SAIF-Refactored Efficiency Interpolation in the HL7 Specifications Development Paradigm

Abstract : The health standard *Health Level 7-Service-Aware Interoperability Framework* (HL7-SAIF) is the most popular and widely-used healthcare-related standard in global operation today. Originally introduced as *HL7* in 1987 and recently melded with the *SAIF* technology, the standard has been *embraced* by the *National Health Services* of the most developed economies in Europe, North and South America, and Australasia.

However, the standard is not without issues. Its current version *v3* which supports *Semantic Interoperability*, the overarching and meaningful exchange of healthcare information amongst participating healthcare enterprises, has been found to be difficult to implement and maintain. A principle component of the *HL7-SAIF v3* development paradigm is the recently integrated *SAIF* component which presents a significant and worrisome element of ambiguity. These issues in essence *subvert* quality specifications development, which permeates to difficult system implementation and sub-standard performance in operation.

This research analysed many of the prevalent *SAIF* issues indepth, and effected smart, delicate, and prudent *refactoring* of this principle *semantic interoperability* driver, to derive optimal efficiencies in specifications development and implementation.

Keywords : *Services-Aware Interoperability Framework* , *Health Level 7*,

1. Introduction

Originally *SAIF* focused on *Working Interoperability* (WI) amongst all *Enterprise* components, and also between participating *Enterprises* in a network. "*The scope of SAIF is the interoperability space between business objects, components, capabilities, applications, systems, and enterprises*". *WI is the instance of two "trading partners", ie., human beings, organizations, or systems, successfully exchanging data or information, for coordinating behaviour, to accomplish a defined task, or both*" [1]. *SAIF* operates on creating *WI*, irrespective of the specific paradigm, ie., *Messages*, *CDA*, or *Services*.

SAIF provides cross-specification *Conformance*, *Compliance*, and *Coherency* validation amongst all *Interoperability Artifacts*, both laterally and vertically, affording complete traceability from the inceptive Enterprise/Business requirements, to the final specifications. This is done by the *Enterprise Conformance and Compliance Framework* (ECCF), one of the four *SAIF* frameworks, via its *Specification Stack* (SS).

2. Material and Methods

SAIF consists of four foundational Frameworks, namely *Behavioural*, *Governance*, *Informational*, and the *Enterprise Conformance and Compliance Framework* [2].

Behavioural Framework (BF) - Provides the *dynamic* semantics of inter and intra-component behaviour in terms of operations, interactions and collaborations.

Governance Framework (GF) - relates to *HL7* Governance, within an interoperability community. It enables "*enterprises to define explicit, organization-specific policies, standards, and roles*" in regard to *Message*, *Document*, or *Service* related *artifact* design [2]. The framework includes :

- *Precepts* – Objectives, Policies Standards, and Guidelines
- *People* – and their Roles including organizations and systems
- *Processes*
- *Metrics*

Information Framework (IF) - provide *Static Semantics* of *Information Framework Artifacts*, eg., *Reference Information Model* (RIM), *Information Models*, *Data Models*, *DAMs*, *Data Type Bindings*, *Deployment Topologies* (UML Models), and *Business Rules*. Populates the *Information*, *Business/Enterprise*, and *Engineering* Viewpoints of the *ECCF*.

Enterprise Conformance and Compliance Framework (ECCF) - The principle *Data Structure* of the *ECCF* is the *Specifications Stack* (SS) which tests *Conformance Statements* of systems with *Conformance Assertions*. Different levels (layers) of Conformance and Compliance can be represented. This grid-like structure consists of Rows and Columns. Rows are derived from *Model Driven Architecture* (MDA) and vertical viewpoints are from *Reference Model for Open Distributed Processing* (RM-ODP).

Other technologies blended into *SAIF* are *Services-Oriented Architecture* (SOA), *Computable Semantic Interoperability* (CSI), and *Distributed Systems Architecture* (Organizational Context).

The *Specifications Stack* (SS) of the *ECCF* is a 3 x 5 collection of *Conformance Statements* all validated by *Technology Binding*. Validation confirmations are called *Conformance Assertions*. The *ECCF* provides a structured method for the validation of *Conformance Statements* in regard to *Informational* (Static) and *Behavioural* (Dynamic) semantics of Software Components, such as *Messages*, *Documents*, and *Services* [2]. All *ECCF Conformance Statements* are related to *Requirements*, *Business Rules*, and *Objectives* about the future system capability provided by the schema of *System Artifacts*, at different levels of *Interoperability*, ie., cross-specification *conformance*, *compliance*, *consistency*, *traceability*, and *compatibility*. Indeed, *Semantic Interoperability* is provided by the cumulative static, functional, and behavioural semantics.

The *SAIF Implementation Guide* (SAIF IG) is the finalized, *SAIF-compliant metadocument* of the *instantiated* and *organization-specific* implementation.

