ICBO: International Conference on Biomedical Ontology
July 28-30, 2011 · Buffalo, NY, USA

The HL7 Approach to Semantic Interoperability

Jobst Landgrebe[1], Barry Smith[2]

[1]Cognotekt LLC, Germany; [2] National Center for Ontological Research, University at Buffalo, NY, USA

Abstract. Health Level 7 (HL7) is an international standards development organisation in the domain of healthcare information technology. Initially the mission of HL7 was to enable data exchange via the creation of syntactic standards for point-to-point messaging. For some 10 years, however, HL7 has increasingly conceived its mission as one of creating standards for semantic interoperability in healthcare IT on the basis of its 'version 3' (v3) family of standards. Unfortunately, v3 has been marked since its inception by quality and consistency issues, and it has not been able to keep pace with recent developments either in semantics and ontology or in computer science and engineering. To address these problems, HL7 has developed what it calls the 'Services-Aware Interoperability Framework' (SAIF), which is intended to provide a foundation for work on all aspects of standardization in HL7 henceforth and which includes HL7's Reference Information Model as general purpose upper ontology. We here evaluate the SAIF in terms of design principles that must be satisfied by a semantic interoperability framework, principles relating both to ontology (static semantics) and to computational behaviour. We conclude that the SAIF fails to satisfy these principles.

1 Introduction

Health Level 7 (HL7) has been producing standards with the goal of enabling interoperability in the health care domain since 1987. Since the end of the 1990s, HL7 has been developing its 'version 3' (v3) set of standards, which have been associated with the goal of achieving *semantic* interoperability, defined as the ability of two or more computer systems to communicate information and have that information properly interpreted by a receiving system in the same sense as was intended by the transmitting system (adapted from [14]).

The central pillar of HL7's attempts to achieve this goal was the HL7 Reference Information Model or 'RIM', initially conceived by HL7 as a modeling language specific to the healthcare domain. HL7's idea was that all data models and message types would henceforth be derived from the RIM in a way that would serve to counteract the formation of local dialects affecting v2 implementations. Unfortunately this strategy has encountered a number of difficulties. The RIM is highly complex and its foundations are idiosyncratic both ontologically [31] and technically [28]. Technically, the RIM is based on a blend of UML [23] and XML 1.0 [4], that is not formally specified, and it deviates in important ways from accepted norms of modeling. This has made the RIM difficult to use, both for humans and for computer systems.

By 2007, the HL7 leadership had come to realise that v3 was not achieving significant uptake. At the same time, the point-to-point messaging-based integration approach traditionally followed by HL7 was itself proving too narrow to meet the demands of the IT industry, where interoperability paradigms rooted in the theory of distributed computing, in ontologies, and in net-centric data standards, were being used with considerable success.

To address these issues, the HL7 leadership initiated in 2007 its Services-Aware Interoperability Framework (SAIF) [12], which is now implemented at the National Cancer Institute as part of the recently scrutinised caBIG initiative [7–9], and which HL7 sees as providing a framework to achieve 'working interoperability' in the E-Health Domain [12].

Our approach in what follows is to identify principles which have been applied in other areas of IT in achieving semantic interoperability, and to evaluate the SAIF in light of these principles. Our analysis concludes that the SAIF fails to satisfy these principles with regard to both architecture and computational behaviour as well as ontology and information modeling.

2 Methods

2.1 Principles for Semantic Interoperability Frameworks

We define a semantic interoperability framework (SIF) as a set of guidelines describing how to create a system architecture for distributed computing [1], whose implementation would allow multiple independent systems to exchange, correctly interpret, reuse and aggregate data provided only that these systems conform to the architecture defined in the framework.

By analyzing the workings of two well established engineering frameworks in which these benefits are realized – IEEE 1471 [15] and the Reference Model of Open Distributed Processing (RM-ODP) [17] – we were able to identify certain basic principles that, we believe, must be satisfied if a proposed framework is to be capable of such realization. We then validated this list of principles by examining further examples, listed in Table 1, of standards specifying guidelines for achieving distributed computational behaviour. We know of no relevant frameworks or guidelines which would not satisfy the principles here listed.

