HL7 Service-Aware Interoperability Framework: Canonical Definition Specification, Release 2

Draft Standard for Trial Use

May 2012

Publication of this draft standard for trial use and comment has been approved by Health Level Seven International (HL7). This draft standard is not an accredited American National Standard. The comment period for use of this draft standard shall end 24 months from the date of publication. Suggestions for revision should be submitted at http://www.hl7.org/dstucomments/index.cfm.

Following this 24 month evaluation period, this draft standard, revised as necessary, will be submitted to a normative ballot in preparation for approval by ANSI as an American National Standard. Implementations of this draft standard shall be viable throughout the normative ballot process and for up to six months after publication of the relevant normative standard.

Copyright © 2012 Health Level Seven International®. ALL RIGHTS RESERVED. The reproduction of this material in any form is strictly forbidden without the written permission of the publisher. HL7 International and Health Level Seven are registered trademarks of Health Level Seven International. Reg. U.S. Pat & TM Off.
IMPORTANT NOTES:

HL7 licenses its standards and select IP free of charge. If you did not acquire a free license from HL7 for this document, you are not authorized to access or make any use of it. To obtain a free license, please visit http://www.HL7.org/implement/standards/index.cfm.

If you are the individual that obtained the license for this HL7 Standard, specification or other freely licensed work (in each and every instance "Specified Material"), the following describes the permitted uses of the Material.

A. HL7 INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS, who register and agree to the terms of HL7’s license, are authorized, without additional charge, to read, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part without paying license fees to HL7.

INDIVIDUAL, STUDENT AND HEALTH PROFESSIONAL MEMBERS wishing to incorporate additional items of Special Material in whole or part, into products and services, or to enjoy additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS as noted below, must become ORGANIZATIONAL MEMBERS of HL7.

B. HL7 ORGANIZATION MEMBERS, who register and agree to the terms of HL7’s License, are authorized, without additional charge, on a perpetual (except as provided for in the full license terms governing the Material), non-exclusive and worldwide basis, the right to (a) download, copy (for internal purposes only) and share this Material with your employees and consultants for study purposes, and (b) utilize the Material for the purpose of developing, making, having made, using, marketing, importing, offering to sell or license, and selling or licensing, and to otherwise distribute, Compliant Products, in all cases subject to the conditions set forth in this Agreement and any relevant patent and other intellectual property rights of third parties (which may include members of HL7). No other license, sublicense, or other rights of any kind are granted under this Agreement.

C. NON-MEMBERS, who register and agree to the terms of HL7’s IP policy for Specified Material, are authorized, without additional charge, to read and use the Specified Material for evaluating whether to implement, or in implementing, the Specified Material, and to use Specified Material to develop and sell products and services that implement, but do not directly incorporate, the Specified Material in whole or in part.

NON-MEMBERS wishing to incorporate additional items of Specified Material in whole or part, into products and services, or to enjoy the additional authorizations granted to HL7 ORGANIZATIONAL MEMBERS, as noted above, must become ORGANIZATIONAL MEMBERS of HL7.

Please see http://www.HL7.org/legal/ippolicy.cfm for the full license terms governing the Material.
Service-Aware Interoperability Framework:  
Canonical Definition  
Draft Standard for Trial Use, Release 1

| Chair | Charlie Mead  
|-------|--------------------------------------------------|
| Vice Chair | Ron Parker  
| National Cancer Institute, Center for Biomedical Informatics and Information Technology  
| Secretary & Editor | Anthony Julian  
| Mayo Clinic  
| Technical Editor | Ann Wiley  
| Sponsoring Work Group | Architecture and Review Board  
| List Server | arb@lists.hl7.org  

In addition, the ArB wishes to acknowledge the contributions of the following persons:

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andy Bond</td>
<td>NEHTA</td>
</tr>
<tr>
<td>Jane Curry</td>
<td>Health Information Strategies</td>
</tr>
<tr>
<td>Bo Dagnall</td>
<td>Hewlett Packard</td>
</tr>
<tr>
<td>Grahame Grieve</td>
<td>Health Intersections Pty Ltd</td>
</tr>
<tr>
<td>Steve Hufnagel</td>
<td>U.S. Department of Defense, Military Health System</td>
</tr>
<tr>
<td>John Koisch</td>
<td>Guidewire Architecture</td>
</tr>
<tr>
<td>Patrick Loyd</td>
<td>Icode Solutions</td>
</tr>
<tr>
<td>Anil Luthra</td>
<td>National Cancer Institute</td>
</tr>
<tr>
<td>Cecil Lynch</td>
<td>Accenture</td>
</tr>
<tr>
<td>Zoran Milosevic</td>
<td>NEHTA</td>
</tr>
<tr>
<td>Wendell Ocasio</td>
<td>Agilex Technologies</td>
</tr>
<tr>
<td>John Quinn</td>
<td>Health Level Seven, Inc.</td>
</tr>
<tr>
<td>Abdul Malik Shakir</td>
<td>Shakir Consulting</td>
</tr>
<tr>
<td>D. Mead Walker</td>
<td>Health Data and Interoperability Inc.</td>
</tr>
</tbody>
</table>
## Table of Contents

1 INTRODUCTION .......................................................................................................................... 7  
   1.1 BACKGROUND .................................................................................................................. 7  
      1.1.1 Overview of the SAIF-CD ....................................................................................... 8  
      1.1.2 The SAIF Value Proposition ................................................................................... 11  
      1.1.3 The Four SAIF-CD Frameworks ............................................................................ 12  
      1.1.4 Conventions Used in this Document .................................................................... 14  
      1.1.5 SAIF-CD and the OASIS Reference Architecture Foundation (RAF) .................... 15  

2 GOVERNANCE FRAMEWORK .................................................................................................. 18  
   2.1 PURPOSE ..................................................................................................................... 18  
      2.1.1 Governance, Management, and Methodology ..................................................... 18  
      2.1.2 Shared Purpose .................................................................................................... 19  
   2.2 GF CONCEPT MAP ....................................................................................................... 20  
      2.2.1 GF Terms of Art ............................................................................................... 21  
      2.2.2 Governance Language ..................................................................................... 24  
      2.2.3 Governance Processes ...................................................................................... 25  
      2.2.4 Relationship between the Governance Framework and the Behavioral Framework ................................................................................................. 26  

3 BEHAVIORAL FRAMEWORK ................................................................................................. 27  
   3.1 PURPOSE ..................................................................................................................... 27  
   3.2 CONTRACT SEMANTICS .............................................................................................. 29  
   3.3 OPERATION SEMANTICS ............................................................................................ 30  
   3.4 PROCESS SEMANTICS ................................................................................................. 32  

4 INFORMATION FRAMEWORK (IF) ...................................................................................... 33  
   4.1 PURPOSE ..................................................................................................................... 33  
   4.2 GOALS ......................................................................................................................... 33  
   4.3 DATA AND INFORMATION ......................................................................................... 33  
   4.4 CONCEPT COMPONENT ............................................................................................. 34  
   4.5 CONTROLLED TERMINOLOGY ................................................................................. 35  
   4.6 UN-ENCODED CONCEPTS ......................................................................................... 37  
   4.7 CONCEPT GROUPING ................................................................................................. 38  
      4.7.1 Code Systems ..................................................................................................... 38  
      4.7.2 Semantic Types .................................................................................................. 38  
      4.7.3 Value Sets ........................................................................................................... 38  
   4.8 DATA TYPE .................................................................................................................. 38  
   4.9 CLASSES ..................................................................................................................... 39  
   4.10 TERMINOLOGY BINDING .......................................................................................... 39  
   4.11 INFORMATION MODELS ......................................................................................... 39  
      4.11.1 Reference Information Model ............................................................................. 41  
      4.11.2 Domain Information Model ............................................................................. 42  
      4.11.3 Bridging between the Domain and the reference model ................................... 42  
      4.11.4 Logical Information Model .............................................................................. 42  
   4.12 TEMPLATES .................................................................................................................. 42  
   4.13 EXECUTABLE MODELS .............................................................................................. 42
## ENTERPRISE CONSISTENCY AND CONFORMITY FRAMEWORK (ECCF)

5.1 PURPOSE ...................................................................................................................... 44
5.2 ECCF TERMS OF ART .................................................................................................. 44

## INTEROPERABILITY SPECIFICATION MATRIX (ISM)

6.1 ISM ARTIFACTS TYPES AND CONFORMANCE STATEMENT TYPES ........................................ 49
6.2 DIMENSIONS .................................................................................................................. 50
   6.2.1 Enterprise Dimension .................................................................................................. 50
   6.2.2 Information Dimension ............................................................................................... 50
   6.2.3 Behavioral (Computational) Dimension ........................................................................ 50
   6.2.4 Engineering Dimension ............................................................................................... 50
   6.2.5 Technology Dimension ............................................................................................... 50
6.3 PERSPECTIVES .............................................................................................................. 51
   6.3.1 Conceptual Perspective ............................................................................................... 51
   6.3.2 Logical Perspective ...................................................................................................... 51
   6.3.3 Implementable Perspective ........................................................................................... 51

## COMPLIANT SAIF IMPLEMENTATION GUIDES

7.1 EVOLUTION OF THE SAIF CD ...................................................................................... 54
7.2 SAIF CD COMPLIANCE STATEMENTS: CRITERIA FOR A COMPLIANCE SAIF IMPLEMENTATION GUIDE ................................................................. 55

## APPENDIX

8.1 ISM SPECIFICATION MATRIX, TEMPLATE AND INSTANCE ......................................................... 72
8.2 FOUNDATIONAL PRINCIPLES ............................................................................................ 76
   8.2.1 Shared Purpose.............................................................................................................. 76
   8.2.2 Fowler’s Accountability Pattern .................................................................................... 77
   8.2.3 “Service-Awareness” .................................................................................................. 77
8.3 DEFINING A SAIF IMPLEMENTATION GUIDE ....................................................................... 79
   8.3.1 “SAIF enough – the Linear Value Proposition” ................................................................ 79
   8.3.2 Deployment Context versus Interoperability Type .......................................................... 80
   8.3.3 Defining Specification Artifacts: Content, Representation, Location ................................ 80
   8.3.4 Building SAIF Specifications ....................................................................................... 80
8.4 GOVERNANCE IMPLEMENTATION CONSIDERATIONS ......................................................... 81
8.5 SAIF-CD ADOPTION AND ADAPTION OF EXISTING AND/OR RELATED WORK ......................... 81

## WORKS CITED

9.1 ........................................................................................................................................ 86
TABLE OF FIGURES

FIGURE 1 RELATIONSHIP BETWEEN SAIF-CD AS A TYPE, COMPLIANT SAIF IMPLEMENTATION GUIDES (IGs) ..............................................................................................................9
FIGURE 2 — SAIF-CD: BASIC STRUCTURE. (SEE FIGURE 1 NOTES FOR MEANING OF COLORS). .................................................................................................................10
FIGURE 3 SAIF-CD ORGANIZATION AND STRUCTURE .................................................................................................................................11
FIGURE 4 INTER-RELATIONSHIPS OF FOUR SAIF-CD LANGUAGES .........................................................................................................................14
FIGURE 5 THE AMOUNT AND TYPE OF GOVERNANCE (USED WITH PERMISSION FROM NCI CBIIT) ..................................................................................20
FIGURE 6 GOVERNANCE FRAMEWORK CONCEPT MAP..................................................................................................................21
FIGURE 7 GOVERNANCE DESIGN DOCUMENTATION TEMPLATE (FROM ERL ET AL., 2011) ................................................................................................................24
FIGURE 8 BF LANGUAGE CONCEPTS AND RELATIONSHIPS FOR DESCRIBING CONTRACT SEMANTICS ...........................................................................27
FIGURE 9 BF LANGUAGE CONCEPTS AND RELATIONSHIPS FOR DESCRIBING CONTRACT SEMANTICS ...........................................................................29
FIGURE 10 BF LANGUAGE CONCEPTS AND RELATIONSHIPS FOR DESCRIBING OPERATION SEMANTICS .........................................................................30
FIGURE 11 BF LANGUAGE CONCEPTS AND RELATIONSHIPS FOR DESCRIBING PROCESS SEMANTICS ...........................................................................32
FIGURE 12 INFORMATION FRAMEWORK CONCEPT MAP.........................................................................................................................33
FIGURE 13 EXAMPLE OF CONCEPTS ........................................................................................................................................36
FIGURE 14 EXAMPLE OF ALTERNATIVE TEXT FOR A CONCEPT ...............................................................................................................36
FIGURE 15 CONCEPT OVERLAP ..................................................................................................................................................36
FIGURE 16 CONCEPTUAL GRAPH DISPLAY FORM ............................................................................................................................37
FIGURE 17 OPENEHR PERSON DEMOGRAPHIC INFORMATION EXAMPLE© (OPENEHR FOUNDATION, 2001-2007) - ..................................................40
FIGURE 18 E_PERSON UNIVERSAL (COCT_RM030200UV08) CMET ........................................................................................................41
FIGURE 19 ARTIFACT CONTEXT WRAPPING ........................................................................................................................................43
FIGURE 20 ECCF TERMS OF ART CONCEPT MAP. (SEE FIGURE 1 FOR COLOR CONVENTION SEMANTICS) ...............................................................44
FIGURE 21 INTEROPERABILITY SPECIFICATION MATRIX CONCEPT MAP. (SEE FIGURE 1 FOR COLOR CONVENTION SEMANTICS) ........................................48
FIGURE 22 INTEROPERABILITY SPECIFICATION MATRIX .......................................................................................................................49
FIGURE 23 EXEMPLAR INTEROPERABILITY SPECIFICATION MATRIX...................................................................................................................72
FIGURE 24 ANOTHER VIEW OF AN IST ...........................................................................................................................................73
FIGURE 25 BINDING II TO SI THROUGH CONFORMANCE ASSERTIONS ........................................................................................................74
FIGURE 26 RELATIONSHIPS BETWEEN THE ISM, IST, AND ISIS. ...............................................................................................................75
FIGURE 27 CONCEPT MAP REPRESENTATION OF THE ACCOUNTABILITY PATTERN OF MARTIN FOWLER ........................................................................77
FIGURE 28 SHARED PURPOSE CONCEPT MAP............................................................................................................................78
FIGURE 29 DEPLOYMENT CONTEXT VERSUS INTEROPERABILITY TYPE MATRIX (COURTESY OF NCI CENTER FOR BIOMEDICAL INFORMATICS AND INFORMATION TECHNOLOGY (NCI CBIIT) ..............................................................................79
1 Introduction

1.1 Background

The development of the SAIF Canonical Definition (SAIF-CD) – which began in early 2008 – was motivated and directed by a high-level set of requirements communicated to the Health Level Seven International (HL7) Architecture Board (ArB) by the HL7 Chief Technology Officer (CTO) and senior representatives of several large national programs whose representatives participate in various HL7 activities. In particular, the ArB was directed to provide a coherent, enterprise-architecture-centric approach that would enable the explicit description of technology components – including, but not necessarily limited to, HL7-specified artifacts – from the perspective of the interactions between those components as they were involved in scenarios whose purpose was to achieve an agreed-upon goal based on “cross-organizational-boundary shared purpose.” The scope of the components themselves was not specified, i.e. a “component” could be defined as a system, a service, a message, a document, an enterprise, or a generic party. The notion of “interactions to achieve an agreed upon goal based on cross-organizational-boundary shared purpose” was assumed to mean – at a technical level – some degree of technical – most likely semantic – interoperability between the involved components that itself was a manifestation of a non-technical agreement and definition of a joint (i.e. cross-organizational-boundary) shared purpose.

NOTE: From this point forward, this document will use the term “cross-boundary” to indicate scenarios that involve interactions, i.e. interoperability across one of a number of possible boundaries, e.g. departmental, disciplinary, organizational, enterprise, jurisdictional, etc. A common – but not required – characteristic of cross-boundary interactions is the fact that not all of the components/systems/technologies/resources required for the interaction are under the control of a single person or organization.

As the ArB began considering its task from the perspective of the collective experience of its members, the core effort soon became focused not on defining an HL7 enterprise architecture, but rather on standardizing a set of languages that could be used to explicitly define and unambiguously specify the various factors that enable interoperability between the components. In particular, the ArB focused on defining a set of canonical frameworks consisting of languages that could then be instantiated in organization-specific Implementation Guides (IGs). The distinction between the languages defined by the SAIF-CD and an organization-specific IG’s instances of those languages – collectively referred to as grammars in this document – is as follows:

**Language (SAIF-CD):** The concepts and relationships defined in the SAIF-CD. Many are taken from the Enterprise Viewpoint and Computational Viewpoint languages of RM-ODP (ISO RM-ODP).

**Grammar (SAIF-CD):** The adoption or adaption, optimization, realization, and/or contextualization of the languages specified in the SAIF-CD for use in organization-specific SAIF Implementation Guides (SAIF IG).

In addition to the need for languages to define aspects of static (informational) and dynamic (behavioral) semantics, as well as a language for use in discussing issues of conformance, compliance, compatibility, and other related terms. it was also recognized that the CD required a formal governance language that allowed the clear expression of the various jurisdiction-based linkages between organization-level definition of shared purpose and its technical realization in specific run-time components. This requirement stems from the fact that technical interoperability is, in fact, a manifestation of a “higher level” of cross-organization/cross-boundary agreements (in the jurisdictional or administrative sense) between human beings and/or the organizations they represent.

In summary, the SAIF-CD defines a minimal set of common concepts and relationships from which compliant SAIF IG-specified models can be defined. These models, in turn, support a number of different technical approaches to interoperability including messages, documents, or services that individually or collectively enable the successful realization of shared-purpose scenarios. A SAIF IG thus adopts and defines both modeling languages and grammars, and document artifact templates, that are compliant with the concepts and properties defined in the SAIF-CD. In terms of the separation between language and grammar mentioned...
above, the SAIF-CD defines a language – or, more correctly a set of inter-linked languages – that a particular organization can use to specify organization-specific restrictions or specializations of those languages, i.e. grammars – documented in the organization’s SAIF Implementation Guide. The content of a particular SAIF IG thus defines how an organization specifies the various dimensions of components from the perspective of interoperability/shared purpose scenarios in which the components are invoked. Thus, IG-specific grammars adopt, adapt, organize, realize, and contextualize the SAIF-CD languages in ways suitable for the organization’s own interoperability requirements and goals using that organization’s adopted (or adapted) modeling conventions and specific grammars, reference models, technology choices, etc. NOTE: It is not strictly true that IGs contain only grammars. In fact, they may contain organization-specific contextualization of general language constructs defined in the SAIF CD. In general, however, the separation of language/CD and grammar/CD is useful and will be the convention of this document.

It should also be noted that the concept of interoperability in the context of the SAIF-CD is rather broad-based. In particular, it is ultimately grounded on the basic notion of shared purpose resulting in defined value for the various parties involved in one or more interoperability scenarios. It should be noted that interoperability at a technical level may be characterized as one of several Interoperability Types, involving simply the exchange of structure (syntax) versus the more difficult exchange of meaning (semantics) between humans (e.g. browser-compatible documents) or machines. Thus, defining and achieving shared purpose between two organizations via an implementation involving various software components designed, developed, and deployed by the organizations includes a context-specific discussion of human-to-human, human-to-machine, and/or machine-to-machine interactions. Experience has repeatedly shown that semantic interoperability between machines – known as computable semantic interoperability (CSI) – is by far the most difficult and expensive type of interoperability to achieve in a scalable, tractable manner, particularly when the interoperability scenarios cross one or more organizational boundaries. The SAIF-CD refers to this construct as the “Deployment Context” of the scenario. (See the Governance Framework and the Appendix for more discussion on Interoperability Type versus Deployment Context.)

Given the fact that an enterprise architecture should support the business of the enterprise that defines and develops said enterprise architecture, it is important to note that the SAIF-CD is specifically meant to function not as a replacement for, but rather as an adjunct to, existing enterprise-centric architecture frameworks including RM-ODP (ISO RM-ODP), Zachman2 (Zachman), TOGAF (The Open Group), DoDAF (US Department of Defense Architecture Framework), Lopez/Blobel’s description of a healthcare-specific architecture (Lopez, 2009), etc. Specifically, the SAIF-CD defines the languages necessary for focusing component specification on cross-boundary (e.g. cross-enterprise), shared-purposed interoperability that is itself focused on achieving a mutually beneficial goal.

