Specification Stack

► NEHTA Interoperability Framework

► Section 1.4.4.13. Comparison of Conformance and Compliance Lexicons: ECCF vs. HL7 Implementation and Conformance Work Group

► Section 1.4.8.2. ISO Terminology: Conformance and Compliance

► Section 1.4.5. Completeness of a Specification Stack Instance

1.4.4.4.1. Example: Compliance in a Specification Stack

Compliance occurs as both vertical and horizontal navigation within a specification stack (SS) instance.

*Figure 20. Compliance in a specification stack.*

Conformance statements are included in the artifacts collected within a specification stack instance. Artifacts are distributed across the various viewpoints (columns) and through the levels of abstraction for the models (rows). An implementation (also called a technology binding) using a technology platform makes pair-wise conformance assertions against the conformance statements, which enables conformance certification to the specification.
A transformation from a source to a target can apply to either vertical or horizontal SS navigation. In the most common cases, vertical navigation occurs as existing artifacts and their associated conformance statements move through the MDA levels of abstraction. In contrast, horizontal (intra-cellular) compliance most often involves embedding an external specification artifact within an SS instance).

**Note 1:** The importance of reuse and compliance cannot be overstated. Using artifacts created within a separate specification stack or as an independent standard or specification is, or at least should be, standard practice. As stated above, the norm for reuse is the application of a constraint pattern, because extending the original specification provides no guaranteed way of ensuring compliance with the original source specification.

**Note 2:** Certification (or more specifically, validation) of compliance is a common focus of internal architecture governance teams that are tasked with validating design consistency and traceability of requirements across multiple projects. In contrast to certification of conformance, certification of compliance is routinely done in a mixed semi-qualitative and quantitative visual inspection process.

**Related information**

- Section 1.4.5.3. Incorporating Reference Specifications in a Specification Stack Instance

### 1.4.4.5. Certification

This topic discusses the ECCF concept of certification.

Based on the discussion in the Section 1.4.4.4 topic, the following definition of conformance certification can be stated:

> ... a validation of trust, usually performed by a third party, which states that there has been a quantitative verification that a conformance assertion made by a technology binding and implementation correctly implements a specific conformance statement made in a given instance of a specification stack. In other words, a given conformance assertion is certified as true for a specific implementation/technology binding. By definition, the conformance certification is granulated to the conformance statement level.
However, the restricted usage of the term does not mean that “certification-like” activities do not occur for constructs such as compliance, consistency, or traceability. Rather, it means that formal, third-party certification activities are restricted to conformance.

Although the Section 1.4.4.3 topic discussed the concept of certification in the context of both conformance and compliance, the term deserves additional comments. Grammatically, the term certification can be modified with one or more adjectives to describe a particular process such as conformance certification. This leads to the question as to whether other types of certification such as compliance certification and consistency certification actually exist.

Although other types of certification are theoretically possible, SAEAF takes the position that the term certification, as commonly applied in the domains of HL7, is restricted to use with conformance certification only in the context of an implantation and its binding to a given specification by a set of pair-wise conformance assertions. SAEAF does not use the term compliance certification; instead, it uses compliance validation, as discussed in the "ISO Terminology: Conformance and Compliance" and "Compliance Validation" topics.

One of the main reasons that compliance is rarely certified formally is that the details of all the legal transformations from a given source specification to candidate target specifications may not be explicitly represented. This fact results in the situation where an invalid transformation is not discovered until the conformance certification process occurs.

Reference: See Section 1.4.8.2 for a discussion of the ISO’s recent adoption of the general term conformity assessment. This term collectively includes both formal conformance certification and compliance validation.

Related information

► Section 1.4.8.2. ISO Terminology: Conformance and Compliance
► Section 1.4.5. Completeness of a Specification Stack Instance
► Section 1.4.4.3. Conformance
► Section 1.4.5.2.3. Conformance Certification: Mature vs. Immature Specification Stacks

1.4.4.5.1. Compliance Validation

This example shows how HL7 or a technology implementation might perform a compliance validation.

The HL7 eXtensible Markup Language (XML) ITS must comply with the World Wide Web Consortium (W3C’s) XML specification. This specification is validated using one of two approaches:

• Within HL7 as part of the larger balloting and specification review processes; or

• By a specific technology implementation (such as an XML parser) operating on test HL7 input. In this case, the conformance certification of the parser includes the implicit compliance certification of the transformation of W3C XML specification to HL7 XML ITS. Note, however, that because compliance is being tested against a specific parser implementation, this is a conformance certification. A better example might be a human review of a given localization transformation, an action that would qualify as a certification of compliance.

In summary, by formalizing the notions of conformance, compliance, and conformance certification, the ECCF provides a multidimensional framework and vocabulary for documenting explicit assumptions on three levels of abstraction:

• Enabling formal conformance testing and certification.

• Facilitating the integration of specifications or standards developed outside of the SS per se.

• Enabling formal compliance validation as part of architecture best practices.
1.4.4.6. Consistency

This topic discusses the ECCF concept of *consistency*.

*Consistency* is a characterization of the logical coherence of the artifacts that are collected in a particular instance of a specification stack. Consistency is normally assessed on a row-by-row basis.

Consistency answers the following question:

*Do the artifacts in a given row of an SS instance—artifacts that are, by definition, sorted by RM-ODP viewpoints—have logical consistency in terms of their cross references and contextual dependencies and reuse?*

For example, a Unified Modeling Language (UML) activity diagram in the Business viewpoint that references static data constructs (for example, documents or data structures) passed between activities should have the relevant static constructs detailed in artifacts. Those artifacts are shown in one or more cells in the Informational viewpoint column of the same SS. This might also be true for a diagram representing the Computational viewpoint, depending on the focus and level of abstraction.

A somewhat more complex, but equally correct, example of consistency within a given SS requires that the Computational viewpoint of the PIM layer contain a direct (or traceable) realization of the business behavior depicted in the activity diagram. The Activity Diagram resides in the Business viewpoint, which uses the static components defined by the Information viewpoint.