Our principles fall into two groups. First are certain high-level principles which must be satisfied in the authoring of frameworks for any kind of computer architecture. Second are principles specific to the requirements of those architectures that are designed to support semantic interoperability.

General Framework Principles

1. Domain boundaries – a SIF must be associated with a single, coherent domain in which it is declared to be valid, and it must define explicitly the boundaries of this domain.
2. Knowledge reuse – a SIF must build upon established scientific and technical knowledge and best practices in order to minimize the barriers to adoption and, by drawing on what has been validated in use, thereby maximize the likelihood of success.
3. Level of abstraction – a SIF must be sufficiently abstract that it can support systems implemented across a broad spectrum of technical alternatives.

Interoperability Framework Principles

1. Enterprise requirements – a SIF must describe how to identify and formalise the requirements associated with those specific business processes, functions, workflows, and desired outcomes which systems specified and engineered on its basis are intended to support [15, 17].
2. Information model and ontology – a SIF must separate information content from the means by which this content is exchanged, and it must specify on two levels how information content will be structured in such a way as to ensure that it can be communicated successfully from one system to another, first on the level of data through an information model and second on the level of linking data to reality through an ontology [33, 17].
3. Computational model – a SIF must allow the specification both of the behaviour of the single systems conforming to the framework and of the interactions between these systems. [17]
4. Architecture framework and conformance model – a SIF must describe how a system architecture must be structured if it is to be conformant to the framework; thus it must include a list of necessary system components and describe how these in their turn need to be structured in order to conform to the framework [15].

Satisfaction of these principles is not, by any means, sufficient for achieving semantic interoperability: the principles are guidelines only. We believe, however, that they represent necessary conditions. They identify, in effect, component features which belong to the very definition of semantic interoperability. Thus if the principles are violated, semantic interoperability is unachievable.

3 Results and Discussion

3.1 Description and Evaluation of the SAIF

The SAIF attempts simultaneously to address three 'interoperability paradigms', namely *messaging* – which is the traditional method for exchanging data in healthcare environments; *documents* – a term here referring to HL7's XML-based Clinical Document Architecture standard [10]; and *services*, where interoperability relates to the ability of two or more computer systems to communicate information and request specific behavior over service interfaces in such a way that meaning is preserved while details of the internal service implementation are hidden from the user.

To realize semantic interoperability it is necessary to address both structure (of information transmitted) and behavior (of interoperating systems). We shall address the former – which is where ontology comes into play – in the section about the Information Framework below. For the moment we note only that HL7 addresses structure in the same way in its treatment of all three paradigms, namely in terms of the RIM.

When it comes to behavior, however, matters are not so simple. Indeed, the document paradigm does not address behavior at all: a Document, for HL7, is simply a packaged set of information with certain metadata attached. HL7 has tried to address behavior in its messaging infrastructure with its 'dynamic model', which can be compared to modeling the way information is exchanged in a walkie-talkie conversation, and which many in the HL7 community now accept as being outmoded [11]. Only in its treatment of the services paradigm do we enter a domain with a level of technical maturity that can in principle allow the kind of specification of the behavior of modern computer systems that is needed to realise interoperability.

Components of the SAIF. As specified by HL7 [12], the SAIF consists of four components: the Behavioural, Information and Governance Frameworks as well as the Enterprise Conformance and Compliance Framework. We will deal with each of these in turn.

The Information Framework (*IF*). This consists of (*i.*) a set of five principles and (*ii.*) a set of artefacts [12].

i. The first three IF principles essentially state that data, information, knowledge and their respective representations are different from each other and that this difference should be respected in a SAIF-conformant framework. These principles are certainly true, but they are also trivial and it is not clear how they relate to the remainder of the SAIF documentation. The fourth IF principle calls for a separation of what HL7 designates as 'formal concept representation' (the realm of clinical terminologies) from what it calls 'clinical linguistics' (the realm of natural language). It is not clarified what this distinction really means and why it is made.