1.1.1 Overview of the SAIF-CD

The purpose of the HL7 Service-Aware Interoperability Framework Canonical Definition (SAIF-CD) is to provide the “top-level” specification of SAIF. As such, the SAIF-CD is written for persons or organizations that are interested in implanting SAIF as an adjunct to existing (or planned) enterprise architecture frameworks because of SAIF’s singular focus on the various dimensions and perspectives associated not with enterprise architecture per se, but rather with achieving predictable, scalable, and effective interoperability between the various software components that collectively populate one or more enterprise architectures. Such implementations are most effectively done through the development of an organization-specific SAIF Implementation Guide (SAIF IG). Indeed, the HL7 SAIF-CD is intended to be used primarily by the authors of an enterprise’s SAIF IG and therefore its value to an enterprise’s analysts, architects, developers, or other enterprise architecture stakeholders is:

1. a general reference regarding the various SAIF languages, and
2. a “how to build a SAIF IG” manual in the sense that the normative version of the SAIF CD – i.e. this document – contains specific criteria that enables SAIF IG authors to assess the degree to which their SAIF IG is compliant with the CD’s specification (Section 2). Examples of some of the specific steps and end results of using the SAIF-CD to define a specific SAIF IG are collected in the Appendices of this document.
In summary, the main constructs in the “SAIF stack” are the SAIF-CD, SAIF IGs, and SAIF-IG-compliant component specifications (Figure 1). In particular, the most visible vestige of the “SAIF stack” – the Interoperability Specification Matrix and its derivatives – is specifically identified. In particular, it is important to note that the SAIF-CD defines a single Interoperability Specification Matrix (ISM) as a type. One-to-many SAIF Implementation Guides (SAIF IGs) can then be defined as profiles on that type. A substantive portion of a SAIF IG is, in fact, the specification of the content, representation, and specific cell location(s) for each artifact in the SAIF IG-specific Interoperability Specification Template (IST). Finally, as a given SAIF IG is operationalized, any number of specification instances are produced, each referred to as an Interoperability Specification Instance (ISI). Following specification, one or more implementation instances of a given specification instance may be developed and – if so desired – subject to conformity testing. Figure 1 depicts the Relationship between SAIF-CD and the Interoperability Specification Matrix as a Type, compliant SAIF Implementation Guides (IGs) and their Interoperability Specification Templates as profiles on that type, instances of component specifications as instances, and Conformant Component Instances (see Sections 2 and 6 and the Appendix for more detailed discussions).

The following conventions are used in this and subsequent concept maps:

- Blue concepts are defined in the SAIF-CD
- Yellow concepts in an organization specific IG
- Green concepts are instance specifications developed using definitions supplied by a specific SAIF-IG
- Purple is used to indicate run-time instances of specification instances, which are, in turn, shown as green

![Figure 1 Relationship between SAIF-CD as a Type, compliant SAIF Implementation Guides (IGs)](image)

The SAIF-CD defines the essential concepts and constructs necessary for an organization to define its own SAIF Implementation Guide (SAIF IG) in such a manner that that IG will be compliant with the SAIF-CD. The basic structure of the SAIF-CD as well as its high-level relationship to enterprises and their architectures and SAIF IGs is shown in the following concept map.
In summary, a critical component in understanding the operationalization of SAIF is having a clear distinction of what is defined where, i.e. what is defined in the SAIF CD, a particular enterprise’s SAIF IG, and the instantiation of IG-specified, individual component interoperability specifications and implementations. These specifications, in turn, are used to produce conformant implementations artifacts which can be subjected to conformance testing/validation.

The following concept map (Figure 3) provides a more complete overview of the SAIF-CD (note that in addition to the color scheme explained in Figure 1, this figure adds a fifth color – terra cotta – to identify external informational resources used in defining the SAIF CD, e.g. RM-ODP):
NOTE: Use of concepts taken from the ODP Viewpoints in combination with SAIF Perspectives provides SAIF the basis for addressing issues that directly emerge from focusing on interoperability scenarios. In particular, the SAIF-CD leverages the core intent of the ODP standards, to provide a technology-independent framework for specifying enterprise distributed systems, while explicitly providing mechanisms for addressing various organizational modeling issues. Examples are organizational and legislative polices defined by the administrative boundaries, and regional and state jurisdictions – issues which are explicitly addressed in the SAIF-CD through the use of Perspectives.

1.1.2 The SAIF Value Proposition

The SAIF-CD defines a specification that can be used by multiple organizations to build organization-specific, SAIF-CD-compliant SAIF Implementation Guides (SAIF IGs). An organization interested solely in intra-enterprise component interoperability could certainly define a “SAIF-like” set of requirements for the artifacts needed to collectively specify a given software component to interoperate with other components without the use of the SAIF-CD per se. However, achieving inter-organization/cross-boundary interoperability presents greater challenges since it is necessary to ensure that the “expectations” of each party involved in a given interoperability scenario – expectations manifested in a particular software component developed by one of the participating parties – have been quantitatively assessed for completeness and correctness.

If both organizations have specified their respective components using their own SAIF-CD-conformant SAIF IG, the task of component specification comparison and harmonization – and, if necessary, refactoring – becomes considerably more tractable because the framework within which the comparison/harmonization/refactoring activities are executed, i.e. the SAIF-CD-compliant SAIF IGs, eliminates or minimizes many of the tricky linguistic and representational differences between the two organizations’ ways of defining component semantics and their representations. Thus, the development of SAIF-CD compliant SAIF IGs enables organizations to explicitly discuss and negotiate their cross-boundary shared purposes as operationalized in component interoperability.
It should be noted that independently designed components may still not be interoperable due to incompatible requirements. However, if specifications are explicit and expressed using the languages provided by the SAIF CD, targeted comparison, harmonization, and refactoring can more effectively and efficiently take place. In summary, negotiations between various information exchange communities can lead to explicit agreements that can result in components participating in a truly distributed, interoperable ecosystem. SAIF thus enables cross-boundary risk reduction in the context of interoperability scenarios requirements.

The SAIF-CD defines the languages for explicitly specifying informational (static) and behavioral (dynamic) semantics at the level of a software component (for example, services, messages, and documents). In addition, it provides direction as to how Conformance Statements may be included in a given specification instance. Specification-specific Conformance Statements can then be associated with pair-wise, implementation-instance-specific Conformance Assertions to assess the conformity of a given run-time Component Implementation.

1.1.3 The Four SAIF-CD Frameworks

1.1.3.1 Governance Framework (GF)

The language of the Governance Framework (GF) enables an enterprise implementing SAIF to define explicit, organization-specific policies, standards, and roles to artifact-specific content and representational choices that use the languages specified in the Behavior and Information Frameworks. The overall management of the life cycle of each SAIF artifact, including the correctness and completeness and any IG-specified RACI relationships, is defined by the Governance Framework language. As such, the GF aides and organization in risk management by providing a language that can be used to apply governance at specific high-risk operational points.

The GF uses a governance documentation framework adopted from a recent publication (Thomas Erl, 2011). As explained in detail in the GF discussion in this document, the framework includes Precepts – further defined specific “governance points” in terms of Objectives, Policies, Standards, Guidelines – People (and their associated Roles and including both organizations and systems), Processes, and Metrics. A SAIF-IG operationalizes the GF language in an organization-specific SAIF IG grammar, to explicitly cover concepts like expectations, granting of authority and resources, verifying performance, managing configuration baselines and related concerns.

Cross-boundary shared purpose as it is achieved through technical interoperability represents a set of agreements between the human and organizational owners of the components that are ultimately deployed and interact to achieve a defined set of shared objectives. In particular, technical, component-specific contracts are specified as a means of providing technical realizations of formal (or informal) contracts between human beings and enterprises. As such, readers of the SAIF-CD will note this intersection of the human and organizational and technical perspectives on interoperability in many of the terms used in both the Behavioral Framework and Governance Framework chapters of the SAIF-CD.

NOTE: The language describing certain targeted types of governance -- e.g. general Interoperability Specification Template and specific artifact well-formed-ness, and conformance and compliance testing and certification of specification-specific implementations – is defined in a separate SAIF-CD chapter, i.e. the Enterprise Consistency and Conformity Framework (ECCF).

1.1.3.2 Behavioral Framework (BF)

The language of the Behavioral Framework (BF) defines constructs to specify the dynamic semantics of interactions in a shared purpose interoperability scenario. The BF focuses on the languages necessary to define
the semantics of contracts, operations, and processes that collectively define shared purpose scenarios at a technical level. Collectively, the BF languages – and their IG-specific grammars – describe “who does what when and how.” In particular, contracts are expressed as implicit or explicit agreements at a number of jurisdictional boundaries including those between business objects, components, applications, systems and/or enterprises/organizations. The BF language specifies constructs describing various system role relationships expected by various stakeholders, system components, and/or applications. These relationships involve information exchanges and behavioral interactions in support of shared purpose scenarios.

The other SAIF-CD frameworks work with – and in support of – the BF. In particular, the GF provides the language to both define the non-technical constructs of shared purpose, as well as to bind organizational and technical risk management to component development and use. The IF and BF languages enable the explicit specification of business objects, components and their services, capabilities, applications, systems and their respective roles, responsibilities and interactions such as information exchanges. The ISM and the ECCF provide the structure and language for documenting and managing technical component specifications.

Note to SAIF IG Developers: It is not necessarily true that a given SAIF IG will cover the complete scope of the BF language. In addition, it is not the case that only a single language or grammar will be required to cover all three of the Interoperability Specification Matrix (ISM) Perspectives with respect to behavioral semantics involved in organization-specific specification content, syntax and representation. In fact, different Perspectives may naturally give rise to different grammars (and representations) in the context of a given conformant SAIF IG.

1.1.3.3 Information Framework (IF)

The Information Framework (IF) defines the language required for discussing and defining the static/informational semantics relevant to interoperability scenarios including concepts such as information and terminology models, metadata, vocabulary bindings, value sets, executable models, etc. that collectively specify the static semantics of interactions. This includes the language to describe patterns of structured and unstructured data, documents, messages and services, quality measures and transformations.

The IF also defines the language necessary to explicitly describe how these various information/static semantic constructs are related to each other in a composite static semantic “whole” in the context of a shared purpose interoperability scenario.

Note to SAIF IG Developers: It is not necessarily true that a given SAIF IG will cover the complete scope of the IF language. In addition, it is not the case that only a single language or grammar will be required to cover all three of the Interoperability Specification Matrix (ISM) Perspectives with respect to informational semantics involved in organization-specific specification content, syntax and representation. In fact, different Perspective may naturally give rise to different grammars (and representations) in the context of a given conformant SAIF IG.

1.1.3.4 The Enterprise Consistency and Conformity Framework and the Interoperability Specification Matrix

The Enterprise Consistency and Conformity Framework (ECCF) defines the language necessary to describe the various relationships – e.g. conformance, compliance, consistency, traceability, compatibility, etc. – between the artifacts that collectively define a given specification, including how a given specification relates to both derived implementations of the specification, and other specifications that use one or more of the artifacts as part of their artifact collection. In contrast, the ISM itself defines the structure – a 5 x 3 non-normalized matrix – that is used to collect the various artifacts that collectively specify information exchange and interaction details that define a component’s capabilities and accountabilities. IG-specific instances of the ISM – referred to as Interoperability Specification Templates (ISTs) – actually collect the various artifacts and artifact-specific Conformance Statements that can be used to evaluate the conformance of a given application instance to a given specification. Thus, the IF and BF formally define the essential concepts and relationships necessary to define within a given SAIF-IG, i.e. what can be specified, the ISM defines how artifacts can be sorted and collected based on their particular Dimension and Perspective, while the ECCF defines the relationships between artifacts, e.g. conformance, compliance, compatibility, etc.
Note to SAIF IG Developers: It is not necessarily true that a given SAIF IG will cover the complete scope of the ECCF language. In addition, it is not the case that only a single language or grammar will be required to cover all three of the Interoperability Specification Matrix (ISM) Perspectives with respect to consistency and conformity semantics involved in organization-specific specification content, syntax and representation. In fact, different Perspective may naturally give rise to different grammars (and representations) in the context of a given conformant SAIF IG.

1.1.3.5 Inter-relationships among the four SAIF-CD Languages

The languages of the four frameworks of the SAIF-CD – i.e. the GF, BF, IF, and ECCF – should not be viewed as siblings. Rather, they have a number of inter-relationships that, when understood, provide a layered, multi-dimensional view of the SAIF-CD as a specification for SAIF IGs. In particular, three relationships and their unifying concepts are of primary importance:

- GF and BF – related through the concepts of Shared Purpose and Objectives, and Role-based Communities and the subtype Governance-based communities
- GF and ECCF – related through the concept of Artifact Governance
- ECCF, BF and IF – related through the concepts of artifact syntax and semantics, and well-formedness.

The following concept map (Figure 4) provides a graphical view of these pivotal SAIF-CD inter-relationships:

![Inter-relationships of four SAIF-CD languages](Figure 4 Inter-relationships of four SAIF-CD languages)

1.1.4 Conventions Used in this Document

1.1.4.1 Index

Readers will find a comprehensive Index at the end of this document. Every attempt has been made to make the Index useful for targeted reference to selected topics within the SAIF Canonical Definition document.
1.1.4.2 **Glossary**

The SAIF Canonical Definition document does not include a Glossary. Rather, the HL7 Architecture Board (ArB) is developing and will maintain an online SAIF Glossary that includes definitions of relevant terms, specialized concepts, constructs, and artifacts as used in either or both the SAIF Canonical Definition and HL7 SAIF Implementation Guide. The online Glossary is updated between publications of the SAIF-CD and will be supported by an OWL representation to ensure ongoing logical consistency.

1.1.4.3 **Reference Material**

Reference Material containing additional information that is not part of the SAIF Canonical Definition including material such as auxiliary diagrams, examples, and additional explanations of material formally presented in the SAIF Canonical Definition document but deemed to not be an essential part of the balloted, normative content can be found in the various Appendices to the SAIF-CD.

1.1.4.4 **Footnotes**

When absolutely necessary for clarification of critical concepts, the SAIF Canonical Definition document includes footnotes. In the SAIF Canonical Definition document, footnotes are not, in general, used to provide definitions as these are collected in the SAIF Online Glossary. (HL7 ArB, 2011)

1.1.4.5 **Reader Feedback**

Readers wishing to suggest improvements to materials in this SAIF Canonical Definition are encouraged to

1. Participate in the ballot process at [www.hl7.org](http://www.hl7.org)
2. Subscribe to the HL7 Architecture Board list server and send their suggestions to [arb@hl7lists.org](mailto:arb@hl7lists.org).

1.1.5 **SAIF-CD and the OASIS Reference Architecture Foundation (RAF)**

Readers of the SAIF CD are strongly encouraged to review a document recently published by OASIS entitled “Reference Architecture Foundation for Service-Oriented Architecture.” The document was developed over the past five years, and represents a substantive, cross-industry effort to surface, define, and discuss in detail various aspects of a number of critical success factors involved in implementing large-scale (i.e. enterprises-level) service-oriented architectures with a focus on achieving both intra- and inter-enterprise technical interoperability.

In addition to providing an independent validation for the both the general focus as well as some of the concrete specifics of the SAIF CD (especially those involving the importance of governance in achieving large-scale interoperability), the OASIS document underscores several important aspects of the SAIF CD including:

1. A validation of one of the SAIF CD’s primary claims, i.e. the need to specifically focus on intra- and inter-enterprise interoperability as a first-class citizen in any enterprise (or cross-enterprise) architecture discussion irrespective of the particular choice of enterprise architecture approach, framework, or implementation technology, e.g. TOGAF, Zachman, ODP, SOA, etc. In addition, the OASIS document clearly articulates – as the SAIF CD does as well – the difficulties involved in achieving that focus in such a manner that it can be manifest in operationally effective and manageable processes and deliverables.

2. An agreement as to the critical importance of governance as the root of any successful effort to implement large-scale, cross-boundary interoperability aimed at achieving a collective shared purpose or goal. In particular, both documents share the notion that “technical-level” governance – e.g. service- or message-level technical interchange specifications – must itself be a manifestation of a higher-level, cross-jurisdictional agreement on desired goals, responsibilities, accountabilities, and deliverables.
3. A validation of the importance of core SOA constructs as constructs useful in expressing many of the central aspects of interoperability irrespective of whether a particular interoperability scenario is actually “realized” using SOA-compatible technologies. (NOTE: Although it might at first appear that the OASIS document is more “service-focused” than the “service-aware” document from HL7, there are considerably more similarities than differences in these slightly different foci secondary to the fact that both documents are intent on describing principles and framework concepts rather than delving into technical details. There are, however, certain instances where content of the OASIS document would be likely to find its analogue in SAIF Implementation Guides rather than in the SAIF Canonical Definition document.)

4. The need for specific, explicit statements of those aspects of a given component that affects its ability to participate in a reliable, predictable manner in a variety of interoperability scenarios. In particular, component characteristics must be explicitly expressed in both design-time and run-time contexts, as implicit assumptions are the root of most failures to successfully achieve cross-boundary interoperability irrespective of the chosen technical details of a particular interoperability instance.

In summary, although the two documents are clearly not identical in their specifics, e.g. there are differences in the language used to name various concepts, constructs, and relationships; there are some differences in levels of abstraction regarding certain topics, etc.; and although the OASIS RAF is more directly focused on services as a final implementation architecture than the HL7 SAIF CD, the commonalities of purpose, content, and approach present in the two documents – documents which were developed by each organization without any knowledge of the others’ work in what clearly are areas of common interest and concern – far outweighs their differences. As such, the HL7 ArB and the OASIS RAF Task Force have agreed to work together going forward to obtain the highest degree of alignment and harmonization possible between the two documents including the possible development of a joint document under the auspices of one of the ISO software engineering threads.

The current version of the OASIS document – as well as all future versions – is available at either: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/csd03/soa-ra-v1.0-csd03.pdf or http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm#technical
2 Governance Framework

2.1 Purpose

The purpose of the Governance Framework (GF) is to provide a language and set of constructs for individual organizations to define explicit sets of terms and processes that make the often-implicit “rules of the game” explicit, and thereby ensure a common – i.e. shared – understanding between the various organizations that are focused on achieving a given jointly negotiated shared purpose. Specifically, this is meant in the context of realizing such shared purpose in a technical solution that requires a specified type of interoperability (see Figure 5: Interoperability Types versus Deployment Context). In addition, the language of the GF enables organization-specific governance activities to be focused on known development-cycle risks, thereby maximizing the effectiveness and efficiency of resources expended in the name of governance.

2.1.1 Governance, Management, and Methodology

Governance is not equivalent to either management or methodology. Rather, it is both influenced by and related to both concepts. Following is a brief list of some of the differences between these three interrelated concepts:

- **Governance** establishes rules that control decision-making.
- **Methodology** establishes processes that comply with governance rules and may introduce additional rules.
- **Management** makes decisions according to governance rules.
- **Governance** does not dictate when or how to make a decision. It determines who should make the decision and establishes limits for that person or group.
- **Methodology** establishes processes that carry out specific types of decision that adhere to governance rules.
- **Management** is responsible for day-to-day operations and for ensuring that decisions made adhere to governance and methodology rules.
- **Governance** cannot replace management or methodology, nor can it compensate for poor management or poor (or inappropriate) methodology.
- Poorly defined and executed **methodology** can jeopardize the business goals associated with governance.
- Poor **management** can undermine a governance system and a methodology and will jeopardize associated business goals.
- Neither management nor methodology can replace governance, nor compensate for poor governance.

Governance is therefore best seen as a “meta” process which describes and oversees “how decisions about decision making” are made. At a high level, a well-defined governance system is characterized as having:

- identified constraints and control guidelines on management decisions
- defined the responsibility for and authority to make various decisions
enumerated the consequences of non-compliance to governance metrics

Thomas Erl’s recent book summarizes governance as follows:

“A good system of governance helps the members of an organization carry out responsibilities in a manner supportive of the organization’s business goals and vision. It mitigates conflict by clearly defining responsibilities and assignments of authority, and further reduces ambiguity by articulating constraints and parameters in practical forms (such as rules and decision guidelines). It also helps balance tactical and strategic goals by expressing the intents and purposes of its rules. (Thomas Erl, 2011)”

2.1.2 Shared Purpose

As stated above, the GF provides a language that can be used to explicitly define a set of “governed items and associated processes” including the relevant artifacts, metrics, roles, etc. It is important to note that the language of the GF is not specific to either the governance of people, organizations, enterprises, etc., or the governance of technology components, i.e. it applies equally well in both settings. This feature is of essential importance since, in fact, the governance that occurs at a computational interface via constructs such as pre-conditions, post conditions, contracts, roles, accountabilities, etc. is, in fact, a technical realization of an agreement between two or more participating parties to achieve a shared purpose. In order to be successful, such an agreement must clearly define responsibilities, expectations, and response to non-performance, the basic content of a contract.

Although governance is an important construct within a single department/organization/enterprise, it becomes a critical success factor when more than one independent entity – i.e. when the entities seeking to achieve a given shared purpose come from different governance spheres. The SAIF-CD assumes that execution context to achieve the shared purpose will be realized through a collection of technology-based components, the explicit details of which can be expressed in artifacts defined by SAIF Implementation Guides using the languages of the Behavioral, Information, and Enterprise Consistency and Conformity Frameworks defined in the SAIF-CD. The details of the shared purpose are not critical to the use of the language of the GF, i.e. governance is needed because the shared purpose of the community is to achieve objectives that cannot be achieved by participants acting autonomously. Thus, the shared purpose could be setting or refining international standards, collaborating to deliver healthcare services, developing technical components to enable system interoperability in order to share information or coordinate component behaviors in the context of healthcare delivery, health program evaluation, research, quality assurance, research or clinical trial needs, regulatory reporting obligations, etc. In the context of technical interoperability and shared purpose, well-defined governance is a Critical Success Factor.