Fig 1. ECCF Specification Stack [1]

ECCF	Enterprise Dimension "Why" - *Policy*	Information Dimension "What" – Content	Computational Dimension "How" - *Behavior*	Engineering Dimension "Where" - *Implementation*	Technical Dimension "Where" - *Deployments*
Conceptual Perspective	✓ Inventory of o Use Cases, Contracts o Capabilities-Services o Stakeholders o Non-Functional Requirements o Methodologies/Processes o Policies & Regulations o Business Objectives ✓ Business Mission, Vision, Scope	✓ Inventory of o Domain Entities o Stakeholders, Roles o Activities o Associations o Information Requirements o Information Models • Conceptual • Domain	✓ Inventories of o Capabilities-Components, o Functions-Services, ✓ Requirements o Accountability, Roles o Functional Requirements, Profiles, Behaviors, Interactions o Interfaces, Contracts ✓ Functional Service Specifications	✓ Inventory of o SW Platforms, Layers o SW Environments o SW Components o SW Services o Technical Requirements o Enterprise Service Bus ✓ Key Performance Parameter s	✓ Inventory of o HW Platforms o HW Environments o Network Devices o Communication Devices ✓ Technical Requirements
Logical Perspective	✓ Business Policies ✓ Use Case Specifications ✓ Governance ✓ Implementation Guides ✓ Technology Neutral Standards ✓ Worksframes of o Architectural Layers o Components and o Associations ✓ Contracts	✓ State Variables ✓ Information Models o Localized o Constrained o Project ✓ Vocabularies ✓ Value Sets ✓ Content Specifications o Messages o Documents o Services	✓ State Machines ✓ Specifications o Use Cases, Interactions o Components, Interfaces ✓ Collaboration Participations ✓ Collaboration Types & Roles ✓ Function Types ✓ Interface Types ✓ Collaboration Scripts ✓ Service Contracts	✓ Models, Capabilities, Features and Versions for o SW Environments o SW Capabilities o SW Libraries o SW Services o SW Transports	✓ Models, Capabilities, Features and Versions for o HW Platforms o HW Environments o Network Devices o Communication Devices
Implementable Perspective	✓ Business Nodes ✓ Business Rules ✓ Business Procedures ✓ Business Workflows ✓ Technology Specific Standards	✓ Schemas for o Databases o Messages o Documents o Services o Transformations	✓ Automation Units ✓ Technical Interfaces ✓ Technical Operations ✓ Orchestration Scripts	✓ SW Specifications for o Applications, o GUIs o Components ✓ SW Deployment Topologies	✓ HW Deployment Specifications, ✓ HW Execution Context ✓ HW Application Bindings ✓ HW Deployment Topology ✓ HW Platform Bindings

2.1. *Object Management Group's MDA* (Model Driven Architecture) *Levels of Abstraction*

These represent the Rows of the ECCF *Specification Stack* [1].

Computationally-Independent Model **(CIM)** – Conceptual view, part of the Requirements and Analysis Phases; maps requirements to functions, capabilities, behaviours.

Platform-Independent Model **(PIM)** – Logical view of Future Service/System during Design phase. Maps Analysis (Conceptual) model to Logical Model. Platform-independent Business rules, interactions an dependencies amongst Services captured here.

Platform-Specific Model **(PSM)** – Implementable View of the design and implementation phases. Maps PIM artifacts to Platform-Specific realizations. Provides traceability down/upwards in each column (MDA Levels) and across/backwards in each row (*RM-ODP Viewpoints*). Layers of Models possible with for example one *DAM* generating multiple *PIMs* and one *PIM* generating multiple *PSMs*.

CIM to *PIM – DAM* refinement could be restriction/removal of Classes/Attributes, expansion by addition of attributes, addition of Classes, etc.

(10)

2.2. *RM-ODP (*Reference Model for Open Distributed Processing) *Viewpoints* or *Dimensions*

These represent the Columns of the *ECCF Specification Stack* [1].

Enterprise/Business Viewpoint - At *CIM*, captures scope and purpose of service/system in keeping with the enterprise business objectives/rationale for the service/system. Applicable Business standards, usage scenarios (use cases), non-functional requirements, links to other artficacts that collaborate in the service. In addition, Analytic Services are captured in Business Storyboards, Process Diagrams, State Diagrams, Activity Diagrams (UML). The *CIM* specifies the Behavioural Framework conceptually, in terms of Roles, Obligations, community, Behaviour, and Goal. The *PIM* may include additional business rules, standards, and policies, and also provide traceability to the *CIM*. The *PSM* also provides traceability to the *PIM* and *CIM*.

A level is required only as needed. After filling the *CIM*, *PIM*, and *PSM*, an unambiguous specification of the Service/System is obtained in terms of its capabilities, scope, and purpose. The required information to fill the *ECCF-SS* is obtained during Requirements collection and Analysis using Use Case diagrams, Models, and requirements documents. The *ECCF-SS* provides a Data Structure for sound, consistent traceability validation, and conformance/compliance confirmation.