The fifth IF principle asks for traceability from 'information concepts to organizational/technical concepts and patterns.' Traceability is essential for any engineering framework. One must, for example, be able to prove that certain user requirements justify a certain technical and financial effort. The standard approach to requirements traceability is a formalisation of the requirements which must be satisfied to meet the needs of an organisation followed by a systematic derivation of a stack of engineering artefacts from these requirements. In the SAIF, however, traceability is defined in a direction exactly opposite to that of all the approaches with which we are familiar – by proceeding from information models to organisation models. It is not clear to us how on this basis traceability can be realized.

Taken together, the five IF principles seem to us to have the character of a preamble. They do not provide context for the subsequent sections of the SAIF specification nor are they referenced by it.

ii. Artefacts of the IF include: domain analysis models, reference information models, domain information models, serializeable information models, localized information models, types – classes, attributes, data types, semantic types, and vocabulary (including value sets and value set bindings to attributes).

It is clear from a number of passages (for example where the subject of discussion is the HL7 'cascade' by which information models are 'scoped' and 'specialised by constraint') that when mention is made here and elsewhere in the SAIF documentation to 'reference information models' then it is the HL7 RIM that is intended. Most significant in this respect is the final sentence of section 2.4 of [12]:

In fact, if the reference information model is abstracted to a coarse level of entities and the relationships of those entities through roles to the actions that they somehow participate in then it can be conceptually applicable to any information domain or sector. One can think of a reference information model as an upper ontology that describes the static semantics of all possible real world information *(emphasis added).*

Something new, here, is that the RIM (in effect) is now being described, and advocated, by HL7 as an 'upper ontology'. This is significant because the RIM constrains everything in reality to fall under only three types, namely: Act (meaning roughly: an intentional action performed, for example by a clinician), Role (as, for instance, the nurse role), and Entity (meaning roughly: something containing molecules as parts, including persons). This attenuated repertoire of types causes problems when the RIM is called upon to serve as an ontology framework 'that describes the static semantics of all possible real world information.' Indeed HL7 v3 has from the very beginning faced problems even when applied to health-related information – for example when it comes to dealing with diseases. For the latter are neither Acts nor Entities nor Roles [31].

Similarly, the RIM leaves no room to represent key items in medical reality of other types, such as drug interactions, infections, accidental falls, processes of inflammation – all of which are identified by HL7 as Acts of Observation.

These, and other, technical problems with the RIM make v3 models hard to author and confusing to read, and they bring well-known problems in achieving interoperation with standard vocabularies such as SNOMED CT [29]. Above all, the large number of message-interchange-related (and thus neither ontology- nor information-model-related) attributes built into the core of the RIM imply a failure to separate content from the means for exchanging content [27]. This leads to problems both in data serialisation and in the creation of cleanly computable structures, and explains in turn the lack of practical adoption of the RIM.

The RIM suffers, too, from the idiosyncractic way in which XML has been used to represent both it and its derived models. This is because the RIM exposes design properties of the XML technology to the end user, thereby violating the SIF design principle of abstraction. As an example, consider the way in which metadata structures are related to class attributes in the RIM classes in a nested fashion in direct correspondence with the underlying nested XML structures. This makes it hard for a modeler to understand the usage and meaning of the class attributes and prevents fluent and intuitive modeling with HL7 v3.

In addition to insisting on the RIM both as modeling language and (now) as upper ontology, the IF's description of its artefacts shows also that HL7 plans to adhere to its existing approach as concerns overall modeling and vocabulary usage. Rather than seizing the opportunity to use well established information modeling practices that are in use across the information technology industry in order to simplify the tasks faced by healthcare information modelers, IF's treatment of artefacts serves rather to justify the existing, complex and in many ways idiosyncratic HL7 v3 approach. (As an illustration of this phenomenon, consider this statement from [12] on Localized Information Models:

A localized model may be derived from a larger serializeable model. For instance, a serializeable model may be localized with constraints on datatypes or more refined concept domains than its parent. In this case, the localized model is a logical model and may be used as a serializeable model. It may also be derived by refinement and constraint of a portion of a domain analysis model and may not be serializeable, in which case it is still a conceptual model like its parent, or may be constrained in such a way from the domain information model that it can be serialized in which case it would be a logical model.)