Finally, it should be noted that governance is not a “one size fits all” construct. In fact, there are numerous dimensions that govern the decisions that will ultimately answer the questions “What needs to be governed?” and “How should it be governed?” In response to the first question, the GF provides language that can productively be applied to mitigate risk. With respect to the second question, two of the most important dimensions that determine “how much governance” a particular negotiated instance of shared purpose interoperability requires in order to succeed are Interoperability Type and Deployment Context. (See Appendix for a detailed discussion of the relationship between these two constructs.)
In summary, the parties participating in a shared purpose scenario realized through technical component interoperability do not need to agree to be governed by the same set of rules for all aspects of their respective operation. Those rules affecting their participation in shared activities need to be explicitly defined and negotiated through a GF-based mechanism. The establishment of shared rules is intended to reduce risks when working across boundaries. Evaluation of the types and impact of potential risks will prioritize those areas where clear “rules of engagement” are essential to success.

2.2  GF Concept Map

The core concepts and relationships of the GF language are pictured in the Concept Map and defined in the following section “GF Terms of Art.” Note in particular that the concept of “governance” itself – as expressed via the use of GF language – is colored yellow to indicate that it is an organization-specific construct that – as
explained earlier in this document – is expressed through an organization-specific instance of the GF language, i.e. it is expressed in an organization-specific GF grammar.

![Governance Framework Concept Map](image)

### Figure 6 Governance Framework Concept Map

#### 2.2.1 GF Terms of Art

The following terms are used in defining precepts and their relationships to each other. The source of these concepts is generally the INTERNATIONAL STANDARD ISO/IEC 15414 ITU-T RECOMMENDATION. 911 - Information technology – Open distributed processing – Reference model – Enterprise language (ISO RM-ODP). The concepts are paraphrased here to be more business-reader friendly and to permit this chapter to be read alone. In some cases, concepts are from other named sources. In addition, some concepts are paraphrased to add clarity for this framework.

**Note:** Several of the concepts in the GF language are similar in meaning to concepts used by the Behavior Framework (BF). If essentially identical semantics for a given BF term are found under another name in the BF, the BF synonym is noted in the GF term’s definition. A concept map showing the relationships between GF and BF terms can be found at the end of this section.

#### 2.2.1.1 Interoperability

Interoperability is the capability of a set of parties to work in concert to achieve a shared purpose. In the context of the SAIF-CD, it is assumed that at least part of the “work” will involve technology components, standards, etc. Interoperability among parties with different jurisdictions requires a clarification of all boundaries and the means to communicate across them, such that information that originates in one party is able to be understood consistently by another. The IEEE definition states that interoperability is the ability of systems to exchange information and use the information exchanged. How information is used in the receiving system depends on the intent of the exchange. Syntactic interoperability refers to the capability to reliably send and receive information. Semantic interoperability refers to the ability to process the information received with the same understanding of the meaning of the information as the originating system and to use the information received appropriately. Being able to have effective computable interpretation of received information requires a significantly greater codification of meaning than to just reliably display information for a human to interpret.

If information is not commonly understood by the human parties in a collaborating community, the capability of systems being used to support such collaboration will be unable to computationally use the information safely.
and effectively. Since health information is exchanged and subsequently used to directly or indirectly influence the care of people, misuse of information poses a significant risk that must be mitigated.

2.2.1.2 Risk

Risks are adverse outcomes of deliberate acts or external events that are considered of sufficient impact to be actively managed. Types of risks may range from not achieving the shared purpose and objectives, to more profound outcomes such as risking patient safety or violating privacy conventions. Managing risk becomes conscious mitigation strategies to minimize the probability of the risk event occurring or to reduce the impact if the risk does occur. In any shared purpose scenario, working collaboratively across boundaries increases the potential of risks as well as opportunities for mutual benefits. A Risk Profile is the set of organization-specific or community specific risks which have been identified, categorized, and assessed with respect to their Likelihood and Impact to the organization and/or specific development projects – as that profile is viewed from the perspective of shared purpose.

2.2.1.3 Community

[ISO ODP 10746-3] defines community as a configuration of objects formed to meet an objective. The objective is expressed in a contract, which states how objectives can be met by defining roles and interactions required, assignments of objects to the roles, and policies governing their collective behavior.

A community is a set of parties collaborating to achieve a shared purpose. The scope of the community could be across disciplines or departments within a single organization; across organizations within a single geographical area; across geographies that are regulated by different legislation within a single country; or across the world.

A federation is a community of collaborating parties with different jurisdictions that cooperate by agreement to meet shared objectives. The key definitional characteristic of a federated community is that some decisions must be made explicitly in concert, rather than being made autonomously by participating parties. Communal decisions may be made by a central authority made up of members with delegated authority from their respective parties. Clearly, not all decisions need to be made communally, but a clear distinction of which decisions must be made centrally and which may be made locally needs to be explicit, especially those affecting the shared purpose.

2.2.1.4 Party

Party: “A party is an enterprise object modeling a natural person or any other entity considered to have some of the rights, powers and duties of a natural person. (Tyndale-Biscoe, Nov 2002)”

A party is a particular identifiable individual or organization that is expected to participate in one or more communities. A party may be described by its identity or by its general type. Defining participating parties by type requires a mechanism for identifiable parties wishing to participate to be able to express interest and be accepted by the interoperability community, either by consensus, or by meeting preset criteria.

Parties play more than one role and a single role can be played by more than one party. Participation in a community occurs via roles that specify the expected collaborating behavior. A party can participate in multiple communities at the same time, taking on different roles in each community.

2.2.1.5 Jurisdiction

Jurisdiction is the delineation of the boundary conditions of the scope of authority of a party. The boundary is determined by a geographical area and a subject matter or policy scope. Parties have jurisdiction within a particular scope of authority which may be delegated from another party with a higher authority. The relationships between jurisdictions may be implicit or may be codified in regulations or policy. An interoperating community has a jurisdiction of its own that is specified by contract of the agreeing participants.
2.2.1.6 Contract

A contract is a formal agreement among parties to behave in accordance with the policies and processes
accepted by the community in which they participate. The contract clarifies the roles, responsibilities and
policies required to act in concert to meet the shared objectives. A specialized community of parties may be
formed to control the establishment and evolution of the contract. Participants of a federated community
represented by the controlling community agree to the contract by actively participating. The very nature of
interoperability is collaboration among parties who give up some autonomy of decision making within the
scope of activities needed to achieve the shared purpose, but retain autonomy in other aspects of their endeavor.

2.2.1.7 Authority

Authority is the ability of a party to act autonomously. In many circumstances authority to act has been
delegated according to particular policies. The party with the higher authority is a principal and the delegated
party is an agent. Delegated authority from the principal party to the agent usually involves an expectation to be
held accountable for the decisions and actions taken. Automated systems typically act as agents of responsible
parties and carry out predetermined behaviors under specified conditions.

2.2.1.8 Accountability

Accountability is the obligation to take responsibility for actions and to demonstrate that actions are completed
satisfactorily. The responsible party agrees to perform certain actions or to produce certain deliverables.
Accountability means that some mechanism must exist for showing that accepted responsibilities are carried out
and to what extent they are successful. Metrics or reporting mechanisms may become elements of interoperable
systems demonstrating the shared objectives have been satisfied.

2.2.1.9 Role

A role is a collector for the behavior of a party needed to carry out its responsibilities according to a community
contract. A specific name is given to the explicit set of responsibilities that identifies the competence of an
organization, a person or an automated component acting as an agent, to perform specified actions. The set of
responsibilities may include actions that have been delegated from a higher authority. Behavior is further
refined into specific actions that may become operations in an automated system.

2.2.1.10 Responsibility

Responsibilities are explicit behaviors or actions associated with a community role. Responsibility for acting is
stated as a permission (you may act), an obligation (you must act), or sometimes as a prohibition (you must not
act), including the conditions under which each action is valid.

While a party in a particular role is expected to be competent to perform all specified actions or behaviors, some
actions may have resource availability or other pre-requisite conditions to be met before they can be performed.
The measure of a role’s ability to act is considered to be the capability of a role. The amount of action due to
resource availability is capacity. Resources can include space, equipment, supplies, specific information or
simply time availability of a party in a particular role.

2.2.1.11 Provenance

Provenance is a term borrowed from the antiques industry. It referred to the documentation of what ownership
a particular antique item has had over time. In the SAIF context, provenance refers to the documentation that
identifies the jurisdiction of the source of each conformance statement (or the artifact containing a group of
them) in a specification, from that statement’s origination as documented requirements to implementable
specifications for technical components. The history may be included within a specification or by reference to
an external artifact.

Provenance may also refer to the auditable history of the context of information that originates in one system
and is used in another, including any transformations that occur along the way. The term Provenance may also
be used for other metrics to identify expected recording of actions taken for accountability purposes.
2.2.2 Governance Language

The Governance Framework language is made up of four interdependent concepts, which taken together define what the rules are, who makes the rules, what processes are needed to implement the rules and how the rules are measured or enforced. The following structure is based on that recommended by the book “SOA Governance: Governing Shared Services On-Premise and In the Cloud” by Thomas Erl, Robert Laird and Robert Schneider.

Governance system design must consider all four together. A tabular structure is a convenient template, although actual documentation styles can vary considerably, as long as the specific concepts are linked.

<table>
<thead>
<tr>
<th>Precepts</th>
<th>People</th>
<th>Processes</th>
<th>Metrics</th>
</tr>
</thead>
</table>

Figure 7 Governance design documentation template (from Erl et al, 2011)

2.2.2.1 Precepts

A precept is an authoritative rule of action. Precepts are the essence of governance because they determine who has authority to make decisions, establish constraints for those decisions, and prescribe consequences for non-compliance.

Precepts codify decision making rules using four “sub-dimensions” or “characteristics describing a given precept”:

- **Objectives**, which broadly define a precept and establish its overarching responsibility, authority, and goals
- **Policies**, which define specific aspects of a precept and establish decision-making constraints and consequences in terms of permissions, prohibitions, obligations or authorizations
- **Standards**, which specify the mandatory formats, technologies, processes, actions, and metrics that people are required to use and carry out in order to implement one or more policies
- **Guidelines**, which are non-mandatory recommendations and best practices

2.2.2.2 Processes

A process is a collection of steps taking place in a prescribed manner and leading to an objective. A step may be associated with multiple roles. Every step shall have one or more actors.

It is important to make a distinction between governance processes and other types of processes. Governance processes provide a means to control decisions, enforce policies, and take corrective action in support of the governance system. Governance processes are further elaborated in the section below.

Other processes, such as those employed to carry out the intended purpose, can be heavily influenced by governance precepts, but are not specifically processes that are directly related to carrying out the governance system. The BF may be used to specify these additional processes. Technically, any process is considered a management activity, but a governance system is dependent on governance processes to ensure compliance with its precepts.

A community is likely to use a variety of processes to support its precepts. Some may be automated, while others require human effort. Automated processes can help coordinate tasks (such as steps required to collect data for approvals), but can still rely on people to make important decisions (such as making the actual approvals based on the presented data).

2.2.2.3 People (Roles)

People (and groups of people) make decisions in accordance with and within the constraints stipulated by governance precepts. For a governance system to be successful, people must understand the intents and purposes of the precepts and they must understand and accept the responsibilities and authorities established by
the precepts. Governance systems are therefore often closely associated with an incentive system. This allows the community to foster a culture that supports and rewards good behavior, while also deterring and punishing poor behavior.

When exploring the involvement of people in relation to governance systems, it is further necessary to identify the role or roles they assume. Community roles position people (and groups) in relation to governance models and further affect the relevance of precept compliance and enforcement.

There are two ways that people can relate to precepts and processes: they can help author the precepts and processes and they can be dictated to by their application. Opportunity for those affected by the precepts to provide feedback to the authors is recommended.

Other entities can take on roles in specifications involving non-governance processes, but only people can participate in governing processes.

2.2.2.4 Metrics

Metrics provide information that can be used to measure and verify compliance with precepts.

The use of metrics increases visibility into the progress and effectiveness of the governance system. By analyzing metrics, we can gain insight into the efficacy of governance rules, and we can further discover whether particular policies or processes are too onerous or unreasonable. Metrics also measure trends, such as the number of violations and requests for waivers. A large number of waiver requests may indicate that a policy might not be appropriate or effective.

The ECCF describes specific types of metrics as conformance statements that are used to determine whether technology components can be certified to fulfill the behaviors specified.

2.2.3 Governance Processes

The processes to establish and maintain precepts and their related components are different from the processes defined within the context of each precept. The governance processes are all about what it takes to make the rules, communicate what the rules are to all interested parties, make exceptions to the rules and evaluate and change the rules when circumstances change or more effective rules are identified.

2.2.3.1 Definition Processes

The definition processes are those by which a precept is established, agreed to and then maintained as feedback on its use is provided. The workflow may include approval for establishing a new precept, authoring a definition and related components, approval for use, deployment into the environment of use, evaluation for relevance and efficacy as circumstances change, and subsequent ratification, revision, replacement, or retirement.

2.2.3.2 Communication Processes

Communication processes about precepts and their related processes and metrics are needed to inform the people expected to follow the processes. Various forms of communication channels may be necessary to raise awareness, clarify specifics, gain agreement and then hold people accountable. Awareness of risks and their consequences, rationale for selecting the specific precepts and their processes and metrics, and support for executing them may also be needed. Tools and other resources that minimize the effort required to comply will increase buy-in. Training for active participants in the processes is also likely to be necessary.

2.2.3.3 Appeal Processes

Appeal processes and transition strategies permit precepts to be overturned or modified by exception. Time-limited dispensations to do something other than what the precepts expect can ease transitions and avoid unnecessary disruption. However, the precepts are intended to reduce risk, and accepting appeals means a conscious decision to accept the increased risks.
2.2.3.4  Revitalization

Every precept and its related components should be evaluated periodically to determine if the related risks are being mitigated effectively, whether the precept is still relevant to the current circumstances, or whether there are possible alignments necessary among interdependent precepts to avoid gaps and confusion. Feedback from related metrics and appeals may be used, as well as evaluation of any rationale or assumptions identified when the precept was defined. New roles, technology opportunities or resource constraints may suggest a review of related precepts. In many ways, changes in circumstances require revisiting governance. Also, changes in governance may cause ripple effects in any automated application that is involved in precept execution.

2.2.4  Relationship between the Governance Framework and the Behavioral Framework

The Governance Framework provides the language for defining the specifics of the various organizational and technical development activities that must be defined, executed, and managed via overarching governance processes to reach agreement on a shared purpose and how to collectively achieve that purpose in the context of one or more defined cross-boundary scenarios. In contrast, the Behavioral Framework provides the language to describe the various contracts, transactions, and processes – at a technical level – which are necessary to produce a technical realization of previously specified shared purpose scenario. The languages defined by the GF and BF are similar in overarching motivation. However, each has a somewhat different focus and emphasis. Following are two lists the first which identifies terms defined in both the GF and BF but used in different contexts within the two languages, and the second listing terms mentioned in the GF but defined in the BF.

Terms defined in both GF and BF
- objectives
- policies
- contracts
- communities
- roles
- processes

Terms mentioned in GF but defined in BF
- operations
- obligations
- objects
- permissions
- prohibitions
3 Behavioral Framework

3.1 Purpose

Figure 8 BF language concepts and relationships for describing contract semantics.

The purpose of the Behavioral Framework is to provide the language necessary to explicitly and unambiguously define dynamic semantics used to specify the behavior of enterprise objects involved in shared purpose scenarios. The BF language is meant to be used in combination with the IF language (which focuses on explicit expression of static/informational semantics) – to fully specify the details of the various roles, responsibilities, capabilities, expectations, accountabilities, etc. of a given object as it is involved in these scenarios. The BF semantics can be grouped together into three categories (see BF Overview Concept Map):

1. **Contracts.** These semantics help to define enterprises as composed of objects (people, organizations, technical components, etc.) organized as communities with certain business objectives, leading them to create agreements called contracts in order to specify their behaviours. The fundamental unit used within the contracts to specify desired behavior is the service, organized following Martin Fowler’s accountability analysis pattern, such that each service explicitly identifies the responsible and commissioning roles. [In particular, the Conceptual Perspective of the SAIF-CD, the BF language surrounding contracts serves – via the use of similar (and often identical) language – as a link between an organization’s negotiated shared purpose and the technical realization of that shared purpose in technical architectures and their associated components.]

2. **Operations.** These semantics break down the details of the information exchange between the roles within a service, organized around the concept of a basic unit of exchange called operation. [The
3. **Processes.** These semantics allow organizations to define complex interactions composed of multiple operations involving potentially many different services and roles.

The three categories of BF semantics do not exist in separate, mutually exclusive realms. Rather, the above categorization is primarily created as a cognitive aid in assimilating the BF language, and secondarily based on the source of the language (contracts and operations coming primarily from RM-ODP, and processes coming primarily from BPMN2). Overall, direct relationships between the concepts are more likely to exist within each category, with a small number of bridging relationships across the categories. In particular, the service concept acts as a bridge between contract and operation semantics, since service is the mechanism used to describe behavior in a contract, and operations are used to specify the details of the interactions within a service. Roles bridge contract and process semantics, since roles are what binds particular enterprise objects to their behavior within a contract, and roles also are used to specify the participants in a process. Finally, operations themselves act as the link between operation and process semantics, since the individual steps in a process which require interactivity between two roles are specified as particular operations of a service.

Shared purpose scenarios are often initially defined at an organizational level and then subsequently manifest at a technical level. The SAIF-CD recognizes this “problem space” vs. “solution space” topology through its use of Perspectives of the Interoperability Specification Matrix (ISM). In particular, the ISM’s Conceptual Perspective represents the problem space view of a given component and is outward facing toward the larger issues of a given organization and its various shared purposes. As such, the BF language applied to the Conceptual Perspective usually focuses on the Enterprise Dimension. In contrast, the ISM’s Implementable Perspective represents the solution space view of a technical component as a realization of the organization’s shared purpose requirements. Finally, the IMS’s Logical Perspective serves as the traceable bridge that links the problem space with the solution space. The concepts defined in the BF language in many cases will have distinct manifestations across the different perspectives, but the BF does not try to create separate concepts for each of the perspectives as this exercise will result in unnecessary redundancy at the canonical level. For example, an enterprise might need to specify a particular enterprise level contract defining business services between real world parties, and its corresponding technical contract to be realized in a particular implementable technical service. The SAIF-CD leaves it to the SAIF IG grammars to explicitly define the distinctions between services, contracts, roles, etc. across multiple perspectives and their correspondences.

The BF language is architecturally neutral in the sense that it allows component designers and developers to unambiguously discuss contracts, isolated operations, and amalgamated processes independent of their particular choices of implementation architectures, modeling constructs, etc. Thus, the BF language can productively be used to define the behavioral semantics of shared purpose scenarios involving any one of a number of interoperability paradigms including messages (e.g. as implemented using various flavors of HL7 messages), services (e.g. as modeled using SoaML or the OASIS SOA Reference Model and implemented using SOAP or REST technologies), or documents (e.g. modeled in HL7 CDA, openEHR archetypes, or 13606 containers). Modeling, design, and implementation paradigms such as these are specified in organization-specific SAIF-CD-compliant SAIF Implementation Guides (SAIF IGs).\(^1\)

---

\(^1\)The BF adopts and adapts RM-ODP (ISO RM-ODP) as a reference model. On one hand, the BF uses a small set of ODP modeling concepts which were found central for defining distributed components from the perspective of achieving shared purpose through interoperability. On another hand, the BF adds further level of detail such as a set of concepts from the BPMN2 metamodel to model processes. It also adds a small set of concepts to facilitate the distinction between conceptual and logical perspectives. The languages defined in the ODP and SAIF-CD are abstract and therefore require elaboration and instantiating in specific SAIF IGs, e.g. through the use of representational grammars such SoaML, UML 2.3, UML profile for ODP, etc.
3.2 Contract Semantics

The BF contract semantics define the idea that enterprises are composed of objects, which could include either real world entities as well as IT systems. Objects are organized into communities, with objectives that include shared purposes requiring some degree of interoperability. In order to achieve these objectives, communities establish contracts between their objects specifying their behaviors. The ability to properly specify these behaviors in order to achieve interoperability is the main topic of the BF language. Agreed upon behaviors in a contract are organized along the abstract analysis pattern known as accountability [cite Fowler], which states that there is an agent responsible for the behavior and an agent that commissions the behavior. In BF contract semantics this accountability is known as a service and the contract allows each object to fulfill the role of commissioning or responsible agent for specific services. Contracts can be further constrained by policies, which can be in the form of prohibitions, permissions, and obligations.

The terms of art (in bold in the previous paragraph) defined by the BF language are taken primarily from the RM-ODP foundations (ISO, 2010) and enterprise language (Tyndale-Biscoe, Nov 2002). The concepts included from ODP were chosen because of their collective expressiveness in describing key organizational and policy concepts, in a way close to their natural language expressions.

The emphasis is not on supporting the description of social concepts such as acts, roles and entities for the purpose of recording information in a system—as such, these terms should not be viewed as synonymous with HL7 RIM terms (for example) – but more broadly to describe enterprise objects that will be involved in instances of shared purpose scenarios. Many of these concepts have analogues in the GF, a reflection of the fact that the shared purpose semantics that are ultimately expressed at the technical component level via component-

Figure 9 BF language concepts and relationships for describing contract semantics.
to-component interoperability are initially determined at an organizational level. In general, readers of the SAIF-CD can view the GF as outward facing, i.e. directed toward the problem space, whereas the BF is more inward facing, i.e. directed toward the solution space. These are not absolute constraints. What follows is a detailed set of definitions for these terms.