Unlike conformance, consistency is neither formally defined nor certified, mostly because it is based on the notion of *logical coherence*. The HL7 ArB and other organizations adopting SAEAF and the ECCF will develop artifact-specific consistency metrics. The specification developers working on SS instances, or the organizational governance body overseeing development efforts, will need to ensure that the content of a given SS is logically coherent, consistent, and complete.

The graphic in the following link illustrates the concepts of consistency, traceability, and compatibility in a specification stack instance.

### Related information

► Section 1.4.5. Completeness of a Specification Stack Instance

1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack

This figure illustrates the concepts of consistency, traceability, and compatibility in a specification stack.

*Figure 21. Consistency, traceability, and compatibility in a specification stack instance.*

Conformance statements are included in the artifacts collected within a specification stack instance. Artifacts are distributed across the various viewpoints (columns) and through the levels of abstraction for the models (rows). An implementation (also called a technology binding) using a technology platform makes pair-wise conformance assertions against the conformance statements, which enables conformance certification to the specification.
1.4.4.7. Traceability

This topic discusses the ECCF concept of traceability.

In common parlance, the term traceability often refers to the ability to link code to requirements. However, because traceability often conflates the navigational and logical linkage notions of conformance and compliance in a collection of artifacts, the HL7 ArB felt that it needed to define the concept formally.

In the context of a given SS, traceability means:

System capabilities explicit in a software component that can be traced down from a CIM-level statement to the PIM-level, followed by a trace to the PSM-level, followed by a trace to an implementation-specific capability. Traceability may apply to capabilities stated as conformance statements.

For example, you could trace the following conformance statement from the PIM and PSM levels to the technology implementation level:

The system should support the semantics-complete HL7 Reference Information Model (RIM) Person and Healthcare Provider classes.

In operational usage, the concept of traceability is particularly valuable in ensuring that the transformations that are applied to third-party specification in a particular SS are correctly integrated into the SS. Verification of traceability is closely linked to the concept of provenance and thus to overall governance.

Reference: For more information about provenance and governance, see the SAEAF Governance (GF) document.

Related information

Specification stack graphic

► Section 1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack

► Section 1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack

► Section 1.4.5.2.2. Traceability of Conformance Statements in an Immature Specification Stack Instance

► Section 1.4.4.11. Provenance
1.4.4.8. Compatibility

This topic discusses the ECCF concept of compatibility.

Compatibility is a relationship between two or more conformance statements involving two or more specification stack instances. The relationship identifies whether two or more implementations certified to be conformant to the specification stack instances can achieve WI without further transformations. If so, the two SS instances and associated implementations are called compatible.

Two implementations are compatible if they do not specify contradictory constraints (i.e., contradictory/inconsistent localizations) to the conformance statements of given specification. A localization can thus be viewed as a consciously applied constraint.

Because different compliant specifications can be drawn from the same specification at one of the levels, a one-to-many relationship between specifications in the same SS subject exists from one level to the next. A technology component could conform to the specification that took one path without being compatible with a technology component that conformed to a specification taking a different path.

In contrast, the two implementations may require additional transformations to achieve WI because of localizations or other applied constraints. In such a case, the implementations are considered incompatible. For example, conformant implementations might be incompatible because of different localizations on data types or restrictions to value-set lists.

In ECCF terms, a situation may arise where two conformant implementations cannot achieve WI without further modification in the presence of documented incompatibility. This situation would be analyzed by examining the source of the incompatibility as manifested in one or more conformance statements and their associated localizations as manifested in each implementation’s pair-wise conformance assertions. Appropriate adapters and transformations or corrections could then enable WI (to make the two SS instances compatible).

Related information:

- Specification stack graphic:
  - Section 1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack

1.4.4.9. Localization

This topic discusses the ECCF concept of localization.

Localization indicates custom modifications or other alterations (including ignoring) to specific conformance statements in a local context. This results in non-compliance to the overarching Enterprise Architecture Specification (EAS). Specific localization semantics are identified in the conformance and compliance assessment process.

Localization represents the real world of specification application. A given party such as an organization or development team often applies one or more localizations to a given specification to make it specifically and exactly contextually relevant to its stakeholder base. Localizations are not the one-off enemy of WI; rather, they are necessary and manageable by virtue of the ECCF’s framework, which facilitates the expression of a given localization as an explicit assumption being made by a
As such, all localization must be in compliance with and traceable to the artifacts that are being localized.

As shown in the specification stack (SS) graphics, the SS contains a localization strip that is instantiated three times in each SS graphic. The localization strip indicates where localizations can occur, showing the possibility of the global application of localization protocols in the following locations:

- Between the CIM and PIM rows.
- Between the PIM and PSM rows.
- Between the PSM row and an implementation.

Note: The localization strip indicates that localization can occur for any artifact above or below the strip.

Localizations can be applied to any aspect of any artifact throughout a specification stack instance, as required. In addition, localizations can be applied using a particular implementation/technology binding. Intra-cellular or reused compliance transformations that apply constraint patterns should be viewed as instances of localization.

**Related information**

*Specification stack graphics:*

- Section 1.4.4.3.1. Example: Conformance in a Specification Stack
- Section 1.4.4.4.1. Example: Compliance in a Specification Stack
- Section 1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack
- Section 1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack

**1.4.4.10. Jurisdiction**

This topic discusses the ECCF and Governance concept of *jurisdiction*.

*Jurisdiction* defines the *boundary* and *scope of authority* of a person, group, or organization. Jurisdiction may refer to a purely geographical boundary such as government, or a more complex domain of influence, such as a business or Standards Developing Organization (SDO).

By definition, the artifacts that populate a particular instance of an SS lie within one or more jurisdictions, even if that jurisdiction is only implicitly (rather than explicitly) expressed or stated. Explicit understanding of jurisdictional requirements and influences are particularly critical in establishing comprehensive, scalable, and extensible Working Interoperability.

**Reference:** For more information about jurisdiction, see the Governance Framework (GF) information.

**1.4.4.11. Provenance**

This topic discusses the ECCF and Governance concept of *provenance*.

