



Published in final edited form as:

J Biomed Inform. 2012 August ; 45(4): 658–666. doi:10.1016/j.jbi.2012.01.008.

Auditing Consistency and Usefulness of LOINC Use among Three Large Institutions - Using Version Spaces for Grouping LOINC Codes

M.C. Lin, MD¹, D.J. Vreeman, PT, DPT^{2,3}, and S.M. Huff, MD^{1,4}

¹The Department of Biomedical Informatics, The University of Utah, Salt Lake City, UT

²Regenstrief Institute, Inc., Indianapolis, IN

³Indiana University School of Medicine, Indianapolis, IN

⁴Intermountain Healthcare, Salt Lake City, UT

Abstract

Objectives—We wanted to develop a method for evaluating the consistency and usefulness of LOINC code use across different institutions, and to evaluate the degree of interoperability that can be attained when using LOINC codes for laboratory data exchange. Our specific goals were to: 1) Determine if any contradictory knowledge exists in LOINC. 2) Determine how many LOINC codes were used in a truly interoperable fashion between systems. 3) Provide suggestions for improving the semantic interoperability of LOINC.

Methods—We collected Extensional Definitions (EDs) of LOINC usage from three institutions. The version space approach was used to divide LOINC codes into small sets, which made auditing of LOINC use across the institutions feasible. We then compared pairings of LOINC codes from the three institutions for consistency and usefulness.

Results—The number of LOINC codes evaluated were 1,917, 1,267 and 1,693 as obtained from ARUP, Intermountain and Regenstrief respectively. There were 2,022, 2,030, and 2,301 version spaces among ARUP & Intermountain, Intermountain & Regenstrief and ARUP & Regenstrief respectively. Using the EDs as the gold standard, there were 104, 109 and 112 pairs containing contradictory knowledge and there were 1,165, 765 and 1,121 semantically interoperable pairs. The interoperable pairs were classified into three levels: 1) Level I – No loss of meaning, complete information was exchanged by identical codes. 2) Level II – No loss of meaning, but processing of data was needed to make the data completely comparable. 3) Level III – Some loss of meaning. For example, tests with a specific ‘method’ could be rolled-up with tests that were ‘methodless’.

Conclusions—There are variations in the way LOINC is used for data exchange that result in some data not being truly interoperable across different enterprises. To improve its semantic interoperability, we need to detect and correct any contradictory knowledge within LOINC and add computable relationships that can be used for making reliable inferences about the data. The LOINC committee should also provide detailed guidance on best practices for mapping from local codes to LOINC codes and for using LOINC codes in data exchange.

Keywords

Controlled Vocabulary; LOINC; Evaluation Research; Clinical Laboratory Information Systems; LOINC Usage; Consistency; Usefulness; Semantic interoperability

1. Introduction

Consistency and usefulness are two important characteristics of good terminological systems (TSs), especially for information exchange. *As we use the terms in this article, consistency* means that between any two terms within TSs there is no contradictory knowledge (as represented by the implicit or explicit relationships between concepts), and *usefulness* means that there is knowledge in the terminology that allows for creation of an efficient algorithm for making inferences using the relationships in the terminology for supporting different kinds of use, e.g. information retrieval, data integration or clinical decision support [4]. Auditing TSs can be a difficult task because of the huge number of concepts, e.g. LOINC has more than 65,000 codes. In order to reduce this task to a manageable size, researchers have used semantic methods to search for similar concepts in the UMLS [9] or used semantic structures to partition SNOMED into smaller groups [28]. Previous reports have shown that most inconsistencies in LOINC mapping result from choosing codes that vary in the ‘method’, ‘scale’ and ‘property’ characteristics of the codes. [3, 20, 26]. The use of Version Spaces is a common technique used in machine learning for concept discovery [24]. Version Spaces are used to divide all hypotheses into smaller subspaces to make it possible to search similar concepts by a given set of constraints. This paper describes a systematic method for auditing the consistency and usefulness of LOINC use and discusses potential strategies to approach best practices in the use of LOINC for interoperable data exchange.