**Contract**: An agreement governing part of the collective behavior of a set of objects. A contract specifies, for each object involved, the different roles they may or must assume. Contracts may also specify policies for the objects, quality of service requirements, indications of duration or periods of validity, behavior which invalidate the contract, liveness (OWICKI, 1982) and safety conditions.

**Object**: A model of an entity (entity is defined as any concrete or abstract thing of interest). An object is characterized by its behavior and its state. Objects are the subjects of a contract and fulfill particular roles in services and processes. Note that the concept of object is broader than the traditional notion of software objects or business objects used in building object-oriented and enterprise systems. It is a model of any entity.

**Community**: A configuration of objects formed to meet an objective. This objective is expressed in a contract.

**Role**: Identifier for a behavior, which is to be fulfilled by an object as part of a contract. Specifically, the BF requires each role to be associated with a service either as a commissioning or a responsible agent. Roles are also the identified participants in a process.

**Service**: A related set of behaviors that add value by creating, modifying, and/or consuming information, involving collaborations between a responsible agent (the service provider), who expresses some guarantees, and commissioning agent (the service user or consumer), who receives the guarantees. The collaborations may involve a complex series of interactions, organized along operations. In a contract, roles fulfilled by particular objects identify who act as the responsible and commissioning agents.

**Policy**: A set of rules applied to a particular purpose. Policies are included in contracts, but may also be applied to many other objects or concepts in any of the dimensions.

**Obligation**: A prescription that a particular behavior is required. An obligation is fulfilled by the occurrence of the prescribed behavior.

**Permission**: A prescription that a particular behavior is allowed to occur. A permission is equivalent to there being no obligation for the behavior not to occur.

**Prohibition**: A prescription that a particular behavior must not occur. A prohibition is equivalent to there being an obligation for the behavior not to occur.

Note: A specific grammar instantiation of the BF language can provide a specific way of defining structuring, behavior and policy aspects of the community (for example, the use of the OMG SBVR notation), add further level of detail to the concept of objective (for example, the use of OMG Business Motivation Model) and so on.

### 3.3 Operation Semantics

![Figure 10 BF language concepts and relationships for describing operation semantics.](image-url)

The BF operation semantics provide a way to specify and organize the information exchanges required for interoperability, specifically the exchanges between the responsible and commissioning roles of a service. The
basic meaningful unit of information exchange is the **operation**, which may necessitate one or more
**interactions**. As an illustrative example, a laboratory results service might include an operation to retrieve a
result given a patient and accession number. This particular operation might involve two interactions, the query
from the commissioning role including the patient and accession number parameters, and the result answer back
from the responsible role. Operations, in some HL7 contexts have also been called “transactions,” but SAIF-
CD prefers the RM-ODP term because “transactions” in a different context (i.e., database systems) imply
specific ACID conditions, including ability to rollback, that are not meant to be part of this concept. An
operation is fully described by its **signature** (which specifies the interactions involved), pre-conditions, **post-
conditions**, and **exception conditions**. Each service provides one or more operations, grouped together into
**interfaces**, which define a specified subset of the total set of operations in a service. This subset serves as a
conformance point in specifications.

The terms of art (in bold in the previous paragraph) defined by the BF language are taken primarily from the
RM-ODP computational language (ISO RM-ODP). In RM-ODP operation is a special kind of interaction, the
others being signals and streams. SAIF-CD maintains the simplicity of a single construct (operation) as the
basic unit of defined behavior, allowing the SAIF IG grammars to specify more varieties based on the needs of
the particular enterprise. The following are the definitions of the concepts introduced by BF operation
semantics:

**Operation**: The smallest unit of behavior, involving information exchange between commissioning and
responsible roles in a service, which provides business value. Operations are specified by their signature, pre-
and post-conditions, and exception conditions.

**Signature**: The precise definition of the interactions involved in an operation, including attributes such as
direction, optionality, and content.

**Interaction**: An atomic piece of information that is transmitted in one direction from an object to another. One
or more interactions must exist together in the context of an operation for there to be business value as part of
the information exchange. A single interaction that is part of a larger operation provides no business value in
isolation, for example, a query without a response.

**Pre-Condition**: a predicate that a specification requires to be true for an operation to occur.

**Post-Condition**: a predicate that a specification requires to be true immediately after the occurrence of an
operation.

**Exception Condition**: exists when an operation fails to fulfill its service guarantees

**Interface**: A grouping of operations of a service required to be implemented together in a specification.
3.4 Process Semantics

The BF process semantics allow for complex behaviors known as processes, which potentially include many different service operations in a sequence, involving multiple participants defined as roles. The sequencing and relationships between the multiple behaviors of a process are described using a set of flow elements, which usually correspond to elements of a particular notation. Although the key concepts in BF process semantics come from the BPMN2 metamodel, the full BPMN notation would be considered a grammar, and its use, if desired, would be specified by the SAIF IGs. The concepts used in the BF language are abstract enough such that a particular SAIF IG may choose grammars other than BPMN and still be SAIF-CD compliant. The main flow elements of the process, specifying the action steps, are activities, which are carried out via service operations when they require information exchange between process roles. Sequence flows are flow elements that determine the sequencing of activities in a process. Events are flow elements that represent triggers or results of activities. Another flow element is the gateway, which serves to organize options and parallelism in sequence.

Process: A collection of steps (defined as activities) taking place in a prescribed manner and leading to an objective. Contracts may specify the participants involved as roles in the process, corresponding to the roles in all the services for which operations may be invoked over the course of the process.

Flow elements: The units used to describe the process and its sequence of steps. In a SAIF IG grammar, the flow elements usually correspond to elements in a particular process description notation.

Activity: A process flow element that represents a step of work to be performed. An activity can be composed of further smaller activities, and described as a sub-process (SAIF IG grammars will determine precisely how this decomposition is to be expressed). Any information exchange that is necessary for an activity must be explicitly carried out as a service operation.

Event: A process flow element that represents some kind of occurrence (“something” that happens), which in turn causes an activity to occur (a trigger) and/or occurs as a consequence of an activity (a result).

Sequence flow: A process flow element that determines the ordering and progression of activities in a process. Typically, a process notation specified in a SAIF IG might denote sequence flows as lines and arrows connecting the activities.

Gateway: A process flow element that controls the divergence and/or convergence of sequence flows. It allows branching, forking, merging, and joining of process flow.
4 Information Framework (IF)

4.1 Purpose
The Information Framework chapter defines the language describing the various artifact types and inter-relationships of the Informational Viewpoint from the three SAIF Perspectives. The concept map below provides an overview of the IF language.

Figure 12 Information Framework Concept map

4.2 Goals
The goal of the information framework is to describe how the static information of importance to a given domain and the experts within that domain is captured and refined through a traceable process to yield an implemented or implementable information artifact. This implementable information artifact, when developed using the methods defined in this framework, delivers the static semantics that contribute to the definition of computable semantic interoperability between systems. The information definitions contained in these artifacts are reusable, and given the appropriate level of enterprise governance in the process of model development, yield consistency across the range of information modeling and terminology binding tasks encountered within an organization.

4.3 Data and Information
Data is the raw material from which information is derived. In order to allow information systems to use data to address most healthcare use cases, we must first convert it to information.

A simple natural example gives us a basic understanding of this conversion process. For example, let’s take images. Light is transmitted through the lens of the eye and focused onto the fovea of the retina where rods and cones transmit the photons of light energy to the visual cortex of the brain, interpreting and preserving color and contrast. The light is processed, its intensity determined, the directionality from the source is noted and the light with context is integrated with the visual context and referenced against other historical information stored in
the brain. All of this data is put into context and thus can be used as information to interpret the raw photons and to assess the light as an image, either of beauty, threat, unclassified wonder, etc.

The parallel information technology process is the capture of a digital image through the lens of a camera. In this case, the photons are focused onto a sensor by the camera lens. The sensor stabilizes the image, activates specific chromatic sensors to determine color, and passes the information to a processor to generate the image in one of a number of possible mime types. Thanks to the standardization of the processing and use of standard mime types, these images can then be used by a variety of applications for a variety of purposes with no loss of information (this is dependent on the mime type used since some are lossy).

Streaming data across an enterprise is no more useful than streaming photons without the processing enabled by the rods and cones of the retina or the processor in a digital camera. There must be context provided so that the data can be used as information for a useful purpose, or rather, a meaningful use in today’s healthcare parlance.

We therefore can say that information is "data in context". Hence the SAIF Information Framework Book is about putting data into a context that information systems can properly manage and apply data for useful purposes. It is the context of data and its unambiguous organization into a hierarchy of information models that provides the properties of semantic interoperability when shared with other information systems. The more a system adheres to the SAIF principles, the more interoperable that system will be with a wide range of other systems that also apply the SAIF principles.

This document is meant to lay out those principles in their canonical form so that these principles may be used across a wide range of implementations and hence is agnostic to the eventual implementation language or model persistence.

Information Framework Components

Note that it is assumed that each artifact described in the IF is associated with a versioning scheme particular to that artifact though this is not explicitly discussed.

i. Concepts and concept organization
   - Un-encoded concepts

ii. Data types

iii. Class objects
   - Terminology binding

iv. Information Models
   - Templates
   - Executable Models
   - Conceptual Information models
   - Domain models
   - Logical Information Models

v. Summary

4.4 Concept Component

A concept is the basic unit of data used in communication and each concept represents an atomic unit of thought that references a concrete or abstract thing. Concepts are organized into terminologies and these terminologies have specific models that define how the concept metadata is described and what, if any, rules can be applied to the concepts to create more complex concepts out of simpler concepts. Note that a concept is only unique within
a coding scheme and that mappings between coding schemes is possible and even common. The simpler concept is called a primitive concept and the more complex concepts formed by the combination of two or more concepts are called coordinated concepts. When a concept expression is built compositionally using two or more primitive concepts within the terminology then it is said to be "pre-coordinated". When the concept expression is constructed outside the terminology such as within a medical record system using the primitive concepts of the terminology, it is said to be "post-coordinated".

This allows a more precise definition of a concept that improves the chances of semantic interoperability between partners.

<table>
<thead>
<tr>
<th>Primitive Concept</th>
<th>Pre-coordinated Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pneumonia</td>
<td>Right lower lobe Streptococcal pneumonia</td>
</tr>
<tr>
<td>233604007</td>
<td>233604007</td>
</tr>
</tbody>
</table>

Figure 13 Example of concepts

There is often a tension between expression of the semantics of healthcare using the terminology model constructs versus the information model constructs and there is no clear answer as to where to draw the line between these in all cases. Complex expressions may be developed in either a terminology through the use of post-coordination or may be expressed using a combination of simple concepts with relationships of these concepts defined by the information model that contains these concepts. These complex terminology constructs can be isomorphic with the simple concepts represented as an information-modeling artifact. There have been several efforts to develop formal rules to help formulate guidelines for defining a boundary between terminology and information models including the HL7 TermInfo project and the UK National Health Service Logical Record Architecture. As more experience with these activities is gained in actual deployments, future versions of this document will address these findings.

4.5 Controlled Terminology

The purpose of a terminology is to provide a clear and unambiguous way to describe concepts so that two or more individuals can gain a shared meaning of those concepts. A concept is the basic unit of communication and each concept represents an atomic unit of thought that references a concrete or abstract thing. A controlled terminology provides the organizational framework for concept ordering, inheritance and rules that govern the use of the concepts. For example, Jim Cimino described several rules that a sound controlled terminology should adhere to. These include vocabulary content, concept orientation, concept permanence, non-semantic concept identifiers, poly-hierarchy, formal definitions, rejection of "not elsewhere classified" terms, multiple granularities, multiple consistent views, context representation, graceful evolution, and recognized redundancy {Cimino, 1998 #94}. (NOTE: The degree to which a given SAIF IG may require these particular attributes in terms of bindings to terminologies is, in fact, an IG-specific decision. The concept of Controlled Terminology is part of the SAIF-CD descriptive language for specifying informational/static semantics.)

The concepts can be expressed in a number of ways. Common expressions of a concept may be verbal, symbolic, textual or coded. Once a concept expression is agreed upon it can be used for the purpose of interacting with trading partners that need to share information.

In verbal communication of these terminological concepts, the spoken language and, in some cases the dialect and inflection, must be known by the communicating parties. Often times those terminological concepts may
have multiple meanings depending on the context in which they are used, even when the spelling in a given language is identical. Therefore, the textual representation of a concept is inadequate to completely provide the meaning of a term when it is separated from its context of use.

Information systems depend on an explicit and unique meaning of a concept and hence cannot rely on verbal or textual representations of concepts. Textual representations may be misspelled, abbreviated, or expressed in a different language with different spellings as the example below shows.

![Figure 14 Example of alternative text for a concept](image)

<table>
<thead>
<tr>
<th>Concept</th>
<th>Streptococcal Pneumonia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternate spelling</td>
<td>pneumonia</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>S Pneumonia</td>
</tr>
<tr>
<td>Misspelling</td>
<td>Streptococcal pneumonia</td>
</tr>
</tbody>
</table>

Figure 14 Example of alternative text for a concept

Concepts must be encoded with unique identifiers in order to disambiguate identical textual or verbal representations of different concepts. These encodings must be unique within a given code system or namespace. There is no guarantee that the code value is unique across other terminology namespaces and in fact there are many instances where the coded representation of a concept is reused across different terminology namespaces. The table below shows a small part of the 921 LOINC and CDC Race and Ethnicity codes that overlap. Without knowing (and sending) the code system with the code, there is risk that ambiguity will exist once the data is subject to query.

![Figure 15 Concept overlap](image)

<table>
<thead>
<tr>
<th>LOINC NAME</th>
<th>LOINC Code</th>
<th>Race Code</th>
<th>RaceName</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCG, Urine</td>
<td>2100-3</td>
<td>2100-3</td>
<td>White</td>
</tr>
<tr>
<td>HCG, Blood</td>
<td>1960-4</td>
<td>1960-4</td>
<td>Tlingak</td>
</tr>
<tr>
<td>HDLc, Serum</td>
<td>2085-9</td>
<td>2085-9</td>
<td>Micronesian</td>
</tr>
<tr>
<td>Insulin 2H, 75 % Glucose</td>
<td>1564-4</td>
<td>1564-4</td>
<td>Scott Valley</td>
</tr>
<tr>
<td>Insulin 3H, 75 % Glucose</td>
<td>1567-7</td>
<td>1567-7</td>
<td>Big Cypress</td>
</tr>
</tbody>
</table>

Figure 15 Concept overlap

Coded concepts are used as a) structural vocabulary or b) descriptive vocabulary. Structural vocabulary is used to describe the model elements that carry the descriptive vocabulary that is used at the instance data of a model.

Finally, vocabulary can be divided into those terms used in the “model of meaning” and those used in the “model of use” as described by Rector (Rector, Rogers et al. 2004). The model of meaning is that model supplied by the definitional structure of the controlled terminology that defines the concepts through either formal definition (description logic for instance) or informal definitions in text including the fully specified names. The model of use describes how a terminology is actually deployed in an electronic health record or other application that includes the grouping into pick lists or value sets, the ordering of the concept presentation, and the display names of those concepts.
4.6 Un-encoded concepts

Not all concepts received either in messages or as service payloads will be encoded in a specific terminology. In many cases the concepts will be included as literals, i.e. not bound to any specific terminology or code system. These are often referred to as “free text” entries. There are several ways to process these entries including natural language parsing, storage as native text entries or conversion to lingual interpretations that can be machine processed.

One of the methods of taking free text entries and converting them to machine process-able data entries is via the ISO 24707 Common Logic specification. While literals (also called sentences in logic languages) can be converted to machine process-able data entries, the process requires an understanding of first order logic.

Common Logic Controlled English Entry: **John goes to Boston by bus.** This entry is called a “sentence” in common logic.

This sentence may be expressed in a machine interpretable format via common logic in the following graphic.

Conceptual graph display form:

![Conceptual Graph display Form](image)

(Figure Reproduced with permission from J. F. Sowa)

**Conceptual Graph Interchange Format (CGIF):**

[Go *x] [Person John] [City Boston] [Bus *y]

(Agent ?x John) (Dest ?x Boston) (Inst ?x ?y)

**Common Logic Interchange Format (CLIF):**

(exists ((x Go) (y Bus)))

(and (Person John) (City Boston))

(Agent x John) (Dest x Boston) (Inst x y))

This syntax is not familiar to most developers and hence is included here as a mechanism for further study of ways to construct logic statements to handle free text or literal entries.

For further discussion about how Common Logic is used to form a semantic formalism between logic languages such as Z, UML and RDF or OWL, please see the works by John Sowa on Common Logic and Conceptual Graphs. **Error! Reference source not found.**
4.7 Concept Grouping

4.7.1 Code Systems

There are several ways to organize concepts for models of use. The collection of all concepts in a particular terminology is called a coding system or more simply, a code system. Some code systems contain only the concepts that describe like or similar concepts. This set of “similar concepts” is referred to as a “semantic type”. Examples of code systems that contain concepts of a single semantic type include the CDC Vaccines Administered code system (CVX) and the Standard Occupational Codes (SOC) code system that defines occupational categories. Other code systems have many semantic types defined in non-overlapping subdivisions, the prime example being SNOMED CT where top level categories include products and geographical locations as well as clinical findings or procedures.

4.7.2 Semantic Types

The semantic type is a category for an item or group of items (concepts in our case) that all share a similar meaning (semantics) as defined for that group. The semantic type can then be used to distinguish the use and purpose of different items in the group. Examples of semantic types taken from the National Library of Medicine’s Unified Medical Language System (UMLS) include virus, fungus, laboratory test and professional society, all placed into a hierarchical structure. It is common to refer to a reference set of semantic types as fillers for an attribute of the abstract information models such as Conceptual Information Models. In this case it is inappropriate to define specific codes or code systems from which these semantic types might originate so that the Conceptual Information Model maintains maximal reuse capability and subject matter expert familiarity. Being able to refer to a semantic type as the appropriate concept group for an attribute allows a domain expert to provide requirements in their language and allows a terminologist downstream in the development process to assign appropriate code System content to that abstract semantic type.

4.7.3 Value Sets

Typically a set of concepts is organized into a group that can be used as fillers for a field in a data entry form. The set of concepts used for this purpose is referred to as a value set. A value set need not draw all of its member concepts from a single code system. The life of a coded concept does not end when the submit button is depressed and the data element is stored in the database. The data will almost always have a secondary use and in order to use that data appropriately, it must be stored with the appropriate metadata to understand the coded concept in context. This will include enough metadata to resolve the exact value set membership at a given point in time, namely at the time the user submitted the data. This means that a value set member must be stored with the date of the value set creation and some unique identifier for the value set. When this value set is ordered in a particular way for optimal use in an interface, it is often called a pick list. There is psychometric evidence that the ordering of a concept in a pick list is important in evaluation of data input and this metadata may be optionally stored as well {Sudman S, 1996 #257}. This attention to value set membership is necessary to enable valid longitudinal analysis of data. Without this metadata it would be impossible to know what coded concepts a user could have chosen from as a response in a form field, hence data would not be comparable over time as the choices could have been changed by addition or deletion.

4.8 Data Type

A data type is a data storage model or template that defines the attributes for a specific type of value or range of values. It acts to formalize the requirements for data of specific types so that the receiver knows all of the attributes needed to process the data.

Data types may be simple where the attributes of the data type each hold only a single data value (primitive types) or they may be complex where the attributes may hold a pointer to other data types that hold the actual data values. The more complex data types may also have a mechanism to define constraints on the data type so that an abbreviated set of attributes may be sent and a processor can still validate the contents of the constrained type without requiring all attributes to be populated. In this way a single data type definition can satisfy multiple
use cases. This constrained data type is called a data type flavor. Note that other constraints may include valid ranges, optionality and cardinality.

Data types can be grouped into a set of canonical types. The canonical data types are classified as nominal, ordinal, quantitative, narrative text or image mime types. Nominal types express a categorical response that does not have a natural ordering. This includes names of entities or simple observations of natural phenomenon such as color or consistency for example. Ordinal values express concepts that have a natural order. Examples of ordinal values include grades such as A-F and sizes such as small, medium and large. Quantitative types include numerical values expressed as ratios, integers, real numbers or ranges that have a mathematical interpretation. Narrative text data types are used to express descriptions in natural language. Finally, there are types of information that are typically symbolic to human interpretation but may be processed by machines as digital data. Examples are radiology images, digital wave forms and gel electrophoresis patterns.

4.9 Classes

A class is a collection of attributes that pertain to a specific encapsulated concept. Note that this definition includes UML classes, OWL classes, and other more loosely defined things such as SNOMED-CT concepts. For example a person can be described by a set of attributes that are always reflective of fixed properties of a human being. The properties include a date of birth, a genetically determined gender, a race to which the person belongs and an ethnicity that reflects an ancestral population group. Attributes have properties that control their use and possible values including their type and are collected into an information structure called a class that can be used as a component of larger information models. Classes have relationships to other classes and relationships have properties of their own such as whether they are monotonic (1:1) or open ended such as 1: many or 0: many. The data elements of a class - attributes and relationships - may be formally defined in the context of a framework such as ISO 11179.

Classes are defined within the context of an information model (see below) that provides the context in which they are understood and used.