*Provenance* is formally defined as “documentation that identifies the traceability of the history of a given artifact within a given specification, from its origination (for example, as a requirement) through its implementation.” In other words, provenance is the documentation of the source of all constraint statements in the specification.

As such, provenance is closely linked to compliance. The documentation required to support provenance may or may not be included as part of the specification. If it is not included, a formal external reference pointer is required. Provenance documentation also includes references to externally developed standards and specifications. Finally, provenance can be viewed as another face
of traceability in the sense that traceability is an instance-level construct, whereas provenance is a
collection-level construct for an entire specification.

Reference: For more information about provenance, see the Governance Framework (GF)
information.

1.4.4.12. Technology: Viewpoint, Bindings, and Implementations

Although not formally included in the list of ECCF foundational concepts, the usage of the terms
technology and implementation need to be formally understood in the context of the ECCF.

Technology (or, more properly, a technology component) is defined in RM-ODP as follows:

A logical unit that conforms to one or more conformance statements, as stated in a
specification or standard.

In turn, an implementation is defined as follows:

A deployable collection of technology components which employ one or more
specific technologies (e.g., Eclipse platform, Oracle Fusionware, etc.) to realize the
details of a given specification. An implementation is thus a concrete, physical
manifestation—a “technology binding”—that conforms to a set of conformance
statements made by a given specification and collected in an associated
specification stack. The conformance of an implementation to a specification may
be formally evaluated by determining the truth value of the implementation’s pair-
wise conformance assertions.

Finally, the Technology viewpoint represents the notion of a technology binding to a given
specification, which itself is contextualized and collectively represented through using the other RM-
ODP viewpoints:

The Technology viewpoint is the RM-ODP viewpoint whereby one or more concrete
implementations of a given specification are described and formally associated with
the other RM-ODP viewpoints for the specification of interest.

As stated in the description of the SAEAF ECCF, the overarching focus of SAEAF is to enable
Working Interoperability (WI). Interoperability is the ability of two parties, either human or machine,
to exchange data or information.

The difficulty of achieving WI for a given context involving specific software components becomes
more scalable, and predictably tractable, when the assumptions that underlay the specification can be
made explicit in an organized, layered fashion. The purpose of the ECCF specification stack is to
provide that organizational framework for enabling and collecting the explicit assumptions that
collectively define the specification’s meaning.

The enemy of Working Interoperability is the existence of implicit conformance statements. When
finally made explicit at the software component level in a concrete implementation, such statements
may manifest behavioral or informational semantics that are not consistent with the implicit
statements made elsewhere in the specification. The degree to which an SS contains all the relevant,
explicitly stated conformance statements against which a given implementation may make pair-wise
conformance assertions is a measure of the maturity of the SS instance. It is also the basis of the
underlying value proposition of the ECCF.

In summary, a technology binding to a specific SS represents a specific implementation of a set of
physical components that may be formally certified as conformant to the specification.
1.4.4.13. Comparison of Conformance and Compliance Lexicons: ECCF vs. HL7 Implementation and Conformance Work Group

Following the definitions of the ECCF’s foundational concepts, this section discusses those concepts in the historical context of HL7’s use of the terms conformance and compliance.

As the following analysis demonstrates, HL7 has used the terms conformance and compliance in an overlapping, often interchangeable manner that has resulted in a certain amount of ambiguity. The ECCF clarifies this ambiguity.

Moving forward, the terms used by HL7 will be aligned with SAEAF in general and the ECCF in particular.

Related information

Section 1.4.8.1.1. Comparison of ECCF and HL7 Implementation and Conformance Work Group Concepts

1.4.5. Completeness of a Specification Stack Instance

The basic anatomy of the ECCF specification stack has been defined and depicted as a three-row, four-column matrix. This topic shows a layered graphic of a specification stack instance.

As defined, the columns in the matrix serve as RM-ODP viewpoints and rows serve as Object Management Group (OMG) levels of abstraction from the Model-Driven Architecture (MDA) paradigm.

Figure 22. Prototype ECCF specification stack instance with an associated technology binding.

Green arrows represent implied compliance to other standards (asserted in isolation).

Black arrows represent conformance statements made at multiple levels of abstraction. The left to right flow indicates movement through the CIM, PIM, and PSM layers.

Red arrows represent validation of conformance statements.

Gray lines represent localization.

Consistency, compatibility, and traceability are not shown in this figure.
Compliance and traceability between the levels of the specification stack is documented via provenance. The ability to reuse (“reference”) a formal specification developed outside the context of a given specification stack, as shown by the Green specification stack placeholders for specifications developed by external organizations (for example, “SDO 2” and “SDO 3”), promotes collaboration, cooperation, and artifact reuse. An example of a natural alignment is the use of certain Computational/Behavioral semantics from the Electronic Health Record System Functional Model (EHRs-FM) in a separate specification stack instance (for example, Specimen Management).

Note: In the graphics of the specification stack table, the MDA layers are in the left column. However, in the layered graphics of the specification stack, the MDA layers are in the back row.

Reference: For a definition of provenance, see Section 1.4.4.11 and the Governance Framework (GF) information.

1.4.5.1. Applying the ECCF

Earlier topics described the anatomy and organization of a generic specification stack (SS), a meta-template from which you can define organization-specific templates. An organization that wants to use the ECCF must first define one or more templates that specify the artifacts—type, format, and content—that will populate specific instances of its various specification stack instances.

As is discussed in more detail in Section 1.4.6, organizations need not use identical specification stacks to achieve Working Interoperability (WI). However, when using different SS templates (or, operationally speaking, when two trading partners use different artifacts), mappings must be developed to resolve semantic (static or behavioral) differences.

The value of using the ECCF is that trading partners can focus their WI discussions around well-specified, non-ambiguous artifacts that surface explicit conformance statements at multiple levels of...
abstraction. This allows the WI discussions to focus on pair-wise conformance assertions made in the context of specific SS instances.