1.1 Auditing TSs on policy vs. use

Early papers on TSs development were focused on functional, structural and policy perspectives. These papers include Cimino’s desiderata for creating controlled medical vocabularies [10], Chute et al.’s study of functional characteristics of comprehensive health terminology systems in the United States [8], and the technical specification published by International Standard Organization (ISO) - “Health informatics – Controlled health terminology – Structure and high-level indicators” [16]. As TS usage increased, discussions shifted to descriptions of practical use. These studies included analyzing coverage of the UMLS for coding of concepts in the Gene Ontology (GO) [6], comparing coding consistency of SNOMED CT among three commercial coding companies [2] and evaluating the performance of LOINC when comparing laboratory data among three hospitals [3]. To summarize all auditing methods for TSs, Zhu et al. have done a thorough literature review on different auditing methods, including manual, systematic and heuristic methods [29].

1.2 The development of LOINC

1.2.1 Rapid evolution of LOINC model—LOINC provides a universal terminology for reporting laboratory tests and other clinical observations. Since 1994, LOINC has grown from about 6,000 codes to more than 65,000 in the current version. As Cimino noted in his desiderata [10], an important characteristic of TSs is to “Evolve Gracefully”, and LOINC tries to adhere to this principle [15]. The LOINC committee has emphasized practical experience in using LOINC to improve its design. Whenever the original design of LOINC is not sufficient, the design is enhanced or a new model is created. Before migrating to the current six-axis model, at least four different earlier models were created (Table 1). For example, the first design of LOINC was a four-axis model, but with more implementation,

the original model was insufficient for specifying some tests, e.g. it lacked the ability to specify timing (24 hours, 12 hours, or 4 hours, etc.). Therefore, “<timing>” was added to create a new model.

1.2.2 LOINC in Action—Many places adopted LOINC in their daily operations, including large commercial laboratories, hospitals, health care provider networks, insurance companies, and public health departments [21]. Recently, LOINC was adopted as the terminology standard for certification of laboratory orders and results, including electronic reporting of lab results to public health agencies as part of the Centers for Medicare and Medicaid Services (CMS) Electronic Health Record (EHR) “Meaningful Use” incentive program. LOINC was also used in a German Hospital Information System (HIS) to identify the document type of reports sent as Clinical Document Architecture (CDA) documents [14] and to retrieve laboratory data of adverse events automatically from clinical trial databases [7]. LOINC is also frequently used in computerized clinical decision support systems [1]. Although the scope of LOINC covers both clinical and laboratory observations, for the purpose of this paper we focus exclusively on laboratory content.

1.2.3 Evaluations of LOINC—Evaluating LOINC performance in actual practice can help to improve LOINC design. McDonald et al. summarized LOINC development and worldwide use [21]. Lau et al. reported LOINC coverage for the laboratory test dictionary in the US Department of Defense (DoD) [18] and Vreeman et al. reported LOINC coverage for tests in the Indiana Network for Patient Care (INPC) [27]. We also conducted a series of studies about LOINC usage among three large institutions. First, we reported that LOINC codes can cover more than 99% of the volume of every day laboratory tests among two institutions and 79% of tests in a reference laboratory [19]. Second, we evaluated the correctness of LOINC mapping and reported that there were 0.45% (4/884) tests mapped to totally unrelated LOINC codes and 4% (36/884) tests containing at least one error in mapping to the 6 axis model of LOINC [20]. An earlier study by Baorto et al. also evaluated LOINC performance when combining laboratory results associated with congestive heart failure patients among three teaching hospitals [3].