4.10 Terminology binding

Attributes of a class can be coupled with the set of concepts used to describe the possible values of that attribute. This identification of the concept fillers for a given attribute in a given class is called terminology binding. The binding at the class level is broad and can usually best be done with a semantic type rather than a value set until such time that the class is used incorporated as a component of a specific information model that is to be used for a specific data purpose in a specific domain. For example, I could have a laboratory class with a result value attribute. When the class is unbound to a specific information model, we can only say that the terminology for that attribute will come from some data set that can express a lab value. That data set might be an ordinal type, a narrative type or a nominal value for example. If I now include my class in a specific information model where I know the only result values that I will get are blood types, I can bind the attribute to a specific value set that contains all of the human blood types and no other values are possible.

4.11 Information Models

Information models represent a collection of classes and the relationships between those classes. The relationships may be classes themselves in more complex modeling methods and are reflective of a specific domain of discussion. In other words, the relationships between classes are not static from information model to information model and change depending on what behavior (or larger concept) the model is expressing. Information models for a given domain may be subdivided into small, reusable sub-models. This is a useful way to provide consistency of class relationships that are common across information models. An example would be the physical address class relation to an entity class which is always a static relationship since a physical entity always occupies some physical location. There are many examples of the small, reusable models in healthcare modeling.

Information models may be UML class or instance diagrams, constraint statements on some other model, ontologies, or terminology models. Information models may be expressed against many underlying definitional
frameworks, or none at all (e.g. concept map); which is appropriate depending on the use to which the model will be put.

Information models may be concrete where they define a specific set of classes with specific relations and specific terminology bindings or they may be abstract where the classes have optionality to the classes they are related to and the terminology is not set by bindings of specific values. These abstract models can be used to define information requirements from which more specific constrained information models are derived.

Useful information models are internally consistent in several senses, including their semantics and their engineering methodology; building these models is challenging. Several different methods may be used to build such models. The classic method is specialization of a class where the parent class has only the necessary and sufficient attributes to define that parent and the children classes add attributes to define specialization of the parent class. This approach favors implementation consistency over semantic consistency. An alternative is to constrain an abstract parent class that contains a superset of all attributes of a class type. This approach favors semantic consistency over implementation consistency.

Below are two examples of demographic information models. The first example is the Person archetype of the Demographic Information Model from openEHR.

![Diagram of openEHR Person Demographic Information Example](openEHR_Foundation, 2001-2007)

Below is the second example, which is the E_Person universal (COCT_RM030200UV08) CMET from (Health Level Seven International, Inc., 2011).
Building such models consistently is a challenge. Adding attributes to classes based on an ad-hoc empiric analysis of a particular domain of discourse is fraught with inconsistency, incompleteness and intense effort and is unlikely to lead to semantically interoperable models (e.g. modeling domains based on ISO 11179 alone with no additional methodology). This is because there is no overarching information model to guide the developers of these “common data elements” in a consistent way and hence each model may be developed via the understanding of a different observer rather than via a guiding information model of the domain. The forms of models described below (Conceptual Information Model and Reference Information model) introduce consistency across the information models and lets one construct a Logical Information Model that is faithful to the business requirements and to the reference information model.

4.11.1 Reference Information Model

A reference information model is a formal model of an entire domain of discourse. It serves as a guide or pattern for all derived concrete classes of a domain or sub-domain of interest. A reference information model is essential to the development of a consistent representation of specific information models in a domain of discourse. It allows for the interpretation of relationships of sub-domains to each other, helps us understand the relationships between artifacts in an information model derived from the reference information model, and allows for the consistent definition of information artifacts and therefore consistent use. It helps to avoid the “re-invention of the wheel”, such as multiple different interpretations of the same concepts in different contexts, by providing a framework that leads a modeler down a well-worn path. Applications may be able to leverage the underlying reference information model to help can share data in a well encapsulated framework.
4.11.2 Domain Information Model

Domain models express the full information model and relationships that exist in a specific realm of knowledge in the business language of the domain itself. This might be a realm such as cancer care or infectious disease surveillance. It is domain specific and does not try to express every contact or peripheral information modeling for related but distinct domains of knowledge.

4.11.3 Bridging between the Domain and the reference model

These two models – the domain model and the reference model – are related in that the expression of the domain model in terms of the reference model provides a stable, robust construct that is suitable for use in interoperability. A bridge must be built to traverse between these two models. Building this bridge is an iterative manual process. The bridging process leads to a model that is called the “Conceptual Information Model” – this is the model from which the actual interoperability specifications are derived.

4.11.4 Logical Information Model

A Logical Information Model is an information artifact that provides a level of granularity such that the model may be directly consumed by a developer to build one or more implementation specific artifacts. Both the conceptual model and the reference model inform the logical model. All classes and attributes are defined and the terminology to be used in implementations has been identified at a level of value domains, but not yet constrained to a point that all values would be used in any specific implementation.

4.12 Templates

A template describes a pattern of use of a model fragment. It is a statement of restrictions on the attribute value domains, cardinality and optionality of the information model when it is applied to a particular use case or context. Templates often provide additional definition and documentary material that describe how the information models are applied to very specific use cases or contexts. This material needs to be consistent with the underlying model fragments to which it applies. Templates may be broken down into reusable modules.

4.13 Executable Models

In order to assist implementation, it is useful to provide executable forms of the models. In these models, the information model is represented in a form that can be interpreted by other software that can perform useful functions such as validate instances or generate code. Examples are W3C XML schema, schematron, etc.; many forms exist. These executable forms are frequently incomplete representations, limited to what the software and/or specifications are capable of doing.

4.14 Summary

Through this canonical information framework, the static information artifacts that serve to provide semantic interoperability between trading partners has been described. It is crucial to realize how each artifact provides additional context to enhance the semantics of its more primitive related artifact. It is this additional semantic layering that allows the progressive levels of interoperability that allows greater understanding of the information at each level.

The diagram below shows how each artifact wraps context around its related artifact.

At each level, a declaration of interoperability capability can be made.
Figure 19: Artifact context wrapping
5 Enterprise Consistency and Conformity Framework (ECCF)

5.1 Purpose

The Enterprise Consistency and Conformity Framework defines the language that describes the semantics of the relationships between the cells formed by the intersection of the dimensions (columns) and the perspectives (rows) of the Interoperability Specification Matrix (ISM). The concepts defined in the SAIF Canonical Definition document to ensure coherent discussions in the context of one or more SAIF Implementation Guides (SAIF IGs). Recall that the ISM is a Type. Each SAIF Implementation Guide (SAIF IG) uses the ISM to define an IG-specific Profile, the Interoperability Specification Template (IST) as a realization of the ISM. A specific collection of artifacts in a particular instance of an IST is referred to as an Interoperability Specification Instance (ISI). A more detailed discussion of the ISM, IST, and ISI and their relationships is provided in Section 6.

Figure 20 ECCF Terms of Art Concept Map. (See Figure 1 for color convention semantics)

5.2 ECCF Terms of Art

The terms consistency and conformity are both composite terms whose meaning is derived from the collective meanings of the ECCF terms of art. In addition, both terms have formal roots in both the ISO standards and ODP arenas. As shown in the concept map (above), ECCF language as defined in the SAIF-CD is instantiated in individual SAIF IGs with a focus on both Conformity Assessment and Well-formed-ness (Consistency) Assessment. Within the context of the SAIF-CD, the two core concepts are defined as follows:

Consistency: “Well-formed-ness” of artifacts both within the artifact itself, i.e. its content and representation conventions, and between artifacts, i.e. identical semantics are correctly and accurately represented across
artifact boundaries, and explicit and implicit dependencies are accurately and consistently represented.

“Steadfast adherence to the same principles, course, form, etc. Agreement, harmony, compatibility, and especially correspondence or uniformity among parts of a complex thing.” (Definitions.net, 2011).

**Conformity:** A measure of the **conformance** of a given implementation instance to a given specification AND/OR a measure of the compliance/correctness of a given specification to another specification, usually in the context of the compliant specification being deemed a valid transformation from the original specification. “Conformity assessment is the name given to processes that are used to demonstrate that a product (tangible) or a service or a management system or body meets specified requirements. (ISO)”

**Interoperability Specification Instance (ISI) Subject:** Each instance of an Interoperability Template, referred to as an Interoperability Specification Instance (ISI), contains artifacts whose scope collectively defines a particular component, for example, system, sub-system, service, document, or message. This scope is referred to as the Interoperability Specification Instance Subject.

**Conformance:** "Conformance relates an implementation to a standard. Any proposition that is true of the specification must be true in its implementation. (ISO, 2010)"

The ECCF provides a language that enables specification developers and consumers to explicitly understand and communicate about various aspects of a given component that impact its use in one or more interoperability scenarios. A key aspect is the ability to speak quantitatively about the degree to which a given implementation satisfies the static or informational and dynamic or behavioral semantics, or both, as defined in the various artifacts contained in an ISI. A given implementation instance is said to be conformant to a given specification if the implementation instance satisfies the various requirements defined in the specification.

The ECCF does not define conformance at the “global” implementation level—an implementation is either conformant or non-conformant to a given specification. Rather, conformance is defined at the more granular level of the **Conformance Statement**, a testable, Boolean-valued statement of a specific requirement (static or dynamic) of the component as explicitly specified in the component’s ISI.

A given implementation then makes **pair-wise Conformance Assertions**, claiming that it satisfies particular Conformance Statements. These claims can be validated on a one-by-one basis through either automated or human-based testing. Thus, within the context of the ECCF, the concept of Conformance has two defining characteristics:

- Conformance is only used to discuss the relationship between an implementation and a specification.
- Conformance is tested and certified at a granularity determined by Conformance Statements contained in component-specific artifacts in an ISI. Conformance Statements in a given ISI are associated pair-wise with Conformance Assertions made by the implementation claiming conformance to the ISI. This relationship is shown in the illustration that follows. Note that Conformance Statements are testable Boolean requirements collected at Conformance Points as defined in RM-ODP.

**Conformance Statements:** Paraphrasing from [ISO/IEC 10746-2 (ISO, 2010)]: "A conformance Statement is a statement that identifies testable requirements at a specified Conformance Point within a specification, explicitly defining the behavior which must be satisfied at these points. Conformance Statements will only occur in standards which are intended to constrain some feature of a real implementation, so that there exists, in principle, the possibility of testing."

The conformance of a given implementation instance to a particular specification is verified based on the truth value of a pair-wise Conformance Assertion made by an implementation instance against a given artifact-resident Conformance Statement within a given specification.

Note that the requirement that each Conformance Statement be testable and verifiable, that is, that each Conformance Statement be a Boolean statement, does not require that the statement be testable by automated means. Often Conformance Statements made from the Conceptual Perspective, and particularly those made in the Enterprise dimension, may only be verifiable as True through human examination of a given implementation instance. Thus, the critical defining feature of a valid ECCF Conformance Statement is its Boolean testability and not its particular mode of verification.
Conformance Assertions: Conformance Assertions are made by a given implementation instance and are linked pair-wise to Conformance Statements made within a given artifact as part of a component specification. The pair-wise association of specification-resident Conformance Statements with implementation-instance-resident Conformance Assertions enables creation of testing harness and user verification frameworks. This enables a given implementation instance to be “verified” or “tested” as “conformant to a given specification.” Note that the words “tested,” “verified,” and “certified” are subject to confusion and conflated definitions and usage. The ECCF therefore uses very specific definitions of terms to proactively prevent this confusion.

Conformance Testing: - Quoting from [ISO/IEC 10746-2 (ISO, 2010)]: “A Reference Point (RP) is a point in the specification which a specifier nominates to be a candidate Conformance Point, that is, a place where behavior may need to be observed to determine conformance. A specifier may define many RPs in the specification but it may be that only a subset of these can be used for testing in specific scenario. These are referred to as conformance points. Note that in the context of SAIF, the notion of an RP can be stated as “the statement(s) in a given artifact that that are referred to as an ECCF Conformance Statement”).

1. Perceptual: an RP where there is some interaction between the system and the physical world, for example, a human-computer interface.

2. Programmatic: an RP where a programmatic interface can be established to allow access to a function.

3. Interworking: an RP where there is a physical communication channel through which information exchange can be monitored.

4. Interchange - an RP where an external physical storage medium can be introduced into the system, for example, in cases where information can be recorded on one system and then physically transferred, directly or indirectly, to be used on another system.

NOTE: ODP defines four broad categories of Reference Points, the first two of which are relevant to the SAIF-CD (points 3 and 4 are only relevant in the context of a specific implementation and are therefore outside the scope of the SAIF-CD and are included simply for completeness with respect the ODP reference).

From the preceding discussion of Conformance Statements and Conformance Assertions, it should be clear that Conformance Testing, that is, the process whereby a given implementation instance is evaluated to determine which of its various Conformance Assertions are valid implementations of a given specification’s Conformance Statements:

- Is a granular construct, i.e. it is determined at the level of individual Conformance Assertions made by the implementation instance and not a global characteristic of a given implementation instance (unless, of course, the specification contains only a single global Conformance Statement against which the implementation instance can claim conformance); and

- Exists in a one-to-many relationship between specifications and implementations, i.e. there is a one-to-many relationship between a given specification instance and the collection of implementation instances that can claim conformance to the specification.

Compliance: Quoting from [ISO/IEC 10746-2 (ISO, 2010)]: “Requirements for the necessary consistency of one member of the family of specifications or standards with another are established during the standardization process. Adherence to these requirements is called compliance.”

In the context of SAIF, Compliance refers to logical consistency and correspondence between a source artifact and a target artifact, with the target having undergone a transformation (usually a restriction). That is, given an existing source artifact such as a specification or standard, and a target artifact that resulted from applying a known transformation to the source, the target is in Compliance with the source if the transformation is considered “legal” by the source artifact’s originator.
Compliance can be established between artifacts in a single ISI cell or, alternatively, across multiple ISI cells. When a Compliance relationship crosses cell boundaries, it can do so either horizontally or vertically. Diagonal Compliance is also possible, although less common than vertical or horizontal Compliance relationships. Thus, localization is considered a form of Compliance.

Unlike Conformance, Compliance is seldom overtly tested since non-compliant transformations producing non-compliant artifacts usually cause other issues which can be discovered in either Correspondence monitoring or Conformance testing.

Certification (Conformance Certification): the outcome of successful conformance testing, i.e. the results of that testing. Certification should not be confused with the testing that results (potentially) from the test/evaluation. Certification of Conformance (or lack thereof) is based on the ability of a given implementation instance to satisfy one or more of the Conformance Assertions made by the implementation instance against the pair-wise Conformance Statement in the specification.

Correspondence and Consistency: Quoting from [ISO/IEC 10746-2]: "Viewpoint correspondence is a statement that some terms or other linguistic constructs in a specification from one ODP viewpoint are associated with (e.g. describe the same entities as), terms or constructs in a specification from a second ODP viewpoint. The forms of association that can be expressed will depend on the specification technique used."

In the SAIF ECCF, Correspondence is used synonymously with the term consistency, the latter term having been chosen over the former as the nom de plume of the ECCF because of the more commonly shared understanding of the term as opposed to the term “correspondence.” Both terms are focused on the notion of logical coherence of a given ISI that is “unified” in its expression of a given component’s various Dimensions and Perspectives. Thus, a consistent, well-formed specification – demonstrates a high degree of correspondence between its various components. This is a somewhat hard-to-define but relatively easy (to the trained eye) to perceive “expressive traceability.”

In summary, the notion of Correspondence underscores the fact that the Dimensions of an IST are not orthogonal, but rather express different aspects of a single component, system, sub-system, and specification.

Traceability: In everyday parlance, traceability refers to the ability to link an instance with a concept, for example, a requirement, with an implementation-resident functionality. In the context of SAIF, traceability has a somewhat more formal meaning. Traceability defines the relationship that links an attribute or other feature of a particular artifact defined in a particular dimension and at a particular perspective. This includes but is not limited to semantics or Conformance Statements. Note that traceability is a vertical relationship spanning all Perspectives and including any implementation instances associated with a given specification. Traceability includes both Conformance and Compliance relationships.

Provenance: The documented “reverse traceability” of an existing artifact from its current state to its origination, including whatever attribution, context or both, is associated with the various lifecycle changes of the artifact. Provenance is, among other things, the source for documenting the various constraints and localizations that a given item undergoes as it moves from, for example, a Conceptual to a Logical to an Implementable specified artifact.

Localization: A specialization of compliance whereby some aspect of an artifact’s semantics, informational (static) or behavioral (dynamic), or other defining attribute is restricted compared to its original occurrence. Localization commonly occurs as a concept passes from one or more of the following: the Conceptual Perspective to the Logical Perspective, the Logical Perspective to the Implementable Perspective, and the Implementable Perspective to an implementation instance.

Compatibility: Given a specification, two implementation instances are said to be Compatible if-and-only-if they can successfully engage – without further modification of their implementation specifics – in any shared purpose scenario that can be expected to be supported based on the reference specification that is implemented by the involved instances. In other words, two implementation instances are said to be Compatible if they do not “localize” by specifying context-specific, non-interoperable constraints.
6 Interoperability Specification Matrix (ISM)

The Interoperability Specification Matrix (ISM) defines a 5-column-by-3-row matrix ("table") which distributes the multiple aspects of a given component’s specification across the various cells of the matrix. The structure of the ISM is based on proven cognitive models for describing complex systems which revolve around the notion of partitioning complexity based on a number of Dimensions while simultaneously viewing each of these dimensions from multiple Perspectives.
6.1 ISM Artifacts Types and Conformance Statement Types

As shown in the preceding concept map, the ISM defines prototypic artifacts types, the specific content and representation of which are defined in a particular SAIF-CD-compliant SAIF IG. In addition, although the SAIF-CD does not define specific artifacts, it does require that specific artifact instances contain testable – i.e. Boolean – Conformant Statements. Thus, in parallel to the SAIF-CD definition of artifact types, the SAIF-CD defines Conformance Statement types. These types are, in turn, defined in SAIF IG profiles. Finally, a given artifact in an ISI can contain multiple Conformance Statement instances against which a given implementation of a component specification can make pair-wise Conformance Assertions. (See Appendix for examples of artifacts and associated Conformance Statements.)

NOTE: In the context of a specific SAIF IG, the ISM defines a <<type>> construct which is then explicitly made manifest in a SAIF IG-specific <<profile>> that specifies the content and representation of all artifacts that collectively comprise a given component’s specification. The process of defining an ISM-conformant matrix for a given IG – a construct referred to as an Interoperability Specification Template (IST) – involves the use of restrictions and specializations of the concepts and constructs used to define the ISM. A collection of specification artifacts for a given component is then an <<instance>> of the profile and is referred to as an Interoperability Specification Instance (ISI). Finally, given a particular specification instance, one or more implementations of that specification can be developed and deployed and, in the process, subject to conformance certification testing to determine the degree of fidelity that the implementation has relative to the specification. (See Figure 2 and Section 7.3 for details and a more complete discussion.)

6.2 Dimensions

The names of the Dimensions in the SAIF ISM are identical to the Viewpoint names in RM-ODP. However, the semantics are not identical. In particular, the SAIF-CD Dimensions are restrictions and/or specializations of the various RM-ODP Viewpoint languages. The SAIF-CD-specific definitions are as follows:
6.2.1 Enterprise Dimension

The Enterprise Dimension focuses on defining salient aspects of the “organizational context.” In the context of interoperability, this means “the intra- or inter-organizational deployment and interoperability context” for which the specification-specific artifacts are being defined.

For each of the three perspectives, the Enterprise Dimension should aspects of the interoperability context that emerge from an understanding of business objectives and business rules. This includes relevant pre- and post-conditions for interoperability scenarios.

Due to the basic nature of the Enterprise dimension, most information at the Logical and Implementable Perspectives originates in the Conceptual Perspective. Very little “new” information is added at the Logical and Implementable Perspectives in the Enterprise Dimension.

6.2.2 Information Dimension

The Information Dimension focuses on defining the informational or static semantics that are relevant with respect to interoperability interactions.

These semantics are expressed using Information Framework (IF) grammar and include constructs such as information and data models, data types, and value sets, discussed in the IF chapter of this document. However, as discussed in the IF chapter, the scope of the Information Framework is not limited to use in Information Dimension specifications.

6.2.3 Behavioral (Computational) Dimension

The Behavioral (Computational) Dimension focuses on defining the behavioral or dynamic semantics that are relevant with respect to interoperability interactions. These semantics are expressed using Behavioral Framework grammar and include constructs such as contract and interface specifications and accountability profiles, discussed in the BF chapter of this document. The BF makes extensive use of the RM-ODP Enterprise Language, a set of well-defined concepts and constructs that are defined as part of the RM-ODP Enterprise Viewpoint. Therefore the scope of the Behavioral Framework is not limited to use in Behavioral Dimension specifications.

6.2.4 Engineering Dimension

The Engineering Dimension focuses on defining the deployment topologies that are relevant with respect to interoperability interactions. The RM-ODP (ISO RM-ODP) contains considerable detail about the construct “transparencies.” Discussion of transparencies is beyond the scope of the SAIF-CD. However, certain SAIF-IGs could benefit substantially from including certain transparency constructs in their organization-specific IGs. Specifically, salient details of different implementable meta-models (for example, specifications supporting interoperability scenarios based on messages, documents, or services) can be explicitly captured across the three perspectives of the Engineering Dimension.

6.2.5 Technology Dimension

The Technology Dimension focuses on defining various implementable standards for hardware or software as relevant, which will ultimately support the specification. This definition is referred to as the “technology semantics” of a component as used in interoperability scenarios.