Examples of organizations currently working with the ECCF to define specific SS templates include:

- HL7
- Open Health Tools Architecture Project
- National Cancer Institute
- Canada Health Infoway
- Australia National Electronic Health Authority

Note: For HL7, the HL7 ArB has begun the process of identifying the specific artifacts that will be required to develop fully specified specification stack instances. The HL7 ArB anticipates that the SS templates for the three HL7 interoperability paradigms (IPs)—messages, documents, and services—will be identical (or virtually identical) for all three IPs in the CIM row of the SS. Differences—especially in the Computational viewpoint column—will appear in the PIM (row 2) and PSM (row 3) rows. These differences will manifest themselves in differences in technology bindings and implementations.

In addition to drawing on the collective artifact repertoire of HL7 (such as the Reference Information Model (RIM), Refined Message Information Models (RMIMs), HL7 Development Framework (HDF), Clinical Document Architecture (CDA), and Vocab) the HL7 ArB’s candidate artifacts for individual SS cells will be subject to review and revision from HL7 membership.

Related information

Section 1.4.6. ECCF and Enterprise Architecture Specifications

1.4.5.2. Mature and Immature Specification Stack Instances

The difference between a mature and immature specification stack (SS) instance is the degree to which the cells of the SS instance have been fully populated; that is, the degree to which the SS subject is fully specified by a complete set of explicit, testable conformance statements.

The first figure shows a mature SS instance and the second figure shows an immature SS instance.

In a mature SS instance, all the cells are populated with the appropriate artifacts.

Mature specifications are, in part, mature because they are comprehensively and relevantly compliant to other standards. Note that implementers will always have to define some things, such local governance when adopting a standard, preferably documented in a compliant specification. Mature SS instances simply make this process more scalable and tractable.

Figure 23. Fully mature specification stack instance

\[ X \]

Denotes artifacts that HL7 or another SDO (or relevant jurisdiction) has defined.

\[ O \]

Denotes artifacts that an implementer must define. This graphic has no O’s; thus, this specification stack instance is mature.
In contrast, in an immature SS instance, not all of the cells of the SS matrix are populated with the appropriate artifacts. Therefore, the effort required to achieve WI is greater than with a mature SS, because implementers will have to translate (or assume) implicit conformance statements to make them explicit in the implementation. Thus, the specification stack is incomplete.

**Figure 24. Immature specification stack instance**

- **X** Denotes artifacts that HL7 or another SDO (or relevant jurisdiction) has defined.
- **O** Denotes artifacts that an implementer must define. Implicit conformance statements ultimately will be made explicit in the technology binding.

### 1.4.5.2.1. Traceability of Conformance Statements in a Mature Specification Stack Instance
One of the more important characteristics of a mature SS instance is that it has full traceability of conformance statements.

In the following figure, the arrows show that the conformance statements are traced through each of the layers and viewpoints, and the conformance at the Platform-Specific Model (PSM) layer.

**Figure 25. Conformance statements are fully traceable in a mature specification stack instance.**

The red arrows refer to the conformance statements in the Technology viewpoint.

The black arrows refer to the conformance statements in the specification stack.

The gray stripes represent localization.

Conformance statements are included in the artifacts collected within a specification stack instance. Artifacts are distributed across the various viewpoints (columns) and through the levels of abstraction for the models (rows). An implementation (also called a technology binding) using a technology platform makes pair-wise conformance assertions against the conformance statements, which enables conformance certification to the specification.

### 1.4.5.2.2. Traceability of Conformance Statements in an Immature Specification Stack Instance

In an immature SS instance, full traceability of conformance statements is NOT supported. This results in gaps in explicit understanding of the specification.

As this figure shows, an immature SS instance has less built-in consistency regarding standardization. Therefore, greater effort is required to achieve working interoperability.
1.4.5.2.3. Conformance Certification: Mature vs. Immature Specification Stacks

Conformance certification is a more straightforward process for a mature specification, which is fully specified with explicit conformance statements.

The advantage of a mature specification is the ease of conformance, because implementation teams do not have to translate (or infer) implicit conformance statements into explicit code. In addition, the fact that a fully specified specification stack (SS) ensures traceability means that the validity of any compliance transformations and constraints that were applied during the development of the SS can be quantitatively assessed as part of the conformance certification process.

Note that the Platform-Specific Model (PSM) conformance statements are directly linked to conformance statements in the higher rows of the SS, which makes the conformance validation of a technology binding and its implementation a robust and reliable activity. As a result, considerable automated conformance certification will mostly likely occur in the presence of mature SS instances for well-defined software components.

In contrast, conformance certification and Working Interoperability are more difficult to achieve when implementers and certifiers are working with immature SS instances. By definition, these instances contain a number of implicit constraint statements. These statements ultimately need to be made explicit, often by implementers who lack the problem domain expertise or context to make the required judgments. Certification of technology that implements an immature specification takes into account only the explicit semantics of integration that have been complied with through the constraint pattern on the specification stack.
Consequently, implementation teams must make assumptions about the missing conformance
statements such as those from the Platform-Independent Model (PIM) and PSM levels, so that they
can be explicitly realized in their technology implementation.

A given conformance statement at the Computationally Independent Model (CIM) level of a SS often
has a one-to-many relationship with conformance statements in the PIM and PSM rows of a SS
instance. For this reason, the process of reverse engineering a given, lower-level conformance
statement is fraught with the potential for considerable work at best and overt error at worst. In the
presence of an immature specification stack instance, the one statement that can be made with
confidence is that the implementers know what is explicit and what is not.

1.4.5.3. Incorporating Reference Specifications in a Specification Stack
Instance

As discussed in this document, the ECCF provides a framework to a given organization for
integrating reference standards and specifications developed externally.

1.4.5.3.1. HL7 Reference Specifications Example

In particular, other Standards Developing Organizations (SDOs) or organizations with significant
operational jurisdiction (for example, government programs) may also produce and require reference
specifications and materials.

From an HL7 perspective, a given reference artifact can be used, applied, referenced, transformed, or
otherwise leveraged by any cell of the specification stack (within the same viewpoint). However,
referenced specifications need not reside in a particular layer of the specification stack. Rather, they
are most often viewed as surrounding or providing input to instances of the SS at any level of
abstraction (row) or column. Note that externally referenced specifications may or may not have
preexisting technology bindings.