1.2.4 Requirements for ideal LOINC use—According to Devanbu et al.’s definition of a good knowledge system [13], the best practice of LOINC should have the following characteristics: 1) **Completeness**: it should have all the necessary LOINC codes to cover the domain of interest, 2) **Correctness**: mapped LOINC codes should be faithful to the original meaning of the tests, 3) **Consistency**: the knowledge implied by different LOINC codes should be consistent, e.g. if two different codes have identical meanings, the codes are duplicates and the consistency principle is violated, and 4) **Competence**: Usefulness is the fundamental goal of LOINC for supporting use of laboratory data in different fields. Support for the use of ontologic relationships is one of the important competencies of TSs [25]. LOINC should define the relations between codes and combinations of codes that allow users to infer equivalence, if their meanings in data instance representation are interoperable. That is, if the combination of two codes has the same meaning as a single code (a difference in the use of pre- or post-coordination), relationships should exist between the codes that support the assertion of equivalence. Previous evaluations have described LOINC with respect to the first two characteristics, completeness [18,19] and correctness [20]. The focus of this paper is on the evaluation of consistency and usefulness.

1.3 Definition of Consistency and Usefulness of TSs

1.3.1 Consistency—Consistency in a system implies that the system does not contain contradictory knowledge. Consistency of TSs could be discussed from two perspectives: 1) **Internal consistency**: Inconsistency can result if there is a failure to uniformly employ

design principles throughout the entire terminology. For example, the hierarchies, semantic rules, mapping policies, etc., can be incomplete or inconsistent. One study investigated inconsistencies in the usage of the ‘parent-child’ and ‘is-a relationship’ in the Unified Medical Language System (UMLS) [11]. Another study investigated inconsistent use of semantic or linguistic rules for representing similar terms for SNOMED [5]. They found that two common modifier terms, “acquired” and “congenital”, used for the disease “porphyria”, are sibling relationships; but for another disease, “acquired keratoderma (D0-22310)” and “congenital keratoderma (D4-40130)” were not sibling relationships but were found in two separate branches in SNOMED [5]. 2) **External consistency:** The terminology allows for inconsistent and non-interoperable representation of instance data. Baorto et al. reported that the same laboratory tests could be mapped to different LOINC codes, e.g. different coding strategies could be used to choose specimen types, serum, plasma or serum/plasma for the same tests across different institutions [3]. A study comparing three versions of SNOMED coding of the same case report forms (CRF) done by three commercial coding companies showed that there was no significant degree of inter-rater agreement in their coding behaviors, because the three coding companies used different coding strategies concerning pre- vs. post- coordination [2]. Sometimes TSs designs were workable (internal consistency), but TSs were not used consistently across institutions (external consistency). Therefore, ensuring that TSs are both internally and externally consistent is crucial for semantic interoperability.

1.3.2 Usefulness—The use of ontologic relationships to support biomedical inferencing, as exemplified in knowledge management, data integration and decision support, is one of the important characteristics of TSs [4]. Often times the same instance data can be represented by different combinations of terms, and an ontology can provide an ability for finding equivalence of those terms. Steindel et al. reported the ability of building the hierarchy of LOINC codes can facilitate public health reporting [26]. To allow consistent reporting of laboratory data to the Centers for Disease Control (CDC) using LOINC, a “Reportable Condition to LOINC mapping table for National Notifiable Disease (NND)” was created for use by all healthcare institutions [22]. However, new LOINC codes for screening tests for NND were continually created and manually adding them into the above mapping table would be labor intensive. To facilitate the update process, a set of rules was developed to allow new LOINC codes to be added to the table according to their similarity in meaning to existing LOINC codes. For example, a methodless measurement was added as a parent for a measurement with a method ‘automatically’, all different methods for the same analyte would be placed as children under the same analyte, and ordinal (ORD) measurements were considered to be children of quantitative (QN) measurements [26]. Therefore, new LOINC codes could be added “automatically” to the mapping table based on these rules.