The Behavioral Framework (BF). As we have seen, achieving interoperability on the service paradigm requires specification of the behaviour of computational units. But this requirement is not at all specific to healthcare. Rather, it is an overarching problem in computer science [2], and has given rise to a number of established frameworks, of which the prime examples, as already mentioned, are IEEE 1471 [15] and RM-ODP [17] (further examples are provided in Table 1). Given that, as we shall see, SAIF in any case uses many elements of RM-ODP, one has to question HL7's rationale for defining its own healthcare-specific hybrid rather than simply using the BF defined by RM-ODP.

The BF documentation [12] falls into two parts. First, sections 3.1 to 3.7 provide a framework for the specification of behaviour drawing on terms and definitions from RM-ODP and the ODP Enterprise Language [16]. The remaining sections of the document present ideas on BF implementation and on what are called 'correspondences', a term not clearly defined, but which seems to refer to links between the BF and HL7's now obsolete 'dynamic model' [11]. Terms like 'compound binding's contract correspondence' seem to have no precedent in the literature on computational behaviour and in the pertinent standards. Comparison of successive versions of the BF documentation makes it clear that a shift has been occurring in HL7 towards incorporating RM-ODP more directly, and that this has provided some incremental benefits. It is not clear what the corresponding sections add beyond RM-ODP, since no references are provided to any previous work or to efforts at empirical validation in existing implementations.

The Governance Framework (GF) has the aim of describing roles and responsibilities of the persons involved in implementing architectures conforming to interoperability specifications. We do not analyse the GF here as the analysis would go beyond the scope of this communication.

Finally, there is the SAIF's *Enterprise Conformance and Compliance Framework* (*ECCF*), whose most conspicuous feature is that it contains as part the SAIF architecture specification. System architecture, in other words, is addressed under the heading of conformance. In all existing interoperability frameworks known to us, it is architecture which subsumes conformance. To see what results from SAIF's approach, consider its treatment of traceability to requirements definitions, which SAIF sees as one component of conformance. A conformance and compliance framework as normally conceived defines a procedure whereby it is possible to follow a set of requirements from specification down to realisation in bits and bytes and thereby test for fulfillment. Conformance hereby presupposes architecture. Because neither the SAIF nor any other HL7v3 standard describes how requirements should be specified, HL7 is unable to specify how the content against which a trace is to be established should be constructed, and so there is no possibility of requirements traceability assessment.

Further problems derive from the core element of ECCF, the so-called Specification Stack, a matrix shown in Figure 1, which is a combination of RM-ODP and of OMG's model driven architecture (MDA) stack [21]. The ECCF claims that specifying artefacts (essentially specifications, code and test case documentation) in terms of how they instantiate the ECCF matrix provides a recipe for creating system architectures enabling semantic interoperability. Unfortunately, however, the ECCF matrix has a construction problem. Mathematically speaking, it is singular, which means that its column vectors are not linearly independent. This is because the viewpoints identified by RM-ODP (designated as 'dimensions' by the SAIF) already cover the full space of system engineering alternatives [17]. It therefore does not make sense to combine RM-ODP with MDA as if this would yield a matrix providing further coherent alternatives. Certainly the RM-ODP viewpoints can be combined with different weights to create different levels of abstraction. But this does not mean that RM-ODP and MDA can be put together to yield a cross-tabulation. The attempt to do so results in a set of instructions that cannot be followed and implemented. Since, if we leave the mentioned problems aside, the useful portions of the ECCF prove once again to be a re-packaging of definitions from the RM-ODP, one must question once more whether there is value that is being added.

3.2 Summary Evaluation of SAIF

We can now summarize our results by examining the degree to which the SAIF realises the framework principles outlined above.