The following figure provides an example of a reference specification.

Figure 27. A tabular sorting of several existing HL7 artifacts using the RM-ODP viewpoints.

1.4.5.3.2. Externally-Developed Reference Specifications

Assume that externally developed reference specifications are specified using ECCF-compliant
artifacts.

The following figures present the relevant aspects of the incorporation of externally developed
specifications. The specification stack (SS) instance of a new specification uses and references
externally developed specifications.

Figure 28. Externally developed reference specifications are assumed to be specified using ECCF-
compliant artifacts.
Although an external specification may be reused in a single cell, it is often used across a new SS instance.

Figure 29. Externally developed reference specifications may be included either piecemeal within a given cell or included in total and, therefore, span more than one cell of an ECCF SS instance.
1.4.5.4. ECCF Specification Stack Diagrams

For your reference, this section shows some of the key ECCF specification stack diagrams that are discussed elsewhere in this document.

1.4.5.4.1. Specification Stack Template

The specification stack template lists the example artifacts for a software component.

Figure 30. A prototype ECCF specification stack (SS) with example artifacts for a specific software component with a well-defined subject.
### Related information

- **Section 1.4.4.6.1. Consistency, Traceability, and Compatibility in a Specification Stack**

### 1.4.5.4.2. The Full ECCF Specification Stack

This diagram shows the an ECCF specification stack instance with a couple of external reference specifications.

**Figure 31. Prototype ECCF specification stack instance with an associated technology binding.**

- Green arrows represent implied compliance to other standards (asserted in isolation).
- Black arrows represent conformance statements made at multiple levels of abstraction. The left to right flow indicates movement through the CIM, PIM, and PSM layers.
- Red arrows represent validation of conformance statements.
- Gray lines represent localization.

Consistency, compatibility, and traceability are not shown in this figure.
1.4.6. ECCF and Enterprise Architecture Specifications

As stated in the SAEAF Introduction and Overview, the January 2008 mandate from the HL7 Chief Technology Officer (CTO) to the HL7 Architecture Board (ArB) was to “develop an enterprise architecture specification (EAS) for HL7.” Instead, the HL7 ArB’s response to that mandate was to develop SAEAF as the framework for developing an EAS. The triptych in the following figure shows the relationship between an Enterprise Architecture Framework, the specifications that that framework enables, and the implementation of those specifications.

Figure 32. The relationship between an enterprise architecture framework, specification, and implementation. Reading L -> R (left to right), the cardinalities of the relationship start at 1 -> * (start at one on the left and imply multiples to the right).
This approach is based on the HL7 ArB’s overarching belief that an EAS must emerge from both the bottom-up and the middle-out efforts rather than from a top-down specification. Conversely, in the absence of a framework in which that work can be structured, categorized, and shared—a framework which itself must be specified top-down—bottom-up or middle-out efforts to develop an EAS are inefficient, ineffective at best, and overtly unsuccessful at worst.

Thus, the HL7 ArB applied the collective experiences, perspectives, and requirements of the Jump Start participants to define and assemble SAEAF during the summer of 2008. Since then, the HL7 ArB has accepted input into the overall content of SAEAF and expects that there will be further contributions to the SAEAF content and mapping of SAEAF to other projects. As SAEAF is applied both within and outside of HL7, the various Enterprise Architecture Specifications of the organizations using SAEAF will emerge. By virtue of each of the organizations using a common framework, however, scalable and tractable WI will be a reality—not only within each enterprise, but also between enterprises. As the scale of WI increases with the requirements of national and trans-national (for example, European Union) healthcare IT infrastructures, the availability and application of SAEAF becomes increasingly important.

### Related information

- Section 1.3.6.1. How SAEAF Got Started
- Section 1.3.6.2. Jump Start Project
- Section 1.3.3.3. HL7 Foundation Components

<table>
<thead>
<tr>
<th>SAIF</th>
<th>EAS</th>
<th>EA Instances/Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behavior Framework (BF)</td>
<td>Zachman, TOGAF, RM ODP, Krutchens 4+1</td>
<td>CTS 2</td>
</tr>
<tr>
<td>Information Framework (IF)</td>
<td>RIM, BRIDG, other DAMs</td>
<td>PASS</td>
</tr>
<tr>
<td>Enterprise Conformance &amp; Compliance Framework (ECFF)</td>
<td>Vocabulary</td>
<td>EA-IP</td>
</tr>
<tr>
<td>+</td>
<td>Bindings, Semantic Infrastructure</td>
<td>NCI CBIIT examples:</td>
</tr>
<tr>
<td>Governance Framework (GF)</td>
<td>Data Types</td>
<td>COPPA</td>
</tr>
<tr>
<td>+</td>
<td>CDA/CSP</td>
<td>SMS</td>
</tr>
<tr>
<td>EAF Snapshot / Explanation</td>
<td>EA Patterns</td>
<td>caEHR</td>
</tr>
<tr>
<td>&quot;SAIF book&quot;</td>
<td>EA Best Practices</td>
<td>Transcend</td>
</tr>
</tbody>
</table>
1.4.6.1. The ECCF as the Skeleton of an EAS

Although a robust, scalable, supportable, extensible, and implementable Enterprise Architecture Specification (EAS) requires all of the critical components specified in SAEAF, the ECCF in general—and specification stack instances in particular—supply the skeleton of an EAS.

An EAS is defined as follows:

An Enterprise Architecture Specification (EAS) consists of a set of component specifications, each focused on a specific business-appropriate subject, which are produced through bottom-up and/or middle-out, project-based activities within an overarching top-down-specified Enterprise Architecture Framework. The component specifications are combined with a set of architecture best practices and patterns and are defined, maintained, and evolved through the collective notions of conformance and governance, and internal consistency across various enterprise perspectives and viewpoints.

When reflected against the anatomy and content of the ECCF specification stack, the analogy of the ECCF as the skeleton or source of atomic constructor units of an EAS emerges.