Informational Viewpoint - The *DAM* is the main Informational Viewpoint Artifact. Platform Independent specifications (of the *PIM*) refine raw domain information. The artifact in the *PIM* is a logical Model. The *CIM* Information Model is refined here to add sub-domain Classes and Attributes. At *PIM*, the Metadata, Terminology, Value Sets, and Data Types are annotated to the Model. Business Rules are defined, and the *IM* semantics are completed by defining Query parameters and describing Results Sets. Thereafter, Conformance and Compliance Statements can be made and Asserted.

At *PSM* level, the model is mapped to an actual Database Schema or Message Schema. All Transformations are documented. In the case of Analytic Services, the *IM* of the Messages used by the particular Service maybe documented. The *PSM* will define a Platform-Specific Service Object Model, DataTypes, and Transforms. In short, the *PSM* will capture all deployable artifacts such as the Database Schema, Message Definitions, and Implementation Model with Platform-Specific Rules and Data types.

Thereafter, the derived Platform-Specific specifications (in the *PSM*) provide testable integration points for traceability through the *ECCF-SS* structure levels(layers), from the Enterprise/Business Viewpoint specifications. Hence, layers of the *ECCF-SS* model can capture one *DAM* generating multiple *PIMs*, and one *PIM* generating multiple *PSMs*.

Computational Viewpoint - At *CIM*, the Behavioural Framework-related Viewpoint (Functional) describes structure of Service/System, Capabilities, Restrictions, Service Policies, and Constraints.

At *PIM*, Service-related interfaces, operations, interactions, in short the functional profile of the Service/System is described.

The *PSM* captures the service interface documentation, and the service realization specification. Required orchestration are also documented. For analytical services, the *PSM* captures the deployable artifacts and their inter-linking such as Service Registry Information, Data Encryption and Access Control details, Communication Protocols, Platform-Specific Interfaces, Policies, Constraints, and Orchestration Scripts.

Engineering Viewpoint - At the PSM level, the application/User interface is designed and the deployment model of the Service/System is documented.

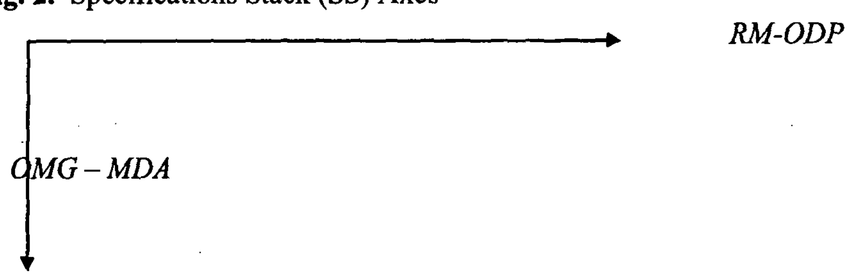
Technology Viewpoint - Tests the Conformance statements collected in the cells.

3. SAIF Issues

This study determined that many issues existed with the foundational structure and the component frameworks of *SAIF*. These are described below.

1. The Framework is constructed utilizing the *Separations of Concerns* (SOC) design approach, ie., *Behavioural* vs *Static Informational.* However clean *SOC* is NOT provided by *SAIF*, the inherent overlap causing the ambiguity present, eg., between columns of the *SS* in *ECCF*. *MDA* and *RM-ODP* are two different technologies which look at the *Requirements Collection* → *Systems Specifications* phases of the *Waterfall Development Model* from different *Viewpoints* and *Level of Granularity*, in the *ECCF SS*. In fact they operate in the same space.

Fig. 2. Specifications Stack (SS) Axes



Hence, instead of independence and discreteness creating a clean separation in the *Row-wise* and *Column-wise* technologies, presently the *SS* is steeped in ambiguity caused by the natural overlap of concerns. Clean *separation of independent concerns* (SOA) in the *SS* cells is non-existent.

2. There is also the possibility of significant ambiguity in the *Behavioural* and *Informational* frameworks. By definition, the *BF* provides the *dynamic* semantics whilst the *IF* provides the *static* semantics for any artifact creation or conformance. But the availability of *DAMs* and use of *UML* technology (such as Activity Diagrams) to model *IF* artifacts connote a dynamic dimension; there is a good chance of overlapping artifacts occurring in both *BF* and *IF*,

causing grave concern to the specifications design process and clean interoperability. The *mis-placement* probability of *Computational Dimension (BF)* artifacts in the *Information Dimension* (IF) is high.

3. This ambiguity also causes some devised artifacts to be duplicated in multiple cells under different *Viewpoints* (Dimensions or columns) in the *Specification Stack* of the *ECCF*.

4. ***Data Services*** and ***Analytic Services*** are captured in the *SS*. Data Services are data-oriented, have input and output data and relatively simple in method. They may provide an object-view to a data resource defined by the information model, or an input query with the result output. ***Analytic Services*** interact with other Services, and are less-data-oriented. Clear capture of Service Semantics very important.