General Framework Principles

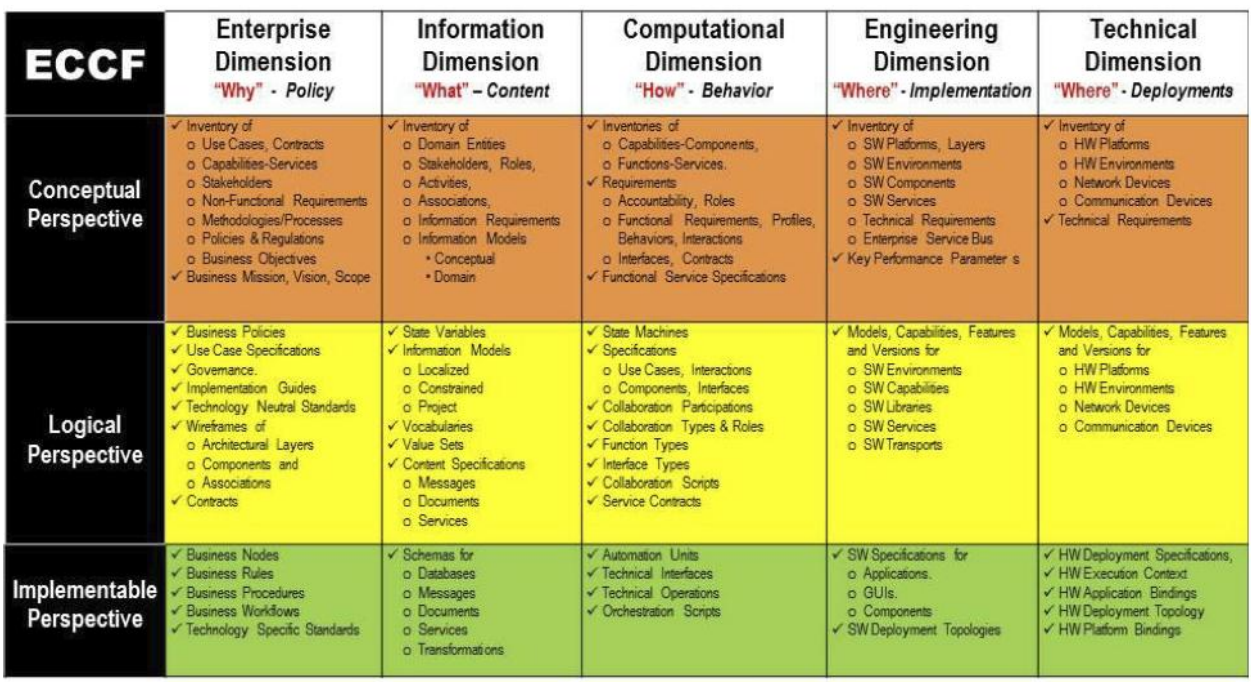
1. *Domain boundaries*. By defining in its preamble its scope as "Working Inter operability ... between business objects, components, capabilities, applications, systems and enterprises" [12] thus with no restriction to the health domain the SAIF indicates that it has arrogated to itself a domain that essentially covers the whole of IT.

2. *Knowledge reuse*. SAIF's definition of its own scope violates, too, the principle of knowledge reuse, raising the question of why HL7 should have included areas within its remit – such as the specification of computational behaviour – which have been already successfully covered by other efforts.

3. *Level of abstraction*. The RIM provides no clean separation of the logical modeling layer from the underlying implementation technology. The RIM in its current form is thus not technology independent – that is, it is not a logical artefact which can be expressed using multiple technologies – because it is permeated by HL7-specific XML design principles.

Level of Abstraction	Domain	Examples of Established Standards
Enterprise Level	Interactions and collaborations within and between enterprises and systems	OASIS Reference Architecture Foundation for Service Oriented Architecture [6], openGroup SOA Reference Architecture [34] (for overview see [19])
System Level	Orchestration (service interactions guided by a controlling master unit)	Business Process Execution Language, Business Process Modelling Notation [18, 35], Unified Modelling Language 2 activity diagrams [25]
	Choreography (service interactions not moderated by a controller but effected by computational units which are peers of each other)	Process algebra (one flavour of which is better known as π-calculus) [2, 22, 13], Occam [20], a process algebra-based language used in parallel computing.
Individual Service Level	Service interface, service behaviour	SOA-ML [24], Z notation [32]