Related information

► Section 1.4.4.2. ECCF Specification Stack
► Section 1.3.3.2.3. Enterprise Architecture Definitions

1.4.7. The Value of an ECCF

In summary, the ECCF is somewhat similar to measurement systems in the building trades, such as a plumb line, level, or ruler.

The enemy of Working Interoperability (WI) is implicit assumptions. As such, the value proposition of the ECCF can be stated in succinct terms:

The ECCF specification stack promotes scalable, tractable Working Interoperability by supporting the publication and sharing of the critical elements of a specification for a software component or other system, as detailed in one or more specification stack instances. These elements effect the attainment of WI through the layered, explicit representation of testable conformance statements.

In some cases, an immature standard (that is, one or more SS instances with insufficient content in one or more matrix cells) is the only specification that can be published. The practical realities of WI are that the more complete (mature) the SS, the easier it is to achieve WI. Regardless, the publishing of an immature SS instance is better than no SS instance.

In fact, achieving WI in this context will involve the filling in by one or both trading partners of the incomplete cells of the SS in two ways:

• By generating explicit artifacts that move the immature SS instance toward its becoming a mature instance; or

• By working out at the code interface, specific adapters that take the place of the explicit artifacts are not completed.

Unfortunately, the process of “filling in” an immature SS through one-off, code-level adapters often hard wires a particular implementation in a brittle, non-scalable manner that impedes extensible,
scalable WI. The absence of traceability caused by the absence of a fully completed SS makes the task
of multiple interoperable implementations considerably more difficult.

As stated earlier in the Section 1.4.2 topic:

*The greater the distance on the Stairway to Heaven, the more difficult the transforms required to achieve WI.*

Note: There may be cases where the appropriate transformations and adapters simply cannot be
developed after the fact for use in a given WI context. This situation occurs if the explicit semantics
required on one side of the WI context are not available on the other side. In the context where
specifications are developed using ECCF SS templates, such instances of non-obtainable WI can be
detected sooner rather than later in the development process so that appropriate assessments can be
made as to the cost and value proposition for course corrections.

With that background, the following list summarizes the ECCF value proposition. The ECCF provides
the following:

• A *structured* way to certify conformance.

• A *well-defined metric* to evaluate compliance (and certify it if necessary).

• A *framework for integrating* externally developed specifications or standards. In particular, an
organization adopting an ECCF can develop specific specifications/standards that are compliant
with other specifications/standards based on published SSs, because the SS explicitly states
important business rules, information constructors, and behavioral contracts. A set of SS
instances does the following:

  1. Enables WI stakeholders to quantitatively assess systems or components to determine
whether they are explicitly conformant with the identified standards and therefore compliant
to the other interoperability standards referenced in a given specification stack.

  2. Provides a framework for defining an EAS using an agreed-upon set of measurement
standards.

The result of applying the ECCF (and its associations with the Governance, Information, and
Behavioral Frameworks) in an intra-enterprise or inter-enterprise context is a set of inter-related
specification stack instances. These instances enable WI and, in combination with an enterprise-
endorsed set of patterns, procedures, and best practices, collectively define an Enterprise Architecture
Specification (EAS).

1.4.8. ECCF Appendixes

These appendixes contain information on the historical usage of the terms *conformance* and
*compliance* by other groups, such as the HL7’s Implementation and Conformance Work Group and
ISO.

1.4.8.1. Harmonization and Mapping between ECCF and HL7 Implementation
and Conformance Work Group

As mentioned at the conclusion of the ECCF Foundational Concepts section, the historical usage of
the terms *conformance* and *compliance* by HL7 does not formally align completely with SAEAF’s
use of the terms in the ECCF.

For example, in HL7 V2.x (x < 5), HL7 spoke of “compliance” with respect to the correct usage of
the message segment header (MSH) as the leading segment in an HL7 message. Beginning with HL7
V2.5 and continuing to the present, the term *conformance* has been used by HL7 to indicate the notion
of testing the ability of a given message generator to generate a specification-compliant message
instance. An eXtensible Markup Language (XML) parser that validates the message instance against
the specified XML schema definition (XSD) usually performs this test.

The term *compliance*, in turn, has been used to indicate certain structural characteristics of a given
message. For example, a message is compliant with the HL7 2.x specification if the first segment is an
MSH and meets the formal structure and content requirements of an MSH segment.

As discussed previously, SAEAF uses the terms in a broader context than the single interoperability
paradigm of messaging. (Interoperability paradigm is a formal HL7 term.) The ECCF has both
disambiguated and granulated several other associated concepts in the larger context of the ECCF
specification stack. As a result, the HL7 ArB and the HL7 SC WG began efforts in April 2009 to
formally reconcile and harmonize their conceptual frameworks so that HL7 can use a single, ECCF-
based lexicon when speaking about conformance, compliance, and other closely related concepts.

### 1.4.8.1.1. Comparison of ECCF and HL7 Implementation and Conformance Work Group Concepts

The following table and reference notes provide a concept-by-concept comparison between the ECCF
Foundation Concepts and those concepts used in the published proposal #605 (Conformance
Documentation Hierarchy) for HL7 2.8.

Readers familiar with document #605 will note that the ECCF addresses the same overall problem as
the HL7 2.8 Conformance document. However, some relevant differences exist in the usage,
specificity, and granularity of various terms between the two frameworks.

Note: The Implementation and Conformance Work Group (IC WG) has always intended that the 2.8
document would apply to 3.0 and, as such, no additional harmonization beyond the present activity
appears to be required to align the lexicon of the IC WG with the ECCF and SAEAF.

Table 1 presents nine relevant terms used by both the ECCF of the IC WG document. The second
column of Table 1 defines each term in language that is consistent with the ECCF. Terms in *bold
italics* indicate inconsistent use of a term between the two frameworks. As can be seen from
reviewing the table, the IC WG and the ECCF use terms associated with the notions of conformance
in an essentially compatible and consistent manner.