Artifacts defined under the Technology Dimension often make reference to artifacts in other ISM cells in order to appropriately contextualize the referenced artifacts. Further discussion of the Technology Dimension is appropriate for SAIF-IGs and includes topics such as technology-specific deployment or configuration guides, technology selection criteria, and maintenance and migration plans. Conformance Statements are not defined under the Technology Dimension as often as they are under the other dimensions. Refer to the discussion of conformance in the ECCF chapter.
6.3 Perspectives

The perspectives correspond to standard role-based terminology of contemporary software engineering processes.

The names of the perspectives or rows of the ISM reflect views of specification artifacts associated with software engineering roles, that is, Domain Expert, Analyst, Architect, Developer, and others as discussed below. The HL7 ArB chose to use three perspectives rather than more finely granulated alternatives, for example, the six Perspectives of Zachman2.

It is possible to associate each specified artifact with a row in a RACI (Responsibility, Accountability, Consulted, and Informed) matrix. This can explicitly link the artifact to the appropriate organizational roles for a SAIF IG.

NOTE: The SAIF-CD definitions of the three SAIF Perspectives and their associated software-engineering role are given in the following discussion. It is important to note that the SAIF-CD Perspectives are not formally linked with the Object Management Group’s levels-of-abstraction in Model-Driven Architecture (MDA). That is, the SAIF Conceptual Perspective is not semantically equivalent to the MDA concept of Computationally Independent Model (CIM), the Logical Perspective is not equivalent to the MDA Platform Independent Model (PIM), nor is the Implementable Perspective equivalent to the MDA Platform Specific Model although this Perspective is the SAIF Perspective that most closely aligns with an MDA analogue.

6.3.1 Conceptual Perspective

The artifacts of the Conceptual Perspective are of interest to and readable by Domain Experts (DEs) or Subject Matter Experts (SMEs). These artifacts are most commonly focused on the “Problem-Space” rather than the “Solution Space,” and contain, distributed across the five columns of an ISM, explicit, unambiguous descriptions of the various dimensions of the component or system that being specified.

Artifacts of the Conceptual Perspective are normally developed by “outward-facing analysts” who have reasonable domain knowledge and can facilitate dialogues with DEs and SMEs. These analysts also take the results of such dialogues and represent the content in structured artifacts which remain understandable to DEs or SMEs. These sometimes formally structured artifacts may include clearly-stated business rules, concept maps, and simple UML class or activity diagrams.

A fully-specified Conceptual Perspective thus should be both readable and traceable by DEs and SMEs and rigorous enough to serve as input into the development in the Logical Perspective.

6.3.2 Logical Perspective

Artifacts in the Logical Perspective represent traceable translations of Conceptual-level artifacts into a form and format, usable by and useful to architects and “inward-facing analysts.” Also included are additional specification materials required by architects preparing artifacts for consumption by developers.

Note that there is no firm or fixed line that definitively and unambiguously determines where the Conceptual Perspective ends and the Logical Perspective begins. The same is true of the lack of definitive boundaries between the Logical and Implementable Perspectives.

For a given SAIF-IG, the most important aspects of defining artifacts in a given perspective are the combination of role-based awareness based on artifact creation and consumption, and consistent placement of artifacts across multiple specifications.

6.3.3 Implementable Perspective

Artifacts in the Implementable Perspective are typically defined by developers, often through dialogues with designers, architects, or both. Note that the artifacts in the Implementable Perspective are not actual implementations, but rather implementable in a number of implementation instances. Thus all the necessary technical bindings, including data types, value sets, class libraries, and interface specifications, can be found
distributed across the ISM dimensions at the Implementable Perspective. These artifacts will enable one or more instances of the specification to be realized by one or more development teams.
7 Compliant SAIF Implementation Guides

In order to quantitatively evaluate whether or not a given SAIF Implementation Guide (SAIF IG) is a well-formed instance of the SAIF Canonical Definition (SAIF CD), the SAIF CD provides a set of Compliance Criteria against which each SAIF IG may be evaluated. This chapter contains a table containing those Compliance Criteria sorted by their relative focus with respect to the content of the SAIF CD. Note that the compliance of a SAIF IG is evaluated on a SAIF CD Compliance-Statement-by-Compliance-Statement basis, i.e. a given SAIF IG may be compliant to one statement but not compliant with another. **However, in order for a particular SAIF IG to be evaluated as fulfilling the minimal criteria necessary to claim SAIF CD compliance, it must be compliant with each and every SAIF CD Compliance Statement of the SHALL “type.”**

In particular, there are four “types” of SAIF CD Compliance Statements as defined in the following table. It should be noted that the choice of this particular representation and semantics for SAIF CD Compliance Statements is aligned with the larger HL7 V3 Publishing Facilitators’ Guide, Section 5.1.5, which states: “HL7 adheres to ISO/IEC Directive, Appendix G, as delineated in the following table:

<table>
<thead>
<tr>
<th>Stringent Use of SHALL, SHOULD, MAY and Other Modal Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>To Convey the Sense of:</td>
</tr>
<tr>
<td>Required/Mandatory to achieve SAIF CD compliance or Absolute Prohibited, i.e. a SAIF IG that implements a SHALL NOT statement can NOT be judged as SAIF CD compliant.</td>
</tr>
<tr>
<td>Best Practice/Recommendation or Suggested as having potentially negative outcomes based on experiences of authors of the SAIF CD</td>
</tr>
<tr>
<td>Acceptable/Permitted based on requirements, scope or other considerations that are known to vary on an organization-by-organization basis.</td>
</tr>
</tbody>
</table>

Following are a few additional notes of clarification on the SAIF CD Compliance Statements:

i. The term *compliance* – rather than *conformance* – is used as the overarching framework for defining the relationship between the SAIF CD and its various, organization-specific SAIF IG as a manifestation of the notion that a given SAIF IG is *not* an implementation whose *conformance* should be assessed, but rather a *specification* derived from the SAIF CD. In particular, referring to the definition of the term *compliance* in the Enterprise Consistency and Conformity Framework chapter of the SAIF CD as “the correctness with which a target specification is derived from a source specification,” one can readily see that compliance is the correct term to describe a well-formed relationship between the SAIF CD and a given SAIF IG.

ii. As indicated in the table and discussion above, the terms “SHALL” and “SHALL NOT” should be considered to be *minimum* SAIF IG requirements (and equally critical pertinent negatives), i.e. a compliant, well-formed SAIF IG must fulfil each and every one of the Compliance Statements of the SHALL type, and cannot violate any of the SHALL NOT Compliance Statements.

iii. In the context of the development of a SAIF IG, the terms “SHOULD” and “SHOULD NOT” should be interpreted as meaning “Considered Best Practice” features of maximally functional SAIF IGs and therefore used as goals for a SAIF IG that is being developed over multiple iterations within a given
organization. The HL7 Architecture Board recognizes that the effort involved for an organization to define, develop, deploy, and manage an organization functioning under a SAIF IG can be considerable and, as such, is very often likely to be approached in an iterative, incremental fashion. As such, the notion of Best Practice should be viewed as end-point goals that may be approached over several iterative releases of a given SAIF IG once it has fulfilled the minimal requirements stated in the SHALL Compliance Criteria.

iv. In contrast, SHOULD NOT statements should be considered aspects which either goes directly against known Best Practices or development characteristics that have been demonstrated to provide less-than-optimal results within an organization developing and deploying a compliant SAIF IG.

v. Statements of type MAY should be considered as options for an organization which may or may not be relevant based on the organization’s particular SAIF IG scope, governance context, maturity, or other organization-specific characteristics.

7.1 Evolution of the SAIF CD

The HL7 Architecture Board (ArB) recognizes that the process of developing multiple organization-specific SAIF IGs may very well uncover errors of commission – or, more likely, omission – in the SAIF CD. In addition, the ArB recognizes that the SAIF CD may evolve, particularly in the first months of organizational implementations. This is the primary reason that the SAIF CD is being published as a Draft Standard for Trial Use (DSTU), i.e. to encourage its formal adoption and adaption in multiple organizations while simultaneously guaranteeing that a future version of the SAIF CD may be developed based on implementation experience and may, if necessary, include “discontinuous” changes compared to the current SAIF CD.

The ArB is committed to maintaining an open, responsive, and timely dialogue with all users of the SAIF CD. As, such, the ArB encourages all organizations who either identify omissions in the SAIF CD, or alternatively, who encounter difficulties in using the SAIF CD as the basis around which an organization-specific SAIF IG is being developed relative to their organization’s specific requirements, to contact the ArB so that questions and concerns may be directly addressed in a timely manner, and, if necessary, edits may be made in the SAIF CD so that it can iteratively and incrementally evolve.
### 7.2 SAIF CD Compliance Statements: Criteria for a Compliance SAIF Implementation Guide

<table>
<thead>
<tr>
<th>A compliant SAIF Implementation Guide... (IG)...</th>
<th>Chapter Reference</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>...SHALL</strong> define – or otherwise reference – a set of modeling languages/grammars that will be used as the basis for defining the artifact templates that populate the IG’s Interoperability Specification Template (IST).</td>
<td>1. CD</td>
<td>Component specifications and their derived Implementation instances are defined and developed based on the content and representation of artifact templates defined in compliance with the Behavioral and Information Framework language defined in the SAIF CD. Thus, a SAIF IG explicitly defines the set of modeling languages/grammars that will be used to define artifact templates that will, in turn, be instantiated as artifact instances which collectively define a given component from an interoperability perspective. These artifact instances are collected in the IG’s IST.</td>
</tr>
<tr>
<td><strong>...SHALL</strong> instantiate the SAIF CD language concepts as concrete modeling languages/grammars that are then used to define specific artifact templates. (See the MAY Compliance Statements for requirements concerning the scope of IG languages/grammars).</td>
<td>2. CD</td>
<td>The concepts and relationships in the SAIF CD cannot be changed in a compliant SAIF CD. However, they can be aliased or extended – or only utilized in part – as long as changes do not conflict with the definitions supplied by the SAIF CD. The rationale for this restriction is the desire to have a single set of conceptual languages – i.e. the languages defined by the SAIF CD – that then serve as the basis for all IG-specific modeling languages/grammars, thereby facilitating cross-enterprise/cross-IG interoperability scenarios focused on achieving Shared Purpose.</td>
</tr>
<tr>
<td><strong>...SHALL</strong> be defined in terms of modeling languages/grammars that are semantically consistent with the languages specified in the SAIF CD, in particular those languages defined in the Information and Behavior Frameworks (IF and BF). (See the MAY Compliance Statements for requirements concerning the scope of IG grammars).</td>
<td>3. CD</td>
<td>The purpose of the SAIF CD is to provide a single specification for the reference languages that can then subsequently be adopted or adapted as required to meet the requirements of a particular organization as expressed in that organization’s SAIF IG. NOTE: This Compliance Statement is meant to clarify/generalize the statement 2. CD which is focused solely on artifact template definitions whereas this statement covers the entire IG’s definition.</td>
</tr>
</tbody>
</table>
...**SHALL** clearly and explicitly define the scope of coverage of the IG including the expected use of the grammars defined in the IG.

<table>
<thead>
<tr>
<th>4. CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>A SAIF IG has a particular scope defined to meet the interoperability requirements for a given organization. This scope should be clearly delineated in the organization’s SAIF IG. The scope definition ultimately specifies the scope of allowable Conformity Certifications that can legitimately fall under the domain of the IG.</td>
</tr>
</tbody>
</table>

...**SHALL NOT** introduce a new foundational abstract concept not defined in the SAIF CD that conflicts with a concept of relationship defined in the SAIF CD, i.e. extensions of a SAIF CD concept **SHALL NOT** change the fundamental semantics of the original SAIF CD concepts or relationships.

<table>
<thead>
<tr>
<th>5. CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>The SAIF CD is expected to be the primary source of the definitions of the semantics of the concepts and constructs used to define a SAIF IG. Cross-organizational interoperability based on Shared Purpose operationalized at the architectural instance level will be facilitated if all SAIF IGs are developed using a single canonical definition. (NOTE: See MAY statement below RE the legitimate inclusion of concepts or relationships in an IG if they do not conflict with those in the SAIF CD.)</td>
</tr>
</tbody>
</table>

...**SHALL NOT** use the option of choosing only a subset of SAIF CD concepts and relationships as a mechanism by which SAIF CD content can be replaced by non-compliant content which takes the place of the missing SAIF CD content not included in the selected subset.

<table>
<thead>
<tr>
<th>6. CD</th>
</tr>
</thead>
<tbody>
<tr>
<td>As noted in 5.CD, an IG may make legitimate extensions from SAIF CD concepts and relationships as required to fulfill an organization’s specific interoperability and/or enterprise architecture requirements. However, a complaint IG cannot subset the SAIF CD’s concepts or relationships and then replace them within the context of a given IG with concepts or relationships that are, in fact, inconsistent with SAIF CD concepts or relationships as such a substitution undercuts the core value proposition of the SAIF CD of supplying a common language base to enable cross-enterprise interoperability even in those cases where the participating organizations have differences in their respective IGs.</td>
</tr>
</tbody>
</table>
### Governance Framework

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>7. CD</strong></td>
<td>IG since definition, design, development, and deployment such an IG represents a considerable effort that is most often best achieved over an extended time period which includes an ongoing process of utilization and pragmatic feedback. “SHOULD” statements are Best Practices that can be prioritized to meet the particular needs of each organization as they develop their respective SAIF IGs.</td>
</tr>
<tr>
<td><strong>8. CD</strong></td>
<td>A certain amount of flexibility and malleability is expected from the SAIF CD’s root definitions of concepts and constructs. However, note the preceding SHALL NOT statement which states that although SAIF CD concepts and constructs may be extended, their fundamental, underlying semantics may not be altered so as to be incompatible with those defined in the SAIF CD specification document.</td>
</tr>
<tr>
<td><strong>9. CD</strong></td>
<td>See comments 5, 6, 8.CD</td>
</tr>
<tr>
<td><strong>10. CD</strong></td>
<td>A given SAIF IG is defined based on the requirements of an organization with respect to its expected involvement in various interoperability shared purpose scenarios. Considerations determining the scope and content of an IG’s modeling languages/grammars include – but not are limited to – Deployment Context, Interoperability Type, and Integration Focus of the various specifications and components developed under the IG’s definitions. See also comments 5, 6, 8.CD</td>
</tr>
<tr>
<td>Number</td>
<td>Type</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1. GF</td>
<td></td>
</tr>
<tr>
<td>2. GF</td>
<td></td>
</tr>
<tr>
<td>3. GF</td>
<td></td>
</tr>
<tr>
<td>4. GF</td>
<td></td>
</tr>
<tr>
<td>5. GF</td>
<td></td>
</tr>
<tr>
<td>6. GF</td>
<td></td>
</tr>
<tr>
<td>Precept</td>
<td>GF</td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
</tr>
<tr>
<td>...SHALL ensure that each Precept is defined in such a manner to ensure its well-formedness including a clear definition of the <strong>Standards</strong> which the Precept must support (e.g. organizational, legal, industry, etc. standards).</td>
<td>7. GF</td>
</tr>
<tr>
<td>...SHALL ensure that each Precept is defined in such a manner to ensure its well-formedness including a clear definition of the <strong>Guidelines</strong> are recommend in support of the Precept (e.g. organizational, legal, industry, etc. patterns).</td>
<td>8. GF</td>
</tr>
</tbody>
</table>
| ...SHALL ensure that each Governance Point is defined in such a manner to ensure its well-formedness including a definition of the **People** (People-in-a-Role) that are either affected by or responsible for authoring and executing the associated Precept. | 9. GF | People act. To ensure consistency over time and across processes, explicit responsibilities may be grouped into roles. All people with the same role have the same responsibilities. People can only act in concert when they know what the rules are and how their actions affect others. People cannot be held accountable for actions they were not aware were expected to be influenced by Precepts. A RACI chart is often used to quantitatively clarify discussions regarding People, their Roles, and their respective Process connections.  

**NOTE:** A Governance Point is an identified point in an organization’s component development (or other governed) process(es) at which one or more Precepts, People, Policies, and Metrics will be defined so as to govern the inputs and outputs of that particular point in the process. |
<p>| ...SHALL ensure that each Governance Point is defined in such a manner to ensure its well-formedness including a definition of the Processes that fall under the scope of the associated Precept either in the form of evaluation of the compliance to the Precept or in the generation of the artifacts required by the Precept. | 10. GF | Processes organize actions. Well-defined processes allow people to act in concert and appropriately apply Precepts. When Precepts change, associated processes may change and when processes are changed, Precepts need to be evaluated for change as well. Otherwise, Precepts may not achieve the objectives for which they were defined. |
| ...SHALL ensure that each Governance Point is defined in such a manner to ensure its well-formedness including a definition of the Metrics that will be evaluated to determine compliance (or lack thereof) with the Precept. |
| NOTE: Metrics must include consequences of failure to meet specified threshold, content, etc. |
| 11. GF | Explicit Metrics allow the evaluation of the precept as well as assurance that the precept is being applied appropriately. Precepts without any consequence if they are not applied serve as instruction at best. They cannot result in consistent specifications that support interoperability. |
| ...SHALL NOT define new Precepts which conflict or contradict existing Precepts. |
| 12. GF | Incoherent actions result from incoherent precepts. Adding new precepts or changing existing precepts needs to be done with a clear understanding of the scope of impact. This is one of the major responsibilities of the person(s) who oversee governance in a particular organization. |
| ...SHALL NOT define Precepts that introduce more risk than they mitigate. |
| 13. GF | One-size-fits-all Precepts may prove to be a burden that outweighs the risk they are expected to mitigate. Clear conditions of application permit the trade-off of managing risk to consider the scope of application, schedule and cost. |
| ...SHOULD include a comprehensive Risk Assessment to define those areas of the implementing organization’s component development strategy that present the highest risks to project failure and that therefore should be the initial targets of defined governance points. |
| 14. GF | Risk Assessment allows the cost/benefit tradeoffs that are inherent in component development to be made explicit. Change management of precepts should be accompanied by a review of the related risk profiles to ensure the full set remain coherent and consistent with the degree of acceptable risk. |
| ...SHOULD include a comprehensive Stakeholder Inventory to identify all stakeholders affected by the Governance Framework specified for implementation in the IG. |
| 15. GF | Without knowing who is impacted by precepts, it is difficult to successful engage with them when introducing precepts. Communication strategies may need to vary depending on the different degree of involvement of people in defining, executing and being impacted by precepts. |
| ...SHOULD identify the Source of Authority from which the Precept and its associated Objectives, Policies, Standards, and Guidelines originate (e.g. organizational policy, legal requirements, etc.) |
| 16. GF | Without knowing the Sources of Authority for Precepts, it is difficult to maintain alignment as organizational structures evolve. Precepts without clear lines of authority are at risk of becoming not only obsolete, but may add unacceptable risk. |
| 17. GF | Governance Points are identified steps in known processes and can therefore be named in “organization-friendly” terms. Ambiguously named and defined Governance Points lead to incoherent, incomplete, or inconsistent actions. |
| 18. GF | The SAIF GF is intended to support interoperability and consistent behavior of people and processes. Not all people are capable of or permitted to play all roles. Roles which require specific skills, authority or permissions should make those prerequisite known. In addition, Precepts focused on governing aspects of technical processes should specify the required inputs that are minimally expected to allow successful completion of the process. |
| 19. GF | Roles may not be applicable all of the time. Any additional conditions that must be true before a role is relevant should be clearly stated. In addition, conditions that apply when a specific person leaves a role must also be clearly stated. Likewise, Precepts focused on governing aspects of technical processes should specify the required outputs of the process, often in the form of the Metrics associated with a given Precept. |
| 20. GF | Accepting a role needs to be an explicit step in a recognized process so that authorities, responsibilities and permissions have an explicit start date/time to be appropriately accounted for. |
| 21. GF | Relinquishing a role needs to be an explicit step in a recognized process so that authorities, responsibilities and permissions have an explicit end date/time to be appropriately accounted for. |</p>
<table>
<thead>
<tr>
<th>Scenario</th>
<th>Precept</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>...SHOULD</strong> explicitly define the <strong>Permissions</strong> associated with each Person-in-a-Role associated with a given Precept.</td>
<td>22. <strong>GF</strong></td>
</tr>
<tr>
<td>Role names may serve as mnemonics for ready recognition, but an inventory of expected actions may be necessary to unambiguously define the scope of a role. This is especially true when similarly named roles are defined by different organizations. The extra effort to explicitly define named roles in terms of their actions will pay off in terms of managing risks across boundaries and over time.</td>
<td></td>
</tr>
<tr>
<td><strong>...SHOULD</strong> explicitly define the <strong>Prohibitions</strong> associated with each Person-in-a-Role associated with a given Precept.</td>
<td>23. <strong>GF</strong></td>
</tr>
<tr>
<td>See comments 22.GF</td>
<td></td>
</tr>
<tr>
<td><strong>...SHOULD</strong> explicitly define the <strong>Accountabilities</strong> associated with each Person-in-a-Role associated with a given Precept.</td>
<td>24. <strong>GF</strong></td>
</tr>
<tr>
<td>See comments 22.GF</td>
<td></td>
</tr>
<tr>
<td><strong>...SHOULD</strong> define the allowable <strong>Delegations</strong> for each defined Person-in-a-Role, i.e. whether the role can delegate authority to another person or role which is then allowed to be held responsible for accomplishing the original role’s assigned tasks, required deliverables, etc.</td>
<td>25. <strong>GF</strong></td>
</tr>
<tr>
<td>Some roles may become a bottleneck in a process unless more people can carry out the expected actions. Distribution of governance processes may require delegation of authority. Effective accountability requires explicitly identified delegations, since delegating does not absolve the delegator of all responsibility for the delegated actions.</td>
<td></td>
</tr>
<tr>
<td><strong>...MAY</strong> add additional governance concepts or constructs as needed to operationalize a given organization’s technical governance as long as the additional concepts and constructs do not alter or replace the semantics of the concepts and constructs defined in the SAIF CD.</td>
<td>26. <strong>GF</strong></td>
</tr>
<tr>
<td>Aligning existing processes and precepts to become compliant with the SAIG CD may be eased if familiar terms are explicitly mapped to SAIF CD Governance terms. More refined concepts that apply only in some conditions are often useful in a local context, but should always be related to the parent concept to support communication across boundaries.</td>
<td></td>
</tr>
</tbody>
</table>

---

**Behavioral Framework**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Precept</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>...SHALL</strong> define one or more modeling languages/ grammars to be utilized in expressing – at the Conceptual, Logical, and Implementable levels – the <strong>behavioral semantics</strong> of components involved in Shared Purpose scenarios.</td>
<td>1. <strong>BF</strong></td>
</tr>
<tr>
<td>The SAIF CD does not fully specify operational modeling languages/grammars, instead only supply the foundation semantics of what is needed by an operationalizable IG. It is therefore the task of SAIF IG developers to concretely instantiate the concepts and relationships defined in the SAIF CD as modeling languages/grammars in a given SAIF IG.</td>
<td></td>
</tr>
</tbody>
</table>
2. BF
The IG grammars may not have concepts that are overlapping, but different than, the concepts in the CD BF. They may have the same concepts with equal or aliased terminology, or they may have additional concepts that do not conflict or overlap with SAIF CD behavioral semantics.