5. The *Reference Information Model* (RIM), an integral component of the *Information Framework* (IF), is itself beset with issues, many documented. For instance, it is strongly *techno-dependant*, being modelled using *XML* (Extended Markup Language).

6. Since the *RIM* is modelled using *XML*, it is suspect is providing a clean separation between the *conceptual*, *logical*, and *physical* abstraction levels, as defined. A logical *RIM* artifact cannot manifest in a multitude of modelling technologies as good *Abstraction* requires, since it is already *instantiated* with *XML* [3].

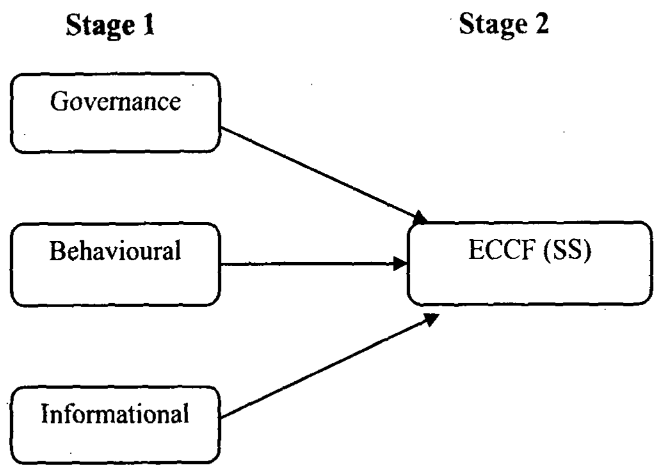
7. The *RIM* has been presented as the *principle Information Ontology* of the *HL7-SAIF* standard. However, this study which focused on deriving optimal sub-process benefits that abound in the *HL7-SAIF* specifications development process, revealed that a common language-based modelling *corridor* that inter-connects the *strongly-coupled Concerns* of the *HL7-SAIF upper ontology*, *RIM*, *messages*, *documents*, *services*, and indeed *SAIF* itself, is the principle requirement to harness sub-process, inter-phase *Working Interoperability*. Currently, technology ranging from *OWL* (Web Ontology Language) for *Ontology* representation, *XML* (Extended Markup Language) for the *RIM* and the three *HL7* paradigms, and *UML* (Unified Modelling Language) together with an assortment of other graphical SAIF representations for dynamic artifact modelling are used. Indeed, a uniform, technology-independent modelling language that overarches all the mentioned development phases would definitely generate sub-process efficiencies and unearth hitherto unseen low-level interoperability, all permeating to greater *Working Interoperability* (WI) and the development of quality specifications.

4. Theory/Calculation

1. The *Governance*, *Informational*, and *Behavioural Frameworks* are *captured/summarised* in the *ECCF-SS*, ie., they feed the *ECCF-SS*. Hence for better *SOC*, a *2-Phase* (Stage) *SAIF* arrangement is proposed.

(13)

Fig. 3. Proposed 2-stage SAIF Structure



2. The present issue with the *ECCF-SS* lacking a clean separation of between the *RM-ODP* dimensions and the *MDA* perspectives is due to the natural overlap in the two technologies. The uncontrolled use of the plethora of modelling technologies, can generate ambiguity and the high-possibility of artifact mis-placement in the *SS* grid. The inherent ambiguity is accentuated by the direct copy of the *MDA* and *RM-ODP* technologies without any adaptation or enhancement. Instead, all overlapping, ambiguous slants in the two technologies should have been removed prior to their *melding* in the *SS*.

3. Once the ambiguity which causes artifacts in the *SS* to be duplicated in multiple cells under different *Viewpoints* (Dimensions or columns) is removed, they would now only occupy singular cells in adherence of the *Do Not Repeat Yourself* (DRY) software design principle.

4. As before, only *Services*-related artifacts are inserted in the *ECCF-SS*, ie., *Data* and *Analytical Services*. The other two paradigmic components *Messages* and *Documents* (Clinical Document Architecture) are also categorized as *Services* for this exercise. The following table lays out the *Services* constituents.

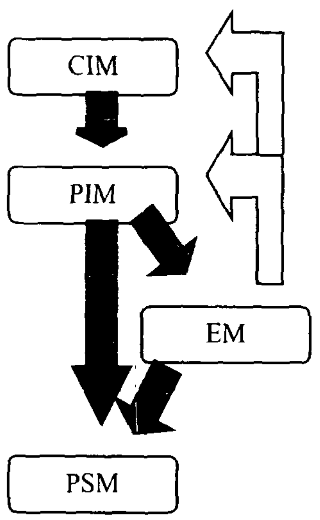
Table 1 : *ECCF-SS* related Services Artifacts

No.	Artifact-related Service	Type
1	***Messages*** - *component to component* (interoperability) - *system to user* (instructional) - *component - system – user* (both)	*Analytic*
2	***Documents*** - *lab device to user* (informative) - *lab device to system* (documentative) - *lab device - system - user* (both)	*Analytic*
3	***System Features and Functions***	*Analytic*
4	***Data*** – *storage, query, retrieval, modification of Static Data*	*Data*
5	***Data*** – *New Data generated through operations-*	*Data*

	related accumulation, assimilation, and/or creation.	