In contrast, disagreement exists on the use of the term *compliance*. Given that the ECCF’s use of the
term is grounded in ISO (although the most recent ISO documentation does not formally use (and has
deprecated the term, viewing it, instead, as a subtype of conformity), the HL7 ArB expects that the
IC WG’s documentation will adopt the ECCF’s definition and usage of the term.

<table>
<thead>
<tr>
<th>#</th>
<th>Definition</th>
<th>ECCF Concept</th>
<th>HL7 IC WG Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A property of an implementation (real-world working software), such that it follows a set of rules provided. The purpose of the rules is usually to increase the chances of interoperability between multiple separate implementations.</td>
<td>Conformance</td>
<td>Conformance</td>
</tr>
<tr>
<td>#</td>
<td>Definition</td>
<td>ECCF Concept</td>
<td>HL7 IC WG Concept</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>2</td>
<td>The different impacts achieved by implementation conformance, depending on how specific that the underlying rules are. The ECCF does not define specific levels other than as related to the row of the SS. The bulk of the HL7 v2.8 document specifies the conformance levels as related to how specific the rules are in the context of HL7 v2 profiles. It is clear that the IC WG has defined useful constructs around the term level. However, this is an area that will most likely require further discussion between the HL7 ArB and the IC WG to disambiguate, and possibly rename, the IC WG’s levels so as not to confuse them with OMG’s MDA levels, as represented in the SS.</td>
<td>Conformance level</td>
<td>Conformance level</td>
</tr>
<tr>
<td>3</td>
<td>A collection of conformance statements is created to promote Working Interoperability; that is, the set of statements to which an implementation is conformant.</td>
<td>Specification stack instance</td>
<td>Standard (ballot-level) profile (localized- or constrained-level)</td>
</tr>
<tr>
<td>4</td>
<td>A statement made by the owner, creator, or vendor of an implementation stating that such an implementation is conformant to a set of rules.</td>
<td>Conformance assertion</td>
<td>Conformance claim</td>
</tr>
<tr>
<td>5</td>
<td>The process is where independent verification is made as to whether the statement made by the owner or vendor regarding compliance is actually true or not. The certification process describes the outcome (Boolean True or False) of each conformance assertion and conformance statement evaluation.</td>
<td>Conformance certification process (ISO Conformity Evaluation)</td>
<td>Test</td>
</tr>
<tr>
<td>6</td>
<td>An official statement by a designated authority that tests have been done and a particular</td>
<td>Conformance certification result</td>
<td>Certification</td>
</tr>
<tr>
<td>#</td>
<td>Definition</td>
<td>ECCF Concept</td>
<td>HL7 IC WG Concept</td>
</tr>
<tr>
<td>---</td>
<td>------------</td>
<td>--------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>7</td>
<td>A property of a target rule (usually a conformance statement, but possibly a statement from a source specification or standard), stating that the target was derived from or refers to the source rule in such a way that implementations that are conformant to the target rule in question are also conformant to the source rule as well.</td>
<td>Compliance (inconsistent)</td>
<td>Conformance (inconsistent)</td>
</tr>
<tr>
<td>8</td>
<td>A claim is made that a rule is appropriately related to a preexisting rule. The notion of “rule” is restricted to conformance assertion.</td>
<td>Compliance assertion (inconsistent)</td>
<td>Conformance claim (inconsistent)</td>
</tr>
<tr>
<td>9</td>
<td>A relationship between two (or more) conformance statements that apply to implementations that are meant to interoperate with each other such that conformant implementations can, in fact achieve, this interoperability. This implies that they have consistent localizations. Thus, compatibility is specifically designed to identify potentially incompatible localizations, restrictions, constraints between two otherwise compatible specifications, standards, and profiles (able to achieve Working Interoperability).</td>
<td>Defined in ECCF based on IC WG’s definition.</td>
<td>Compatibility</td>
</tr>
</tbody>
</table>

The two views of conformance are essentially the same. The notion is that a particular implementation of a specification is valid or conformant to a particular set of pre-defined rules. The rules are most often contained in an HL7-defined specification or standard. In addition, both frameworks take the view that a claim that an implementation is conformant (usually made by a commercial entity in the context of a specific product), should be treated with a certain amount of distrust. This requires the validation of the conformance claim by an independent third party who can certify in some way that the claim of conformance made by the implementation against the specification or standard is, in fact, true. In addition, both frameworks contain the notion that “rules that apply in a particular specification or standard can be derived from preexisting rules.”
As noted in the above discussion, the concept of compliance in the ECCF is used in a broad sense that does not apply to implementations. The concept of compliance applies to inter- and intra-cell navigation within an SS instance and/or the reuse by a given SS instance of a specification (or parts of a specification) developed outside the boundaries of the specification of interest. Specifically, as previously discussed, the term compliance is used in both its usage contexts to describe the fact that a transformation from a source artifact to a derived target artifact has been done correctly.

Reference: See Section 1.4.8.2 for an explanation of how ISO currently defines the notion of conformity and how it maps to the ECCF foundational concepts of conformance, compliance, and certification.

1.4.8.1.2. Reference Notes for the SAEAF and HL7 2.8 Comparison

Following are additional notes about the SAEAF and HL7 2.8 comparison submitted by the HL7 ArB to the IC WG during the harmonization process.

These reference notes are based on Table 3.

1. The use of the term conformance is essentially identical between the two frameworks, although SAEAF only defines the term in the context of a single conformance statement and its implementation-specific conformance assertion. If the conformance assertion can be verified as true, the implementation is said to be conformant relative to that specific conformance statement and conformance assertion pair.

In the HL7 2.8 Conformance document, the notion of 7 levels of conformance are semi-quantitative assessments of the degree of difficulty involved in obtaining what SAEAF calls Working Interoperability (WI). SAEAF does not currently have a numerical system for assessing the degree of difficulty in obtaining WI, relying instead on the three MDA levels of abstraction as indicators and, in addition, more finely granulating the artifacts that collectively define a given specification.

In general, the use of the term level in the HL7 2.8 Conformance document implies that a higher level (a larger number) meets all the requirements of lower levels in a progressive fashion similar to that of the rows of the specification stack in SAEAF. However, a closer examination of the levels in the HL7 2.8 document appears to confirm that this is not the case.