3. BF
The following five concepts are fundamental to any description of semantics of behavior. Note that non-service oriented paradigms must still have a corresponding concept to that of service, as defined in BF, even if it is aliased.

4. BF
See comment 3.BF

5. BF
See comment 3.BF

6. BF
See comment 3.BF
| **SHALL** ensure that its modeling languages/grammars explicitly utilize the semantics of the following SAIF CD BF-defined concepts (i.e. terms may be aliased by semantics must be used as defined). Specifically, an IG must utilize – and may not change – the semantics of the term **Interaction** as defined in the SAIF CD. | 7. BF | See comment 3.BF |
| **SHOULD** ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept **Contract.** | 8. BF | Contract is a very helpful construct that puts together multiple roles and their services, highly recommended but not absolutely necessary for an enterprise |
| **SHOULD** ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept **Interface.** | 9. BF | Many services might expose multiple interfaces, but this level of complexity might not be necessary for certain enterprises. However, the concept of interface as the externally-available description of a component’s information content and behavior remains a valuable Best Practice concept for defining cross-boundary shared purpose scenarios even if it is only utilized at the Conceptual and/or Logical level. |
| **SHOULD** ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept **Process.** | 10. BF | Process – Required in order to define sequencing and orchestration of multiple operations. However, simple services and operations can be described by an enterprise without defining processes |
| **SHOULD** ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept **Pre-Condition.** | 11. BF | Component Pre- and Post-conditions, as well as Exceptions and Signatures provide critical detail that helps formally describe how components may interact in various cross-boundary, shared purpose interoperability scenarios and are valuable regardless of implementation technologies or architectural paradigms. |
| **SHOULD** ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept **Post-Conditions.** | 12. BF | See comment 11.BF |
| **SHOULD** ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept **Exception-Condition.** | 13. BF | See comment 11.BF |
### Information Framework

<table>
<thead>
<tr>
<th><strong>SHOULD</strong> ensure that its modeling languages/grammars explicitly utilize the SAIF CD BF-defined concept <strong>Signature.</strong></th>
<th><strong>14. BF</strong></th>
<th>See comment 11.BF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SHOULD</strong> include or define a graphical or other suitable notation for quantitatively defining workflow.</td>
<td><strong>15. BF</strong></td>
<td>This is necessary in order to fully realize the value of processes</td>
</tr>
<tr>
<td><strong>SHOULD</strong> ensure that its modeling languages/grammars for describing behavioral semantics explicitly define how static (informational) semantics as defined in the IG’s Information Framework (IF) chapter are bound to/interact with the behavioral semantics defined in the IG’s BF chapter.</td>
<td><strong>16. BF</strong></td>
<td>This is the necessary glue that brings together the static and dynamic content. IG grammars must explicitly state how the information is to be bound to the interactions</td>
</tr>
</tbody>
</table>
| **MAY** include the following BF concepts in its modeling languages:  
- Community  
- Policy  
- Permission  
- Prohibition  
- Obligation  
- Flow Element  
- Event  
- Activity  
- Sequence Flow  
- Gateway | **17. BF** | These concepts are helpful for specific situations, but not all enterprises need this level of detail |

---

<table>
<thead>
<tr>
<th><strong>SHALL</strong> define one or more modeling languages/grammars to be utilized in expressing – at the Conceptual, Logical, and Implementable levels – the <strong>informational semantics</strong> of components involved in Shared Purpose scenarios.</th>
<th><strong>1. IF</strong></th>
<th>The SAIF CD does not fully specify operational modeling languages/grammars; instead only supply the foundation semantics of what is needed by an operationalizable IG. It is therefore the task of SAIF IG developers to concretely instantiate the concepts and relationships defined in the SAIF CD as modeling languages/grammars in a given SAIF IG.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SHALL</strong> ensure that the modeling languages/grammars defined to represent <strong>informational semantics</strong> do not contain concepts or relationships that conflict with the concepts or relationships defined in the Information Framework chapter of the SAIF CD.</td>
<td><strong>2. IF</strong></td>
<td>The IG grammars may not have concepts that are overlapping, but different than, the concepts in the CD BF. They may have the same concepts with equal or aliased terminology, or they may have additional concepts that do not conflict or overlap with SAIF CD informational semantics</td>
</tr>
</tbody>
</table>
3. **IF** The IG should be understandable from all reader perspectives. This enables review by domain experts to ensure that all business requirements are taken into account by the IG and by those technical architects that must translate these business concepts into an implementable perspective.

4. **IF** Any healthcare information model must be able to fully express the information requirements that allow interoperability across a wide range of healthcare applications. The ISO healthcare data types act as the reference point for the carriage of the data elements important in healthcare and can be constrained by flavors to meet local requirements while preserving the needed attributes to satisfy a broader range of stakeholders.

   **NOTE:** This requirement may be changed to an equivalent domain-specific set of complex data types when the SAIF CD is used to develop SAIF IG’s in domains other than healthcare. The important aspect of this requirement is the essential commitment to a single common data type semantics “on the wire” for successful interoperability in a shared purpose scenario.

5. **IF** All terminology bindings to coded attributes need to be resolvable by all trading partners and to do this, code systems must be unambiguously defined.

6. **IF** The only means for implementing longitudinal analysis of a dataset in a statistically valid way where coded elements are used is to be able to resolve a coded member of a valueSet to a specific valueSet definition.

7. **IF** Any change in the definition of concepts in any of the chapters will lead to confusion in adoption of other concepts both in the IF chapter and other chapters that depend on the IF.
<table>
<thead>
<tr>
<th><strong>...SHOULD</strong> have the logical model defined in the IG derived and traceable to a formal reference information model.</th>
<th><strong>8. IF</strong></th>
<th>To allow reproducible models that aid in the achievements of interoperability, all logical models should derive from a top level reference model where one exists in the domain.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enterprise Consistency and Conformity Framework</strong></td>
<td><strong>1. ECCF</strong></td>
<td>The ECCF defines the concepts that allow various relationships between instances of artifact templates managed including the correctness and completeness of a given artifact instances content and representation. In order for artifact instances to be effectively and efficiently governed, each artifact template must be clearly defined to ensure that both conformance (implementation) and compliance (other artifact template instances) to the source artifact instance can be quantitatively assessed.</td>
</tr>
<tr>
<td><strong>...SHALL</strong> ensure that the definition of each artifact template referenced by the IG’s IST includes explicit <strong>Conformity Statements</strong> – as defined in the SAIF CD ECCF chapter – by which artifact template instances can be evaluated to determine the correctness and completeness of the instance.</td>
<td><strong>2. ECCF</strong></td>
<td>The primary responsibility of a canonical definition is to provide the definition of the “universe” in which products derived from that canon can be effectively shared, compared, etc. As such, the ECCF defines the canon of Terms of Art required to discuss the various relationships between and among multiple artifact instances both within and between component specifications.</td>
</tr>
<tr>
<td><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Compliance</strong> must be fully respected and implemented.</td>
<td><strong>3. ECCF</strong></td>
<td>See comment 2.ECCF</td>
</tr>
<tr>
<td><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Conformance</strong> must be fully respected and implemented.</td>
<td><strong>4. ECCF</strong></td>
<td>See comment 2.ECCF</td>
</tr>
<tr>
<td><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Correspondence</strong> must be fully respected and implemented.</td>
<td><strong>---</strong></td>
<td><strong>---</strong></td>
</tr>
</tbody>
</table>
### Interoperability Specification Matrix

<table>
<thead>
<tr>
<th><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Compatibility</strong> must be fully respected and implemented.</th>
<th><strong>5.</strong> ECCF</th>
<th>See comment 2.ECCF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Localization</strong> must be fully respected and implemented.</td>
<td><strong>6.</strong> ECCF</td>
<td>See comment 2.ECCF</td>
</tr>
<tr>
<td><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Traceability</strong> must be fully respected and implemented.</td>
<td><strong>7.</strong> ECCF</td>
<td>See comment 2.ECCF</td>
</tr>
<tr>
<td><strong>...SHALL</strong> ensure that the relationships between artifact templates defined in the IG’s IST are related according to the semantics of the Terms of Art defined in the SAIF CD. Specifically, the semantics of the term <strong>Provenance</strong> must be fully respected and implemented.</td>
<td><strong>8.</strong> ECCF</td>
<td>See comment 2.ECCF</td>
</tr>
<tr>
<td><strong>...SHALL NOT</strong> change the semantics of any of the navigational terms defined in the SAIF CD.</td>
<td><strong>9.</strong> ECCF</td>
<td>Consistent definitions of the various terms that allow predictable descriptions of and navigation between the various artifact templates that collectively define an IG’s IST is a critical success factor for any IG that is considered compliant with the SAIF CD.</td>
</tr>
</tbody>
</table>

**...SHALL** define **at least one** IG-specific **Interoperability Specification Template (IST)** which constrains and/or explicitly clarifies and instantiates the Behavioral, Informational, and Governance modeling **languages** defined in the SAIF CD as IG-specific modeling **grammars**.
2. **ISM**

In order for component specifications to be effectively and efficiently compared between Shared Purpose stakeholders, the *lingua franca* of the comparison must be standardized to a certain degree. This standardization is, in part, accomplished by having component specifications be defined using a standard template, i.e. the IST, whose columns and rows have clearly specified definitions derived from ODP (columns) and software engineering process (rows).

3. **ISM**

See comment 2.ISM

4. **ISM**

See comment 2.ISM

5. **ISM**

A given artifact template must clearly define both the content and representation of the artifacts for which it is providing the template in order for template instances to be effectively and efficiently governed.

6. **ISM**

*Conformance* of a given implementation to a given specification is assessed by quantitatively evaluating – through automated, semi-automated, or human-based – the veracity of each *Conformance Assertion* provided by the implementation. As such, a given Conformance Assertion should be pair-wise linked to Conformance Statements defined as part of the content of a given artifact instance. **NOTE:** Conformance Statements do NOT have to be testable via automated means. They do, however, need to be verifiable as TRUE or FALSE without significant inter-rater differences in results, i.e. high inter-rater reliability.
<table>
<thead>
<tr>
<th>...SHALL NOT change the order of the columns or the rows in its IST relative to how they are defined in the SAIF CD.</th>
<th>7. ISM</th>
<th>The SAIF CD provides the <em>lingua franca</em> for discussing both intra- and inter-organizational components from an interoperability perspective. In that context, the IST the framework for collecting component specification artifacts. Changing the semantics or order of the columns or rows of a particular IG’s IST would severely undercut the effectiveness and efficiency of cross-component comparisons or scenario-based interoperability discussions.</th>
</tr>
</thead>
<tbody>
<tr>
<td>...SHALL NOT change the semantics or scope of the columns or rows in the IST as they are defined in the SAIF CD.</td>
<td>8. ISM</td>
<td>See comment 7.ISM</td>
</tr>
<tr>
<td>...SHALL NOT redefine the semantics of the IST’s rows to align with software engineering artifacts (e.g. the OMG’s MDA semantics)</td>
<td>9. ISM</td>
<td>See comment 7.ISM</td>
</tr>
<tr>
<td>...SHOULD ensure that the placement of artifact templates within IST cells is consistent for all artifact template defined by the IG.</td>
<td>10. ISM</td>
<td>Consistency is far more important than particular IST cell locations.</td>
</tr>
<tr>
<td>...SHOULD provide a clear mapping of all IG terms which are renamed or otherwise aliased from SAIF CD terms.</td>
<td>11. ISM</td>
<td>The use of SAIF CD-compliant SAIF IGs facilitate cross-organizational, component-based discussions focused on Shared Purpose scenarios and their objectives based on the <em>lingua franca</em> of the SAIF CD. As such, organizational aliases of SAIF CD concepts within the scope of a given organization’s SAIF IG should be clearly explicated in a map for consumption by stakeholders using other SAIF IGs.</td>
</tr>
</tbody>
</table>
12. ISM

Best Practice dictates that a “complete” SAIF IG should have at least one artifact template for each of the 15 cells in the IG’s IST. It may be the case that certain artifact templates will be defined later in the life cycle of the IG’s development based on the organizational maturity of the organization developing the IG, e.g. Implementable artifact templates may be the easiest to define initially followed by Conceptual and finally followed by Logical as the organization embraces more “formal” architectural procedures and practices. NOTE: A given component’s full specification need not necessarily be defined via a single Interoperability Specification Instance (ISI) and its defined artifacts, i.e. it is expected that solutions will often be defined through a collection of ISI’s, each of which is conformant (since artifacts are implementations of templates) with the artifact template definition specified in the IST. This is the reason that an ISG needs to define at least one artifact template per cell for the 15 IST cells.

13. ISM

Organizations may choose to use “organization-friendly” naming conventions/aliases for SAIF CD terms. However, as noted above, aliases are not allowed to change the semantics of the aliased terms.

14. ISM

Because the columns of the IST are not strictly orthogonal, it is possible for that the semantics defined in a single artifact template may fall under more than one column’s semantics, e.g. an Activity diagram may contain both Informational and Computational semantics and/or may be relevant at both the Conceptual and Logical Perspectives.
8 Appendix

8.1 ISM Specification Matrix, Template and Instance

The SAIF Interoperability Specification Matrix (ISM) defines a structure for categorizing artifacts that collectively describe a complex component or system. As such, the ISM can be viewed as a formal Type. The ISM defined by the SAIF Canonical Definition is ultimately realized as an ISM Profile, referred to as an Interoperability Specification Template (IST) in a particular SAIF IG. An IST defined by a particular SAIF IG specifies the content and representation of specific artifacts in the various dimensions and perspectives of the ISM.

<table>
<thead>
<tr>
<th>Specification Subject</th>
<th>Enterprise</th>
<th>Informational</th>
<th>Computational</th>
<th>Engineering</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>Business Context Reference Content</td>
<td>Domain Analysis (Information) Model</td>
<td>Collaboration Analysis, Functional Profiles, Service Roles and Relationships</td>
<td>Existing Platform capabilities</td>
<td></td>
</tr>
<tr>
<td>Implementable</td>
<td>Rules, Procedures</td>
<td>Localized Information Model, Transforms, Schemas</td>
<td>Collaboration scripts, Orchestrations, Realized Interfaces</td>
<td>Execution Context, Platform Bindings, Deployment Model</td>
<td>Security Services Routing Services</td>
</tr>
</tbody>
</table>

Figure 23 Exemplar Interoperability Specification Matrix

Figure 23 depicts an exemplar Interoperability Specification Template (IST) containing named artifacts, the specific content and representation of which would be formally defined in the SAIF IG in which the IST was defined.

Once the requirements for specifying artifacts have been defined, multiple instances are produced using the appropriate tools and technologies. Each instance contains actual artifacts whose content and representation are conformant to the criteria specified in the IST. A specific collection of artifacts describing a particular component – e.g. service, message, document, etc. – is referred to as an Interoperability Specification Instance (ISI), i.e. an ISI is an instance of an IST.

Finally, a given ISI may then be implemented via one or more specific implementations, each of which may be evaluated for its conformance to the specification instance through the evaluation of implementation-specific Conformance Assertions which are made and linked pair-wise to associated Conformance Statements in the specification instance as illustrated in the following graphic:
Figure 24 Another view of an IST

Figure 24 depicts another view of an IST notated to indicate some of the specific relationships defined by the language of the ECCF. Note the presence of Localizations between each Perspective as well as between the Implementable Perspective and candidate implementations. Specific Localization semantics are an example of one type of contextualization that a SAIF IG may make in its application of the SAIF-CD languages.
Figure 25 Binding II to SI through Conformance Assertions

Figure 25 depicts the graphical representation of the binding of an implementation instance to a specification instance through the use of testable Conformance Assertions made by the implementation against pair-wise Conformance Statements defined in the Interoperability Specification Instance.
Figure 26 shows the relationship between the ISM, the IST, and ISIs. The Interoperability Specification Matrix (ISM) is a type as defined in the SAIF-CD. The Interoperability Specification Template (IST) is a profile which is defined in each SAIF IG through the application of restrictions and specializations of the ISM language. The multiple component specification, referred to as Interoperability Specification Instances (ISIs), are instances of the artifact content and representations specifics defined in the IST. Note that the terms “type,” “profile,” and “instance” are represented in the illustration as UML-like stereotypes.

Note that neither the definition of the ISM nor its realization in a given SAIF-IG as an IST specifies a process whereby a given matrix instance is to be populated. That is, there are no rules such as “all of the required artifacts in the Conceptual row of the ISM should be fully specified before artifacts in the Logical row are specified.”

Each ISI has a particular scope, for example, system, sub-system, or service, i.e., a scope that is defined by the collection of artifacts in the ISI. The scope of the ISI is referred to as the Specification Subject (SS). Each cell in an ISI can contain multiple artifacts which may or may not contain artifact-to-artifact links or relationships, and which may be hierarchical in terms of level of detail or abstraction.

The normative content of the Enterprise Conformance and Compliance Framework of the SAIF Canonical Definition is the definitions and details of the various inter-cell and inter-artifact relationships. Refer to the discussion in the ECCF chapter.

Given a particular ISI that, by definition, contains artifacts that collectively specify a given component from the perspective of one or more interoperability scenarios, one or more development teams can develop an implementation of the specification, thereby “binding” a specific implementation instance to the specification.

The ECCF chapter of the SAIF Canonical Definition establishes the concept of conformance of a given implementation instance to a given ISI in terms of evaluation of specific Conformance Statements made within specification artifacts, and the Boolean veracity of those statements to Conformance Assertions made by a given implementation instance. These concepts are discussed more fully in the ECCF chapter of this document.

In summary:

- The artifacts collected in a given ISI contain descriptions of a given component’s informational or static and behavioral or dynamic semantics, features and functions.
• Specifications regarding a component’s informational or static semantics and other informational aspects are expressed using the Information Framework grammar.

• Specifications regarding a component’s behavioral or dynamic semantics and other behavioral aspects are expressed using the Behavioral Framework grammar.

• The relationships between artifacts within and between cells, row-by-row, column-by-column, or column-by-row basis, are defined using the Enterprise Conformance and Compliance Framework grammar.

• The content and representation of each artifact must be defined in the context of the organization’s SAIF IG.

• The overall management of the life cycle of each artifact, including the correctness and completeness of the artifact as well as RACI relationships for the artifact, are defined by the Governance Framework grammar.

8.2 Foundational Principles

The material in this section is not part of the Canonical Definition of HL7 SAIF. It is included to provide context for the definitions of the four SAIF-CD Frameworks. Four Foundational Principles are discussed:

1. Shared Purpose

2. Fowler’s Accountability Pattern

3. “Service-Awareness”

4. Relationship of SAIF-CD to the Reference Model for Open Distributed Processing (RM-ODP)

8.2.1 Shared Purpose

Shared Purpose between participating parties is manifested in cross-enterprise or cross-organizational interoperability, i.e. communication across organizational boundaries. Both parties must decide on the multiple details that collectively define an interaction or set of interactions. There must be an agreed upon value received for cost and effort expended. At minimum, the basic dimensions of a Shared Purpose agreement answer the questions “who,” “what,” and “when.”