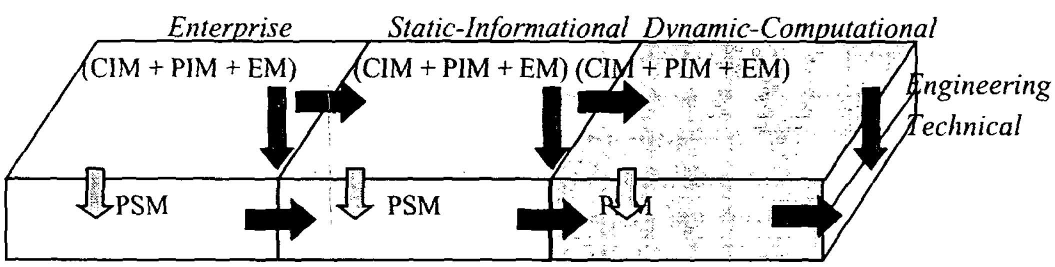
5. The current *SS* structure is enhanced by the addition of the *Empirical Model (EM) Level of Abstraction* for high investment, large, *prototyping-oriented* systems., ie.,

Fig. 4. Proposed Supplemented Levels of Abstraction in the *ECCF-SS*



6. The new, proposed *ECCF-SS* model is given below. It is a *3-D 1 x 9 Single Array*. Each of the dimensions *Enterprise*, *Static-Information*, and *Dynamic-Computation* are further subdivided into three, ie., *CIM*, *PIM*, and *EM*.

Fig. 5. Refactored ECCF-SS Model



7. The *Dimensions* (Viewpoints) or *Columns* in the new, *refactored ECCF-SS* would be
   - *Enterprise/Business*
   - *Static-Informational*
   - *Dynamic-Computational*
   - *Engineering*, and
   - *Technical*

Note : The arrows indicate *correlation*, *cross-checking*, and *cross-referencing*.

8. In order to ensure clean, discrete *SOC*, between *rows*, *columns*, and *cells* in the new *ECCF-SS*, the following strict definitions are made in regard to the *Static-Informational*, *Dynamic-Computational*, *Engineering*, and *Technical* dimensions.

   Static-Informational - contains *strictly* static information related artifacts for conformance testing, modelled using appropriate *static* technologies.

   Dynamic-Computational - contains strictly dynamic (behavioural) *Services* artifacts, both *Data* and *Analytic*, for conformance testing. These *artifacts* (listed in Table 1) are modelled using appropriate *dynamic* technologies. *Data Service* artifacts act upon static information and thus conformity in this case includes *consistency*, *accuracy*, and *precision* preservation of the data.

   Engineering - The new, refactored *ECCF-SS* functionality requires continuous, lateral and longitudinal *cross-checking* and *cross-referencing* with the *Engineering* (software-related) dimension, along the third dimension of the new *ECCF-SS* structure. This activity which is performed throughout from the inception of the conformance testing process, ensures that *compatible*, *feasible*, *practicable*, and *viable* engineering artifacts are devised. Further, complete, contiguous, bi-directional *traceability* is assured within the conformance testing process.

   Technical - The new, refactored *ECCF-SS* functionality requires continuous, lateral and longitudinal *cross-checking* and *cross-referencing* with the *Technical* (hardware-related) dimension, along the third dimension of the new *ECCF-SS* structure. This activity which is performed throughout from the inception of the conformance testing process, ensures that *compatible*, *feasible*, *practicable*, and *viable* technical artifacts are devised. Further, complete, contiguous, bi-directional *traceability* is assured within the conformance testing process.

9. The new, refactored *ECCF-SS* provides traceability of general *Services* artifacts, as well as of *stand-alone data, data bundled in Messages*, and *Message progression*. Thus, complete traceability to domain requirements is provided in these situations.

10. Points 5, 6, 7, under section 3, *SAIF Issues*, all refer to the *techno-dependance* of the *XML* manifestation of the *RIM*. This research has already devised and proposed a *techno-platform-independent* modelling language named the *Unified DataAtom* (UDA) representation which provides a sound and secure solution to the mentioned *RIM-related* issues. The articulation of this *UDA* solution is outside the scope of this paper.

11. A smart blending of the *v3 RIM* with the *Information Viewpoint* (IV) of the *ECCF-SS* is a good approach to achieve greater foundational structure efficiency. Currently, they are related but operate separately and asynchronously. Strong *Coupling* but relatively weak *Cohesion*. The proposed approach is to minimize this coupling to a minimal, if not totally eliminate it, through the blending of *RIM* with *IV*.

Note : *RIM* is already included in *Information Framework* (IF) together with *data types* and *vocabulary mappings*.