Table 1. Examples of standards specifying guidelines for achieving distributed computational behaviour



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 ✓ Wireframes 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

Figure 1. ECCF specification stack [12]

Interoperability Framework Principles

1. *Enterprise requirements*. The principle to the effect that a SIF must describe how to determine and formalise requirements associated with pertinent business processes is not addressed by HL7, which has no process or formalism to define requirements [26]. This means that SAIF-conformant information models may be created without traceable link to formalised requirements, which would preclude the achievement of interoperability.
2. *Information model and ontology* is violated because, for the reasons indicated in the IF section above, the RIM is neither a useable information model nor a functional ontology.
3. *Computational model* asserts that a SIF must allow the specification of both the behaviour of and of the interactions between conformant systems. This principle is addressed by the SAIF's Behavioral Framework. As described above, the BF is neither comprehensive nor consistent. Overall, it would have been better to simply assemble a consistent selection of the results of relevant mainstream efforts and to indicate how each may be applied within a healthcare specific context, following a methodology already proven, for example, in the defense [3] and logistics [30] industries.
4. *Architecture framework and conformance model*, too, is not adequately addressed. As we saw, the normal relationship of architecture and conformance is inverted; such architectural guidelines as are provided are in consequence superficial. In addition, the core of the ECCF, the specification stack, fails to fulfill its role for the reasons given above. Overall, the ECCF reflects an interpretation and usage of RM-ODP that is not in accordance with the latter's ISO specification.

4 Conclusion and Recommendations

It has to be acknowledged that, with the SAIF initiative, HL7 is attempting to evolve its standards in the direction of contemporary interoperability paradigms. However, in light of the above analysis, we believe that the SAIF still has serious defects when measured in these terms. For SAIF to make a positive difference, we recommend that it be replaced by an approach that is based on a more fundamental reassessment of HL7 v3 that is in compliance with the interoperability design principles presented above. This does not require the development of new frameworks and methodologies: almost all of the needed components, including requirements formalisms (e.g. [5]), can be taken from existing standards and frameworks. Those components which genuinely need to be healthcare specific fall under the heading of what we have called 'information model and ontology'. To obtain useful results here, we recommend that HL7 adopt a tested upper-level ontology framework and an efficient, scientifically well founded, modular and composable domain-specific modeling language. We also recommend that HL7 replace its current approach to decision-making where it involves the use of balloted standards when addressing technical issues; when facing the sorts of complex challenges encountered in the realization of semantic interoperability, this is not an appropriate mechanism to constrain how engineers do their work. We recommend further that, like OMG and some other standards development organizations, HL7 adopts the requirement that working implementations should be provided before a standard can be published and that these implementations should be subject to a process of technical validation. Given its impressive expertise in the clinical domain we are confident that HL7 can produce real value; but to this end it should focus on what it can do best: specifying requirements describing those healthcare processes that need to be supported by Information Technology in the healthcare domain.

References

1. Attiya, H., Welch, J.: Distributed Computing: Fundamentals, Simulations, and Advanced Topics. Addison Wesley (2004)
2. Baeten, J.C.M., Basten, T., Reniers, M.A.: Process Algebra: Equational Theories of Communicating Processes, Cambridge Tracts in