1.4 How to audit TSs efficiently

TSs usually consist of many thousands of terms. It is not an easy task to audit TSs, because scanning all pairs of terms and examining their relationships would create huge numbers of combinations. A key strategy in developing an efficient approach for comparing terms is to generate only the most likely pairs, instead of scanning all pairs. In general, it is not very meaningful to compare two things which are not related, e.g. comparing a bacterial screening test to a sodium measurement is less interesting than comparing two different bacterial screening tests. There are two common approaches for searching similar things to generate pairs: **1) Searching equivalent concepts:** When examining one term, only search for other terms which have similar meaning. Cimino used semantic methods to search for terms having similar meaning to audit the UMLS, e.g. using synonyms and semantic types

of concepts. For example, he neglected different *word order* in searching for equivalent concepts, and he found duplicate concepts as shown below [9].

C0000760: ABNORMAL PAP SMEAR

C0240660: PAP SMEAR ABNORMAL

Cornet et al. used a similar approach to detect duplicates in DICE TS and they successfully found 4 duplicate concepts in a set of 2500 concepts [12].

2) Segmentation: This approach divides TSs into several smaller groups based on their characteristics. It has been applied to evaluating the National Cancer Institute Thesaurus (NCIT) and SNOMED by Perl's group and their colleagues [23, 28]. Their method has two main phases for auditing terminologies: the automated preparation phase and 2) the manually guided-discovery phase. The first phase is composed of four steps: 1) divide all concepts into several groups called an area-taxonomy based on concepts having the same roles, 2) construct a compact abstraction network, 3) refine each division into groups of concepts, called P-area's and 4) finally, construct an enhanced abstraction network, called the P-area taxonomy [23]. Basically they segmented NCIT's Biological Process hierarchy, by using a general to specific (divide and conquer) approach using roles (e.g. biological process role, chromosomal location role) to place terms into groups (areas) for constructing the area-taxonomy. The terms within each area could be categorized into partial areas by their semantics as captured by the root role of the partial area. Using this approach, a partial area taxonomy could be created that segmented NCIT contents based on structure and semantics for support of manual auditing, by identifying partial areas with a high likelihood of errors [28].

1.4.1 Concept Learning—Concept learning is a common approach in Machine Learning where searching for similar concepts is based on a given set of attributes [24]. LOINC utilizes six attributes to specify the meaning of laboratory tests, which is very similar to how a specific hypothesis is defined in concept learning. One approach to concept learning is to divide all instances into several *version spaces* based on a given set of constraints [24]. Using this approach, all instances within a given version space will share similar attributes.

1.4.2 Notation—In concept learning, all hypotheses can be represented by a set of attributes. LOINC term names consists of a vector of 6 constraints (Analyte, Method, Time, Scale, Property, System). For each constraint, the attribute could be entered using the following tokens [24]:

- “?”, allow any value
- allow a specific value (e.g., Hematocrit,)
- ‘ \emptyset ’, empty set, do not allow any value

Therefore, the most general hypothesis, which includes all LOINC codes, could be represented by the expression

$\langle ?, ?, ?, ?, ?, ? \rangle$

And, the most specific concepts (doesn't allow any LOINC codes), is represented by

$\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$

We can specify a group of LOINC codes for measuring Sodium from Blood specimens, no matter what their methods, time, scale, or property by this expression:

$\langle \text{Sodium}, ?, ?, ?, ?, \text{Blood} \rangle$

Using the same methodology, the following expression represents a group having the same analyte.

<Analyte,?,?,?,?,?>

The ‘Analyte’ could be substituted by any more specific analyte. If we are counting the number of hypotheses <Analyte,?,?,?,?,?> in a dataset, it equals the count of instances that represent analytes. This expression is also equivalent to the structured query language (SQL) statement ‘Group by Analyte’. Similarly, the following expression is used to specify the group having the same analyte and system. The analyte and system could each be substituted by any analyte or system. The number of hypotheses that exist is equal to the number of combinations of analyte and system. It is also equivalent to the SQL statement ‘Group by Analyte and System’.