5. Results

Let *X* be the source ***ECCF-SS*** (3x5) ***2-D*** representation and *Y* be the target ***ECCF-SS*** (1x9x2) ***3–D*** representation. Let ***T*** be the ***strict*** mapping transformation from *X* to ***Y*** preserving ***completeness***, ***accuracy***, and ***integrity*** of the data represented. Further ***T*** satisfies the necessary condition of the mapping.

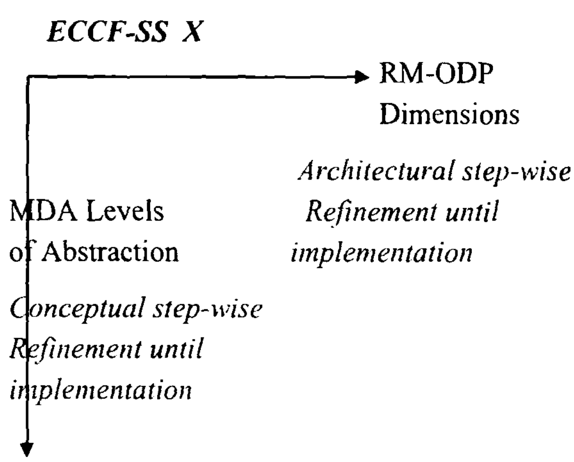
So the ***k*** constituent elements (artifacts) of *X* (ie., elements, wrapper elements, attributes, values) are represented as x_i ($i = 1,2,3,4, \ldots\ldots\ldots, k$). Since ***T*** signifies a strict syntactic and semantic mapping, there is no change in the number of *mapped* elements (artifacts) during the transformation Thus the number of validated artifacts produced in the new *ECCF-SS* set ***Y*** is ***k*** as well.

Required to Prove : In the mapping $T : X \rightarrow Y$ is an ***Equivalence*** relation, meaning the result of the mapping ***T*** produces a target *set* ***Y*** equivalent to the source set ***X***. In essence this means that any artifact successfully validated in ***X***, can also be successfully validated in ***Y***.

Note : the old *ECCF-SS* design ***X*** of fifteen (*3 x 5*) distinct cells referencing fifteen (*mda, rm-odp*) coordinate pairs, is still contained in its entirety in the new *ECCF-SS* design ***Y***, *without the loss of any original semantics*. The only difference being that the original ***2-D*** structure has been *torqued* to a new ***3-D*** shape.

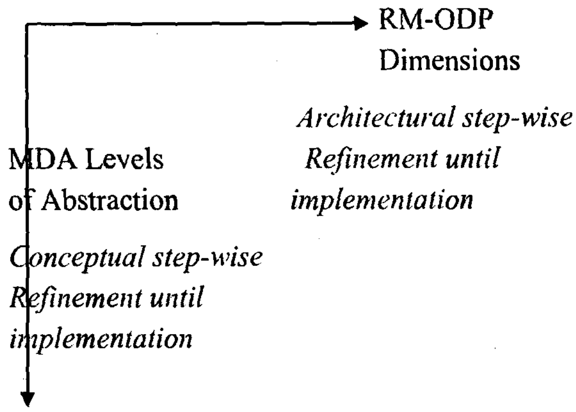
The original semantics of the *ECCF-SS* ***X*** is also preserved.

Fig. 6. Semantics Preservation



The new design *ECCF-SS* ***Y*** is identical and the semantics preservation is complete.

ECCF-SS Y



T is ***Equivalence*** relation if it is ***Reflexive***, ***Symmetric***, and ***Transitive***.

(i) Reflexiveness

Consider any $x_i \in X$, $\{i = 1, 2, 3, 4, \ldots\ldots, k\}$

$T : X \rightarrow Y$ produces *artifacts stream* $\{y_1, y_2, y_3, y_4, \ldots\ldots, y_k\}$ where

$y_i \in Y$, $(i=1,2,\ldots.k)$. (Each y_i is an *artifact* in target *ECCF-SS* ***Y***)

By definition, *T* is *Reflexive* if $\forall\, x_i \in X$, $\{i = 1, 2, 3, 4, \ldots\ldots\ldots\ldots, k\}$

$T : X \rightarrow X$ is True ⟶ (A)

Suppose this is NOT so.

Then $\exists x_m \in X \wedge \forall x_i \in X$ $(i, m = 1, 2, 3, 4, \ldots\ldots, k)$ such that $T : x_i \not\rightarrow x_m$ (B)

x_m lies outside the mapped target region of strict relation *T*

for all source $\forall x_i \in X$ $(i, m = 1, 2, 3, 4, \ldots\ldots, k)$

In general terms this says that a certain artifact *conformance validated* in source *ECCF-SS* ***X*** does NOT MAP to a conformance validation in target *ECCF-SS* ***Y***. It may map outside ***Y***.
This is saying *T* is NOT Reflexive.