A Shared Purpose is at the heart of any successful instance of technical interoperability. Successful execution of a Shared Purpose agreement as it is realized in technology depends on explicit definition and representation of contracts, roles, interactions, behaviors, accountabilities, policies, and enforcement (governance). The SAIF-CD has leveraged considerable work by multiple sources in the area of Shared Purpose, in particular by adopting and adapting material from:

• Martin Fowler—Accountability pattern

• SOA literature—conceptual notion of “service-awareness”

• Reference Model for Open Distributed Processing—selected terminology (ISO RM-ODP)

Discussion follows of the contribution and context of each of these resources as used in the SAIF-CD.
8.2.2 Fowler’s Accountability Pattern

The Accountability Pattern of Martin Fowler (Fowler & Feathers, 1997) defines the notion of a Contract through the explicit representation of Accountability, that is, a Commissioning Party establishes a contract with a Responsible Party to accomplish one or more tasks. The success of the Responsible Party’s actions can be assessed by the Commissioning Party via one or more agreed-upon Accountabilities which can take a form such as deliverables or tasks executed (Fowler & Feathers, 1997).

Although not shown in the diagram, Fowler’s Accountability pattern formalizes the notion of a contract as a “collection of accountabilities” which have been agreed to by the Commissioning and Responsible Parties between whom the contract is established. Accountabilities are assumed to be the result of behaviors on the part of either or both parties (more likely the Responsible Party), and a variety of interactions between the two Parties can also be described in the context of Accountabilities. For example, in order to accomplish a particular task, the Responsible Party may need the Commission Party to do something first. Also implicit in the diagram is the notion that the contract exists for a specified period of time.

Although some of the terminology used by this pattern—Commissioning Party, Responsible Party, is not used in the SAIF-CD, it is replaced and elaborated upon by specific language from the Reference Model for Open Distributed Processing.

8.2.3 “Service-Awareness”

The Service Aware Interoperability Framework Canonical Definition (SAIF-CD) has matured and evolved over the three years since the HL7 Chief Technology Officer asked the HL7 Architecture Board (ArB) to provide a roadmap and specific deliverables that would result in development and specification of an enterprise architecture for HL7. In that time, there has been considerable confusion over the term “service-aware.” In contrast, the term “interoperability framework,” although broad with respect to the exact type of interoperability, is much less subject to confusion.

The “Service-Aware” in the SAIF-CD indicates that the behavior of a given component is the primary classifier of that component from the perspective of the component’s involvement in an interoperability scenario focused on achieving a shared purpose. Other terms are often associated with design, implementation, or run-time specifics that are important but secondary to characteristics that define the expected interaction-based behavior of a given component. As a consequence, the term “service-aware” replaces other concepts often used to describe a component, including those based on specific implementation technologies and information-exchange types.

The term “service-aware” is used as the primary identifier of the frameworks of the SAIF-CD because each of the concepts is overtly considered when working an environment based on contemporary service-based architecture paradigms. Examples are SOA and service-based technologies such as SOAP or REST paradigms. The concepts
can also be realized in a non-service environment assuming there is a commitment to formalizing the semantics of interactions. The ArB chose the term “service-aware” to underscore the importance of these core concepts where the requirement for interoperable interactions is of central importance. The SAIF-CD and any conformant SAIF-IG can be operationalized without the use of service-based technologies. Interoperability scenarios to achieve Shared Purposes can productively be executed using approaches based on messages, documents, or other hybrid strategies and technologies. However, definition and specification of every scenario, without regard to implementation technology, relies on certain core concepts and constructs that are collectively defined as bringing “service-awareness” to the discussion. These concepts, most of which are at least implicit in Fowler’s Accountability pattern and which are elaborated in RM-OPD, include:

- Role (a scenario-specific application of Fowler’s Party)
- Behavior
- Contract
- Interaction
- Accountability
- Policy (not covered in Fowler although it is implicit in Contract)
- Exchanged Information (not covered in Fowler although it is implicit in Accountability)

The following diagram shows the core concepts and relationships that result from contextualizing and making explicit the semantics of Martin Fowler’s Accountability pattern in a Service-Aware framework such as the SAIF-CD.

![Figure 28 Shared purpose concept map](image)

A Shared Purpose is defined by two or more parties and is explicitly described in a contract. The SOA literature refers to implementation-based parties in terms of Roles rather than the more general notion of Party, recognizing the fact that a given instance of a Party can assume more than one Role. Roles (that is, time-limited capabilities and competencies) are capable of executing specific behavior, a subset of which is relative to the contract-of-interest and
referred to as Interactions. Contract-specific Interactions may require the exchange of Information as specified in the Contract. Contracts also specify Accountabilities (i.e. Deliverables and/or Tasks to be completed) and Policies (which may constrain or modify Accountabilities)

8.3 Defining a SAIF Implementation Guide

8.3.1 “SAIF enough – the Linear Value Proposition”

A common misunderstanding regarding the application of the SAIF Canonical Definition to a given enterprise revolves around the two-part question:

- What artifacts should be included in the enterprise’s SAIF-IG?
- Given the artifacts specified in the SAIF-IG, does each component need to be fully specified in order to be considered SAIF-IG-compliant?

During the development of the SAIF-IG, at the Center for Biomedical Informatics and Information Technology (CBIIT) of the National Cancer Institute (NCI), the concept of “just enough specification” was introduced in response to the second question. It became clear that the answer to the question was a definitive NO, that is, all components did not have to be equally well-specified. Further, the best method for determining how much effort to devote to a given component’s specification is a value-proposition-based decision based on understanding both the Deployment Context in which the component would be involved in interoperability scenarios, and the Interoperability Type required by those scenarios. Well-localized Deployment Contexts requiring “only” syntactic interoperability require minimal semantic specification using the various ISM artifacts defined in the CBIIT SAIF IG. As the Deployment Context becomes larger and the Interoperability Type moves from Syntactic to Computable Semantic (or both), the requirements for increased levels of explicit specification increases.

The important concept that emerged was what CBIIT terms the “linear value proposition,” that is, easy things such as deploying PERL code in a single lab, should be easy; harder things should be harder, and really hard things such as deploying a service into the global community with the requirement that it support machine-to-machine computable semantic interoperability, should be the hardest.

Figure 29 Deployment Context versus Interoperability Type matrix (courtesy of NCI Center for Biomedical Informatics and Information Technology (NCI CBIIT)
8.3.2 Deployment Context versus Interoperability Type

A Deployment Context is “the size and/or diversity of the community that is negotiating one or more shared purpose scenarios.” For a given Deployment Context, the Interoperability Type (that is, the specific requirements for the level of interoperability needed between a given component and other components in the same Deployment Context (such as Syntactic, Human Semantic, or Computable Semantic) may vary. As the size or diversity of the Deployment Context increases and/or the Interoperability Type becomes more computation-centric, the requirements for explicit representation of technical details of the involved components increases. The SAIF-CD supports the notion of a “linear value proposition” by enabling an environment where “just enough specification” to tractably satisfy the requirements of a given shared purpose scenario can be defined and managed. (Graphic courtesy of the Center for Biomedical Informatics and Information Technology (CBIIT) of the National Cancer Institute (NCI)).

8.3.3 Defining Specification Artifacts: Content, Representation, Location

As indicated above, the canonical representation of SAIF does not specify the content, representation, or location of individual artifacts. Artifact specification is, instead, done in the context of a given enterprise’s SAIF-IG. (Note that several SAIF-IGs have been and are being developed by HL7, the US Department of Defense, Canada Health Infoway, Australia NeHTA (National eHealth Transition Authority), and the Center for Biomedical Informatics and Information Technology (CBIIT) of the NCI and are generally available for review and study.)

In general, the most important aspect of artifact specification is its content, followed by its representation. Its location in a given ISI is really only of major importance with respect to the consistency of the location of a given artifact (or, more correctly, artifact type) across multiple specification instances within the context of an IG.

In addition, a given artifact may occur in more than one ISI cell, a reflection of the fact that the Dimensions and Perspectives of the ISI matrix are not normalized (as would be the case, for example, if the ISI were instantiated using the Zachman2 matrix of Dimensions x Perspectives). From the perspective of interoperability scenarios, normalization and cell-specific location are not as important as explicitness and consistency.

8.3.4 Building SAIF Specifications

From a standards development point of view, the SAIF is about providing sets of artifacts that can be compiled in specifications to discuss the terms of interoperability for a particular subject or topic. The Interoperability Specification Matrix is therefore concerned mainly with providing the means by which implementation groups, realms, or enterprises will describe these terms.

By itself, the Canonical SAIF does not provide sufficient foundation to achieve a shared purpose interoperability scenario. A given Implementation Guide must also provide

- Sets of principles used to craft specifications
- Discussion of the concepts being used from the SAIF, additional concepts, and refinements if necessary
- Templates for specifications that will include artifact types, cardinality of concepts, optionality, choices of interaction and communication patterns, and other characteristics as needed.
- Potential sample choices for artifact selection
- The implications for conformance when using a given artifact

Thus, while the Canonical SAIF provides a framework for what concepts need to be expressed and why they need to be expressed, it cannot denote how to express them, when an artifact surfaces methodologically, or where an artifact will be realized.

An implementing enterprise can also specify terms of compliance for HL7 specifications. For example, it may be useful for HL7, as a SAIF-implementing enterprise, to say that in certain Logical specifications, all information
models need to be compliant with the RIM. All Implementation Guides will not be created equal, and may use different artifacts to demonstrate the same SAIF concept. Implementation Considerations

8.4 Governance Implementation Considerations

Governance is a means to reduce risk. What is governed is dependent on the shared purpose. A common understanding and agreement on a shared purpose is the first order of business in establishing a community. Aspects of interoperability that need to be governed include, but may not be limited to:

- Community participation refers to what parties in what roles are eligible to participate and what are the prerequisites for their participation.

- Policies refers to those policies within each party’s jurisdiction that influence the interoperability behavior of participating systems. Systems may encode business rules that are not explicitly specified but cause incompatibilities in exchanged information or unanticipated behavior of participating systems. Aligning policies across jurisdictional boundaries is one of the most difficult tasks of a federated community.

- Identity management refers to how instances of people, people in roles, systems, technical components, information artifacts and other factors are to be uniquely identified and tracked through processes included within the scope of interoperability.

- Artifact definition and approval refer to the change management process for each type of artifact, which may be for that artifact only and may be independent from other types. Artifacts may be dependent on one another and the relationships among them must be explicit and also tracked. In the SAIF context, the full slate of ECCF artifacts are interdependent and must be managed as a coherent whole in order to support technology components that are fit for purpose and whose interoperability capabilities are consistent with each other.

- Technology component configuration refers to system interoperability for potentially multiple dependent components each having their own change management processes while being interdependent. The usual system lifecycle of development, testing and deployment is increasingly complex in an interoperability environment. Multiple technical architectures can interoperate effectively if their interfaces are conformant to specifications that constrain the behavior across system boundaries to enable consistent operations.

- Accountability refers to accountability for the completeness, quality, integrity and security of information that originates in one system and is transmitted to and used by another.

- Change management refers to an essential element in collaborations, as interdependent parts often undergo change on different schedules. The ability to assess the impact of change prior to implementation can minimize anticipated disruption as changes occur. Continual change is the expected state in a volatile environment and flexible designs and evolutionary implementation are reasonable responses.

8.5 SAIF-CD Adoption and Adaptation of existing and/or related work

With respect to the criticism voiced by several members of the community that the SAIF-CD specification is not sufficiently aware of existing work, it is important to understand that the SAIF Canonical Definition defines common concepts and patterns that will subsequently be instantiated through the concrete artifact specification definitions in the various IGs. The reuse of existing work is thus – for the most part – an IG-level and not a Canonical Definition-level issue.
The ArB does not agree with statements that suggest that SAIF is not aware of work in other groups, for example, OASIS, UML/OMG, and TOG. SAIF makes considerable use of the ODP’s Enterprise and Computational languages. In particular, the development of the UML profile for ODP and other UML specifications, for example, SoaML, MOF, and certain aspects of UML 2.x, have been directly influenced by ODP. Finally, there is considerable alignment between ODP and the latest OASIS SOA Reference Architecture Foundations and the TOGAF 9 meta-model. All of these developments and correspondences underscore the validity of the ArB’s choice to use ODP as the basis for the SAIF Canonical Definition.

However, the ArB does believe that many of these efforts cited above are insufficiently focused on the important issue of the explicit representation of computationally-capable static and behavioral semantics, that is, they do not *a priori* start from the position of “interoperability as a 1st-class citizen.”

The efforts tend to be focused on a single enterprise rather than taking a cross-enterprise view and, as a result, do not bring sufficient rigor to the importance of cross-enterprise standards at both the human and technology level in the larger context of understanding component capabilities from a cross-enterprise interoperability perspective; and the efforts do not explicitly define their various “viewpoints” from multiple role-based perspectives, a feature that is essential in surfacing critical component characteristics from an interoperability perspective.
<table>
<thead>
<tr>
<th>Page</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>Interoperability Specification Matrix. See</td>
</tr>
<tr>
<td>106</td>
<td>Jurisdiction, 17</td>
</tr>
<tr>
<td>107</td>
<td>language, 4</td>
</tr>
<tr>
<td>108</td>
<td>Language, 4</td>
</tr>
<tr>
<td>109</td>
<td>Language (SAIF-CD), 4</td>
</tr>
<tr>
<td>110</td>
<td>Localization, 42</td>
</tr>
<tr>
<td>111</td>
<td>Logical Information Model, 36, 37</td>
</tr>
<tr>
<td>112</td>
<td>Logical Perspective, 6, 23, 42, 46</td>
</tr>
<tr>
<td>113</td>
<td>Machine Computable, 14</td>
</tr>
<tr>
<td>114</td>
<td>Management, 13</td>
</tr>
<tr>
<td>115</td>
<td>meanings, 4, 30, 39</td>
</tr>
<tr>
<td>116</td>
<td>methodology, 13, 35, 36</td>
</tr>
<tr>
<td>117</td>
<td>Methodology, 13</td>
</tr>
<tr>
<td>118</td>
<td>Metrics, 20</td>
</tr>
<tr>
<td>119</td>
<td>morpheme, 4</td>
</tr>
<tr>
<td>120</td>
<td>Morpheme, 4</td>
</tr>
<tr>
<td>121</td>
<td>morphology, 4</td>
</tr>
<tr>
<td>122</td>
<td>Object, 25, 46</td>
</tr>
<tr>
<td>123</td>
<td>Objectives, 9, 11, 19</td>
</tr>
<tr>
<td>124</td>
<td>Obligation, 25</td>
</tr>
<tr>
<td>125</td>
<td>obligations, 14, 19, 21, 24</td>
</tr>
<tr>
<td>126</td>
<td>Operation, 25, 26</td>
</tr>
<tr>
<td>127</td>
<td>operations, 9, 13, 18, 21, 22, 23, 25, 26, 27, 57</td>
</tr>
<tr>
<td>128</td>
<td>Party, 17</td>
</tr>
<tr>
<td>129</td>
<td>patterns, 10, 11, 34, 56</td>
</tr>
<tr>
<td>130</td>
<td>People (Roles), 19</td>
</tr>
<tr>
<td>131</td>
<td>Perceptual, 41</td>
</tr>
<tr>
<td>132</td>
<td>permission, 18, 25</td>
</tr>
<tr>
<td>133</td>
<td>Permission, 25</td>
</tr>
<tr>
<td>134</td>
<td>permissions, 24</td>
</tr>
<tr>
<td>135</td>
<td>Perspective, 6, 10, 22, 23, 40, 42, 45, 46, 47, 49</td>
</tr>
<tr>
<td>136</td>
<td>Perspectives, 6, 9, 10, 22, 23, 28, 42, 43, 45, 46, 56</td>
</tr>
<tr>
<td>137</td>
<td>policies, 9, 17, 18, 19, 20, 21, 24, 25, 52, 57</td>
</tr>
<tr>
<td>138</td>
<td>Policies, 9, 19, 25, 54, 57</td>
</tr>
<tr>
<td>139</td>
<td>Policy, 25, 54</td>
</tr>
<tr>
<td>140</td>
<td>Post-Condition, 26</td>
</tr>
<tr>
<td>141</td>
<td>post-conditions, 26, 45</td>
</tr>
<tr>
<td>142</td>
<td>Precepts, 19</td>
</tr>
<tr>
<td>143</td>
<td>Pre-Condition, 26</td>
</tr>
<tr>
<td>144</td>
<td>pre-conditions, 14, 26</td>
</tr>
<tr>
<td>145</td>
<td>pre-coordinated concepts, 29</td>
</tr>
<tr>
<td>146</td>
<td>primitive concept, 29</td>
</tr>
<tr>
<td>147</td>
<td>Process, 27</td>
</tr>
<tr>
<td>148</td>
<td>processes, 9, 13, 14, 18, 19, 20, 21, 23, 25, 27, 40, 46, 57</td>
</tr>
<tr>
<td>149</td>
<td>Processes, 19</td>
</tr>
<tr>
<td>150</td>
<td>profiles, 7, 44, 45</td>
</tr>
<tr>
<td>151</td>
<td>Programmatic, 41</td>
</tr>
<tr>
<td>152</td>
<td>Prohibition, 25</td>
</tr>
<tr>
<td>153</td>
<td>prohibitions, 19, 21, 24</td>
</tr>
<tr>
<td>154</td>
<td>Provenance, 18, 42</td>
</tr>
<tr>
<td>155</td>
<td>reference information model, 36</td>
</tr>
<tr>
<td>156</td>
<td>reference model, 23, 37</td>
</tr>
<tr>
<td>157</td>
<td>Responsibility, 18</td>
</tr>
<tr>
<td>158</td>
<td>responsible role, 13, 18, 22, 24, 25, 26</td>
</tr>
<tr>
<td>159</td>
<td>Revitalization, 21</td>
</tr>
<tr>
<td>160</td>
<td>Risk, 17</td>
</tr>
<tr>
<td>161</td>
<td>RM-ODP, 4, 5</td>
</tr>
<tr>
<td>162</td>
<td>Role, 11, 25, 54</td>
</tr>
<tr>
<td>163</td>
<td>SAIF IG, 1, 4, 5, 6, 7, 8, 9, 10, 23, 27, 39, 44, 46, 48, 51, 52, 55</td>
</tr>
<tr>
<td>164</td>
<td>SAIF Implementation Guides, 3, 4, 7, 8, 14, 23, 39</td>
</tr>
<tr>
<td>165</td>
<td>SAIF-CD, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 22, 26, 41, 44, 45, 51, 52, 53, 54, 56</td>
</tr>
<tr>
<td>166</td>
<td>Semantic Types, 33</td>
</tr>
<tr>
<td>167</td>
<td>semantics, 3, 5, 8, 9, 10, 11, 12, 22, 23, 24, 25, 26, 27, 28, 30, 33, 35, 37, 39, 40, 42, 45, 49, 52, 54</td>
</tr>
<tr>
<td>168</td>
<td>semiotics, 4</td>
</tr>
<tr>
<td>169</td>
<td>Sequence flow, 27</td>
</tr>
<tr>
<td>170</td>
<td>Sequence flows, 27</td>
</tr>
<tr>
<td>171</td>
<td>Service, 1, 5, 25, 52, 53, 54, 60</td>
</tr>
<tr>
<td>172</td>
<td>service-aware, 53</td>
</tr>
<tr>
<td>173</td>
<td>Service-Aware, 53</td>
</tr>
<tr>
<td>174</td>
<td>Service-Awareness, 53</td>
</tr>
<tr>
<td>175</td>
<td>shared purpose, 4, 5, 7, 9, 10, 13, 14, 15, 16, 17, 18, 21, 22, 23, 24, 42, 53, 56, 57</td>
</tr>
<tr>
<td>176</td>
<td>Shared Purpose, 14</td>
</tr>
<tr>
<td>177</td>
<td>signature, 26</td>
</tr>
<tr>
<td>178</td>
<td>Signature, 26</td>
</tr>
<tr>
<td>179</td>
<td>signs, 4</td>
</tr>
<tr>
<td>180</td>
<td>Standards, 9, 19</td>
</tr>
<tr>
<td>181</td>
<td>syntax, 4, 5, 9, 10, 11, 32</td>
</tr>
<tr>
<td>182</td>
<td>Table of Contents, 2</td>
</tr>
<tr>
<td>183</td>
<td>Table of Figures, 3</td>
</tr>
<tr>
<td>184</td>
<td>Technology component configuration, 57</td>
</tr>
<tr>
<td>185</td>
<td>Technology Dimension, 45</td>
</tr>
<tr>
<td>186</td>
<td>template, 3, 19, 33, 37</td>
</tr>
<tr>
<td>187</td>
<td>terminology, 10, 30, 31, 32, 34, 37, 46, 52, 53</td>
</tr>
<tr>
<td>188</td>
<td>terminology binding, 34</td>
</tr>
<tr>
<td>189</td>
<td>The Behavioral Framework, 9</td>
</tr>
<tr>
<td>190</td>
<td>The Governance Framework, 9</td>
</tr>
<tr>
<td>191</td>
<td>The Information Framework, 10</td>
</tr>
<tr>
<td>192</td>
<td>TOGAF, 5, 12, 60</td>
</tr>
<tr>
<td>193</td>
<td>Traceability, 42</td>
</tr>
<tr>
<td>194</td>
<td>transactions, 22</td>
</tr>
<tr>
<td>195</td>
<td>Value Sets, 33</td>
</tr>
<tr>
<td>196</td>
<td>Zachman2, 5, 46, 56</td>
</tr>
</tbody>
</table>
9 Works Cited