Theoretical Computer Science, vol. 50. Cambridge University Press, Cambridge, UK (2010)
3. Bardo, B. (ed.): Proceedings of Fifth Annual Conference of the Systems Engineering for Autonomous Systems Defence Technology Centre. SEAS DTC (2010)
4. Bray, T., Paoli, J., Sperberg-McQueen, C. (eds.): Extensible Markup Language (XML) 1.0. W3C, http://www.w3.org/TR/1998/REC-xml-19980210 (1998)
5. Cockburn, A.: Writing Effective Use Cases. Addison Wesley (2003)
6. Estefan, J., Laskey, K., McCabe, F., Thornton, D. (eds.): Reference Architecture Foundation for Service Oriented Architecture. OASIS, http://docs.oasisopen.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf, 1st edn. (2009)
7. Goldberg, P.: Contracting disputes and software bugs plague NCI's 200m bioinformatics venture. The Cancer Letter 37(8) (2011)
8. Goldberg, P.: How NCI's plans for software giveaway sank in scientific and legal disputes. The Cancer Letter 37(9) (2011)
9. Goldberg, P.: Scrutiny of caBIG Exposes Conflicts, Bonanza for Contractors. The Cancer Letter 37(11) (2011)
10. HL7 (ed.): Clinical Document Architecture R2. HL7, http://www.hl7.org/implement/standards/cda.cfm (2005)
11. HL7 (ed.): HL7 Version 3 Standard: Core Principles and Properties of Version 3 Models, Release 1. HL7, www.hl7.org/ctl.cfm?action=ballots.home (2011)
12. HL7 Architecture Board (ed.): HL7 Service Aware Interoperability Framework: Canonical Version, Release 1 (Unique Ballot ID: SAIF CANON R1 I1 2011MAY). HL7, www.hl7.org/ctl.cfm?action=ballots.home (2011)
13. Hoare, C.: A model for communicating sequential processes. Prentice Hall (1985)
14. IEEE (ed.): IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries. IEEE (1991)
15. IEEE (ed.): IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-intensive Systems. ISO, http://www.isoarchitecture.org/ieee-1471/ (2000)
16. ISO/IEC (ed.): Information Technology – Open Distributed Processing – Reference Model – Enterprise Language. ITU-T X.911 ISO/IEC 15414 (1998)
17. ISO/IEC (ed.): Information technology Open Distributed Processing. ISO 10746, 1st edn. (1998)
18. Jordan, D., Evdemon, J. (eds.): Web Services Business Process Execution Language Version 2.0. OASIS, http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpelv2.0-OS.html (2007)
19. Kreger, H., Estefan, J. (eds.): Navigating the SOA Open Standards Landscape Around Architecture. Object Management Group, OpenGroup, OASIS, http://www.opengroup.org/onlinepubs/7699909399/toc.pdf (2009)
20. McEwan, A., Schneider, S., Ifill, W., Welch, P. (eds.): Communicating Process Architectures 2007 -WoTUG-30. IOS Press (2007)
21. Miller, J., Mukerji, J. (eds.): MDA Guide Version 1.0.1. OMG, 1.0.1 edn. (2003)
22. Milner, R.: Communication and Concurrency. Prentice Hall (1989)
23. Object Management Group (ed.): Unified Modeling Language. OMG, 1.0 edn. (1997)
24. Object Management Group (ed.): Service oriented architecture Modeling Language (SoaML). OMG, 1.0 edn. (2009)
25. Object Management Group (ed.): Unified Modeling Language, Superstructure. OMG, 2.2 edn. (2009)
26. Oemig, F., Blobel, B.: A communication standards ontology using basic formal ontologies. Stud Health Technol Inform. 156, 105–113 (2010)
27. openEHR (ed.): openEHR-Clinical mailing list archives. openEHR, http://www.openehr.org/mailarchives/openehr-clinical/msg00210.html (2005)
28. openEHR (ed.): openEHR and HL7v3. openEHR, http://www.openehr.org/206OE.html (2007)
29. Rector, A., Qamar, R., Marley, T.: Binding ontologies and coding systems to electronic health records and messages. Applied Ontology 4, 51–69 (2009)
30. Smirnov, A.V., Shilov, N.: Business network modelling: SOA-based approach and dynamic logistics case study. IJISMD 1(4), 77–91 (2010)
31. Smith, B., Ceusters, W.: HL7 RIM: An incoherent standard. In: Medical Informatics Europe 2006. pp. 133–138 (2006).
32. Spivey, J.: The Z Notation. A Reference Manual. Prentice-Hall, 2 edn. (1992)
33. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Record (ACM Special Interest Group on Management of Data) 31(4), 12–17 (2002)
34. The OpenGroup (ed.): SOA Reference Architecture. openGroup, http://www.opengroup.org/projects/soa-ref-arch/, 1st edn. (2009)
35. White, S., Miers, D.: BPMN Modeling and Reference Guide. Future Strategies Inc., Lighthouse Point, Florida (2008).