Since (B) above is a general relation, subscript *i* can take any value in the range *i*, .. , *k*.
Therefore, we have $i=1,\ T : x_1 \not\rightarrow x_m$

$i=1,\ T : x_2 \not\rightarrow x_m$

$i=1,\ T : x_3 \not\rightarrow x_m$

$i=1,\ T : x_4 \not\rightarrow x_m$ and so on until

$i=1,\ T : x_k \not\rightarrow x_m$

From the above we can therefore write

$x \in X \quad T : x \not\rightarrow x_m$ ⟶ (C)

(18)

This is true because T is *strict* and *complete* so that *every* source constituent *mappable* artifact is transformed with single cardinality to an equivalent conformance validatable target artifact discretely.

(C) $\forall x_i \in X$ $\{T : X \nrightarrow \}$ is NOT TRUE ⟶ (D)

(D) contradicts (A)

Therefore *T* is *Reflexive* as shown in (A)

(ii) Symmetry

Consider any $x_i \in X$, $\{i = 1, 2, 3, 4, \ldots\ldots, k\}$, $y_j \in Y$, $\{j = 1, 2, 3, 4, \ldots\ldots, l\}$

$T : X$ validates *artifacts stream* $\{u_1, u_2, u_3, u_4, \ldots\ldots, u_k\}$ where

$u_i \in X$, $\{i=1,2,\ldots\ldots, k\}$

Similarly,

$T : Y$ validates *artifacts stream* $\{v_1, v_2, v_3, v_4, \ldots\ldots, v_l\}$ where

$v_j \in Y$, $\{j=1,2,\ldots\ldots, l\}$

By definition, T is *Symmetric* if (X, Y) is in T **THEN** (Y, X) is also in T ⟶ (E)

(X, Y) is in T $\{T : X \rightarrow Y\}$ is TRUE

validates *artifacts stream* $\{u_1, u_2, u_3, u_4, \ldots\ldots, u_k\}$ where

$u_i \in X$, $\{i=1,2,\ldots\ldots, k\}$

Similarly,

(Y, X) is in T $\{T : Y \rightarrow X\}$ is TRUE

validates *artifacts stream* $\{v_1, v_2, v_3, v_4, \ldots\ldots, v_l\}$ where

$v_j \in Y$, $\{j=1,2,\ldots\ldots, l\}$

Suppose $\{T : X \rightarrow Y\}$ is TRUE AND $\{T : Y \nrightarrow X\}$ IS NOT TRUE

$\{T : Y \nrightarrow X\}$

$\exists v_j \in Y$, $\{j=1,2,\ldots\ldots, l\}$ say v_m where

$T : v_m \rightarrow u_i$ $\forall u_i \in X$, $i=1,2,\ldots\ldots, k$

However, $v_m \in Y$, a validated artifact in the set Y. Currently v_m does not map to an equivalent *validatable artifact* $u_i \in X$, in relation T. $i=1,2,\ldots\ldots, k$

By definition of T. $\exists v_m \in Y$, $T : v_m \nrightarrow u_i$ ⟶ (F)

But as $\{T : X \rightarrow Y\}$ is TRUE, $\forall x_i \in X$, $\{i = 1, 2, 3, 4, \ldots\ldots, k\}$

validates *artifact stream* $u_1, u_2, u_3, u_4, \ldots\ldots, u_k$ where

$u_i \in X$, $\{i=1,2,\ldots\ldots, k\}$ $\longrightarrow$ (G)

Following from (F), $\exists v_j \in Y$ such that $u_i \in X$, $T: u_i \not\rightarrow v_j$

This v_j is outside the *validated artifact schema* in *Y which is related to X*.

$\forall x_i \in X$ $T: X \rightarrow Y$ is NOT TRUE $\longrightarrow$ (H)

(H) contradicts with (G)

Therefore *T* is ***Symmetric*** as shown in (E)

(iii) Transitivity

Consider any $x_i \in X$, $\{i = 1, 2, 3, 4, \ldots\ldots, k\}$

$T: X$ validates *artifact stream* $\{u_1, u_2, u_3, u_4, \ldots\ldots, u_k\}$ where

$u_i \in X$, $(i=1,2,\ldots.k)$

Now $T: U_{TRANS} \equiv u_1\{ \xrightarrow{T} u_2 \quad T, \xrightarrow{u_3} T, u_4 \rightarrow T, \ldots\ldots \rightarrow \ldots u_k \quad T \quad \rightarrow \}$

$\{u_{11}, u_{22}, u_{33}, u_{44}, \ldots\ldots\ldots\ldots, u_{kk}\}$

Y

Thus, *T* is *Transitive* if $T: X \rightarrow U_{TRANS}$ AND $T: U_{TRANS} \rightarrow Y$ THEN

$T: X \rightarrow Y$ is TRUE $\longrightarrow$ (I)

If (I) is NOT true.

Then $T: X \rightarrow Y$ transformation is not syntactically correct and accurate.