For example, level 3 requires the profile to be specified in a way that is machine processable, and level 4 requires the profile to be “implementable.” (All options are removed so that no more constraint is possible.) Therefore, it is possible to have an implementation with a conformance claim against an implementable profile (level 4) that is not machine processable (level 3). Thus, level 4 conformance does not imply level 3 conformance.

In the SAEAF SS, conformance to a conformance statement in row 2 (PIM) guarantees conformance to the corresponding conformance statement in row 1, and likewise for conformance statements in row 3 relative to row 2, a fact guaranteed by the formal notion of row-to-row, MDA-based compliance. Finally, note that level 6 addresses compatibility, not conformance (as stated in the comments to the Conformance document).

2. SAEAF is meant to have application beyond HL7, such as for non-SDOs, which produce specifications to enable Working Interoperability for healthcare, life sciences, and clinical research. Hence, the more general term specification is used rather than the more restrictive term standard.

While SAEAF refers to the collection of rules that are defined to achieve Working Interoperability as the specification, an individual rule—if it is testable as a point of conformance—is referred to as conformance statement. (See conformance assertion below in list item #eccf_reference_notes_for_saeaf_and_hl7_28_comparison__nt4)
SAEAF uses the term conformance statement taken from RM-ODP to mean “a verifiable statement used in the context of a specification of a component.” Likewise, a given implementation of a specification or standard contains one or more conformance assertions, one for each conformance statement that the implementation claims to satisfy, such as statements that the implementation claims to be “conformant to.”

Note: In the HL7 2.8 Conformance document, the term conformance claim is never formally defined. However, it is used in a manner that is consistent with the SAEAF definition.

4. SAEAF recognizes that the term certification is used in two different senses:

1. As a description of the process of determining the validity of a given conformance assertion; and

2. As a statement that a given conformance assertion has in fact been verified as True. The HL7 2.8 document refers to the former meaning as a “test,” while referring to the latter as “certification.”

5. SAEAF recognizes that the term certification is used in two different senses:

1. As a description of the process of determining the validity of a given conformance assertion; and

2. As a statement that a given conformance assertion has in fact been verified as True. The HL7 2.8 document refers to the former meaning as a “test,” while referring to the latter as “certification.”

6. Although the definition of the SAEAF term compliance is used in the HL7 2.8 document, it is used as an overloading of the term conformance. As noted in the discussion of the Section 1.4.4, the ECCF identifies the two terms conformance and compliance as distinct concepts, employing the distinction established by ISO and other national health IT programs.

7. SAEAF distinguishes two types of rules:

1. Conformance assertions; and

2. Transformable content of specifications or standards that a given specification or standard uses.

In the context of the first use of the term, see row 4 in Table 3 and its associated comments. In the context of the second usage of the term, some difference in usage exists between SAEAF and the HL7 2.8 Conformance documentation (hence the marking of the row in red). In particular, at the original rule level in the “standard,” the HL7 2.8 Conformance document does not use a separate terms to differentiate what SAEAF calls conformance vs. compliance contexts. However, at the testing level, the HL7 2.8 documentation differentiates “profile testing,” which in SAEAF calls “compliance certification.” For more information about what SAEAF calls “conformance certification,” see rows 5 and 6 in Table 3, and the Section 1.4.4.5 section.

8. The HL7 2.8 Conformance document’s use of the term compatibility talks about issues that could prevent interoperability between two implementations of the same specification or standard. An example would be two SS instances defined for the same subject. In HL7 2.8 terms, the two implementations are incompatible if, in spite of being “legal” and “conformant” implementations to a given standard, they each make localizations or constraints that prevent them from interoperating.

1.4.8.2. ISO Terminology: Conformance and Compliance
ISO has periodically produced standards that address the core constructs around which the ECCF is defined, that is, conformance and compliance.


Note 1: The subject field of conformity assessment includes activities defined elsewhere in the above ISO references, such as testing (ISO 9000:2005, 4.2), inspection (ISO 9000:2005, 4.3), and certification (ISO 9000:2005, 5.5), as well as the accreditation (ISO 9000:2005, 5.6) of conformity assessment bodies (ISO/IEC 17000:2005, 2.5).

Note 2: The expression "object of conformity assessment" or "object" is used in this International Standard to encompass any particular material, product, installation, process, system, person, or body to which conformity assessment is applied. A service is covered by the definition of a product (see Note 1 to ISO 9000:2005, 3.3).

Consistency of ISO and ECCF Terms

Conformance:
The ECCF term conformance is synonymous with the ISO term conformity, but it is deprecated by ISO. There is, therefore, no inherent semantic conflict between the ECCF and the existing ISO standard.

Compliance:
This term is not used directly by ISO, but rather is viewed as a subtype of conformity assessed through a conformity assessment. Therefore, there is no conflict with the ECCF’s use of the term (see discussion below).

Certification:
The ECCF foundational concept of certification is subsumed by the ISO term conformity assessment, the ISO umbrella term for the collected activities of testing, inspection, measurement, and calibration. In addition, ISO Guide 67 defines product certification as an activity by which a third party gives written assurance that a product (including process and service) fulfills specified requirements. (Conformity assessment - Fundamentals of product certification). The critical feature of both the ECCF and ISO terminology is the administration of the process by a “trusted third-party organization” not involved in the construction of the component whose conformity is being assessed. Note that the term conformity assessment carries a more general connotation than certification, the latter often being restricted to automated rather than non-automated verification processes. Finally, note that the term conformity assessment vs. conformity. The act vs. the result is clarified through using the two terms, a fact that is not true when one uses the term certification. The conclusion, therefore, is that the ECCF should probably more closely align with ISO terminology, thereby deprecating the use of the term certification and being renamed to the Enterprise Conformity Assessment Framework.

Non-automated certification:
Non-automated certification of certain conformance assertions might be required if the corresponding conformance statements are made only at the CIM or even the PIM level. Only PSM-level conformance statements have a high chance of fully automated certification.