$\exists u_{jj} \in Y$ $\{j = 1, 2, 3, 4, \ldots\ldots, k\}$ where $\forall x_i \in X$, $T: x_i \not\rightarrow u_{jj}$ $\longrightarrow$ (J)

According to the second part of the Transitivity condition above, $\exists u_m \in U_{TRANS}$, $\{m=1,2,.., k\}$
where $T: u_m \rightarrow U_{jj}$ (since *T* is a strict transformation and u_{jj} is a *validated artifact* $\in Y$)

According to the first part of the Transitivity condition above, $\exists x_i \in X$, $i = \{1, 2, 3, 4, \ldots, k\}$
where $T: x_i \rightarrow u_m$ (since *T* is strict transformation and u_m is a *validated artifact* $\in U_{TRANS}$)

According to the latter two statements, $\exists x_i \wedge \exists u_m$ $\{i, m = 1, 2, 3, 4, \ldots\ldots, k\}$
where $T: x_i \rightarrow u_m \wedge T: u_m \xrightarrow{u_{jj}}$
$T: x_i \rightarrow u_{jj}$ (by definition of Transitivity) $\longrightarrow$ (M)

(M) contradicts (J)

Therefore *T* is *Transitive* as shown in (I)

$\rightarrow$

$\rightarrow$

In the mapping $T : X$ Y, T is an ***Equivalence*** relation, meaning the result of the mapping T produces a target *set* Y equivalent to the source set X. This would also satisfy the *necessary* condition for the X Y mapping.

6. Conclusion

The proposed *refactored SAIF* infuses much sought sub-process efficiencies into the *HL7-SAIF* specifications development paradigm. Coupled with proposed overarching common vocabulary *UDA* injection into the *HL7* ontology fabric (the *Upper Ontology* and the *RIM*) strongly-coupled and inter-locked with the allied paradigms of *Messages*, *Documents* (Clinical Document Architecture) and *Services* in the development continuum, this research enabled capitalizing on the abounding merits of the hitherto unexplored sub-process realms. Obviously, the accrued subprocess efficiencies and associated low-level *Working Interoperability* (WI) would promote high-calibre specification generation, which would in turn boost *International Interoperability* and *Inclusive Efficiency* in system operation in the globalised network of participating healthcare enterprises.

Recasting the *ECCF-SS* as a *3-D 1x9x2 Single Array* economises the artifact-related conformance testing exercise. Cross-checking and cross-referencing the *Enterprise/Business*, *Static-Informational* and *Dynamic-Computational* artifacts against the *Engineering* and *Technical* dimensions and artifacts from the inception of the artifact conformance testing process, ensures that *compatible*, *consistent*, *practicable*, *feasible*, and *viable* conformance exists, and that these attributes manifest in the generated specifications. The cross-referencing and checking is *strict*, and occurs laterally across the triple [*Enterprise/Business*, *Static-Informational*, *Dynamic-Computational*] and down the third dimension axis of *Engineering* and *Technical*, and longitudinally across the triple [*CIM*, *PIM*, *EM*] and down the third dimension axis of *PSM*. Further, to eliminate *inter-dimension* (column) ambiguity, the *Static-Informational* and *Dynamic-Computational* dimensions are defined to be strictly *discrete* in terms of the artifacts they accommodate.

Thus, the accrued benefits of sub-process efficiencies and low-level interoperability in the *HL7-SAIF* specifications development process, together with the enhanced and efficient *ECCF-SS* for *strict, unambiguous artifact conformance testing* would ensure and facilitate high-calibre specifications generation, promoting true, network-wide *International Interoperability* and *inclusive efficiency* (our principle goals) during system implementation and operation.

7. References

[1] "*HL7-Service Aware Interoperability Framework (SAIF)*", SAIF Book Introduction for March-April 2011 Ballot, Mar 2011.

[2] Architectural Review Work Group, "*HL7 Service-Aware Interoperability Framework – Canonical Definition, Release 1*", HL7 Informative Document, September 2011.

21

[3] Landgrebe, J., Smith, B., *"The HL7 Approach to Semantic Interoperability"*, ICBO : International Conference on Biomedical Ontology, Buffalo, NY, USA, July, 2011.

[4] Mead, Charles N., *"The Service-Aware Interoperability Framework (SAIF) : Making Cross-Boundary Interoperability a First Class Citizen*", Service Technology Magazine, January 2012.

[5] Health Level 7, *"HL7 Services-Aware Interoperability Framework (SAIF)"*, April 2010.

[6] Oermig, F., Blobel, B., *"Semantic Interoperability between Healthcare Communication Standards through Formal Ontologies"*, Medical Informatics in a United and Healthy Europe, 2009.

[7] *HL7 International New"*, Health Level 7 International, May 2011.

[8] Oermig, F., Blobel, B., *"Semantic Interoperability between Healthcare Communication Standards through Formal Ontologies"*, Medical Informatics in a United and Healthy Europe, 2009.