<Analyte,?,?,?,?, System>

1.5 Proposed framework

We wanted to investigate the consistency and usefulness of LOINC concepts by creating a version space. By examining all the relationships between any two LOINC codes within each version space, we wanted to:

1. Determine whether there is any contradictory knowledge, e.g. duplicate codes, or code ambiguity
2. Detect any combinations of pre and post coordinated tests that are equivalent or where relationships can be created that will enable systems to interoperate
3. Provide suggestions for best practices for LOINC users that will improve the semantic interoperability of LOINC

2 Methods

2.1 Collecting Extensional Definitions (EDs) of LOINC from three large institutions

We first sent out invitations to several major institutions, and three institutions agreed to provide their laboratory data for the experiment. These three institutions were: 1. Associated Regional and University Pathologists, ARUP Laboratories (Salt Lake City, UT) 2. Intermountain Healthcare, Intermountain (Salt Lake City, UT) 3. Regenstrief Institute, Inc. (Indianapolis, IN). With IRB approval, de-identified patient data for general laboratory tests for the year of 2007 for each institution were selected for this study. In our previous studies [19, 20], we used all data collected from the five institutions that contribute data to Regenstrief. To avoid selection bias, only the institution with the highest volume of tests was included in this study. We developed a parsing program written in JAVA and Python to process patient data to generate EDs, which include local code, local description, numeric value, units of measure, coded variables and LOINC mappings, from each institution shown as Table 2. Then, we distributed installed parsing programs to each institution and asked collaborators to process the de-identified patient data within the virtual machines. Only processed EDs were sent back to us for analysis.

Table 3 shows how EDs can be used to determine appropriate LOINC naming. For example, there were EDs of two genetic tests retrieved from two different institutions which had identical local names, ‘ALPHA-1-ANTITRYPSIN PHENOTYPE’. The local codes were mapped to two different LOINC codes that had different ‘method’, ‘property’ and ‘scale’ attributes. Considering only their local names, there was not enough information to determine whether two LOINC codes with different ‘property/scale’ were needed. But if we examined the values reported for each test as summarized from their EDs, we can determine

that these are similar tests. This comparison of EDs can help us determine whether LOINC naming for specific tests was accurate.

2.2 Creating Version Spaces for searching similar LOINC concepts (Find-S: Finding a maximally specific hypothesis)

As noted previously, making version spaces is an approach used in concept learning to create subspaces of hypotheses using different constraints. One approach to creating version spaces is Find-S: Finding a maximally specific hypothesis, which creates hypotheses from the most specific to least stringent by relaxing constraints one by one. For example, the most specific hypothesis for specifying a unique LOINC concept is:

<Analyte, Method, Time, Scale, Property, System >

A less stringent example of a hypothesis can be made by relaxing one constraint (property):

<Analyte, Method, Time, Scale,?, System >

It has been reported that choosing different ‘properties’ was the most frequent reason that different coding choices were made in a study about LOINC coding behaviors in congestive heart failure patients [3]. In our previous study about correctness of LOINC mapping, choosing different ‘Method’, ‘Scale’ and ‘Property’ attributes was the most common reason for different coding choices among three large institutions [20]. For example, in the ‘Method’ axis, some institutions usually use a code that specifies the method (when available), whereas other institutions always choose terms that are “methodless”. Another example, is that in the ‘Scale/Property’ axes, LOINC uses two distinct styles (Prid:Nar VS. Prid:Nom) for reporting the interpretation of laboratory tests (e.g. CFTR gene mutation analysis). The Narrative (Nar) scale is for free text results (sentences, paragraphs, sections), whereas the Nominal (Nom) scale is used for representing coded values, as when selecting an organism found on culture from a coded list of bacteria. The differences between these types are often subtle and require understanding the reporting system. Steindel et al. also concluded that for some purposes, such as finding any code that could be used to indicate the presence of a particular disease, rolling up LOINC codes and ignoring some LOINC axes (e.g. method, scale, or property) can be beneficial [26]. For example, the following LOINC codes have differences in method, scale or property, but they all can be used to diagnose the infectious disease ‘BACILLUS ANTHRACIS’:

BACILLUS ANTHRACIS AB	EIA	PT	ORD	ACNC	SER
BACILLUS ANTHRACIS AB	CF	PT	ORD	ACNC	SER
BACILLUS ANTHRACIS AB	ID	PT	QN	TITR	SER

Using the version space constraint notation described above, all three of these concepts can be represented by the following expression:

<BACILLUS ANTHRACIS AB,?, PT,?,?, SER>

Based on the most common kinds of mapping errors, we decided to choose <Analyte,?, Time,?,?, System> as the optimal design for auditing LOINC.

2.3 Constructing the Version Space for the <Analyte,?, Time,?,?, System> expression

After receiving the data from each institution, all EDs were loaded into the database. We then grouped all LOINC codes into smaller subspaces by creating Version Spaces matching the expression <Analyte,?, Time,?,?, System> shown as Table 4. Within each Version Space, all LOINC codes shared the identical three axes “Analyte”, “System” and “Time”, while having different values for ‘Scale’, ‘Property’ and ‘Method’. LOINC flags codes that can be used in post-coordinated expressions by using ‘XXX’ as a value in certain axes,

especially “System” and ‘Time’. The ‘XXX’ signifies that the information content of that axis is communicated elsewhere in an instance of data, for example in a different field in an HL7 message. In creating our Version Spaces, we interpreted ‘XXX’ to be any value. For example, we added the code <Leukocytes, Automated count, Pt, NCnc, Qn, XXX> into the Version Space <Leukocytes,?, Pt,?,?, Bld > and <Leukocytes,?, Pt, ?,?, CSF >. We also processed those LOINC codes having ‘XXX’ in the ‘Time’ axis using the same approach. In the real world, the meaning of ‘XXX’ in a term is not literally anything; its meaning is constrained by the nature of the ‘Analyte’. Since we used existing concepts from LOINC with ‘XXX’ the broad interpretation of XXX is not a significant problem.

2.4 Semi-automated Review

We developed a semi-automated review process, which could be used to systematically discover the relationships of terms and determine the characteristic of the relationships, such as consistency and usefulness. This process consisted of three phases: 1) Discovery phase: A python program was written to discover all patterns of similar pairs in each version space for manual review, 2) Discussion phase: All similar patterns were manually reviewed and discussed by experts, and 3) Analysis phase: We analyzed all patterns to define formal descriptions (a taxonomy) for them, e.g. (Method vs. Methodless) or (Pre vs. Post-coordinated).

2.4.1 Discovery phase—A python program was developed to scan all LOINC pairs from each version space. For example, in the Table 4, version space <Hemoglobin,?, Pt,?,?, Bld>, between ARUP and Intermountain, there were four pairs, <5,1><5,7><6,1><6,7> chosen for review.

One important principle was that we focused on general issues for analytes having similar patterns. For example, the following patterns, <Imp:Nar> and <Imp:Nom> were considered contradictory designs (ambiguous), because LOINC codes with the same analyte but using these two different patterns were found to report similar things when their full EDs were considered. All analytes having these designs were found to be contradictory.

<13514-5: Hemoglobin pattern: Electrophoresis: Pt: **Imp: Nar:** Bld>

<12710-0: Hemoglobin pattern: Electrophoresis: Pt: **Imp: Nom:** Bld>

or

<49291-8: Prophyryns: None: Pt: **Imp: Nar:** Urine>

<44014-9: Prophyryns: None: Pt: **Imp: Nom:** Urine >

Thus, the patterns, <Imp:Nar> and <Imp:Nom>, were flagged as ‘contradictory’, and automatic checking for these patterns became part of the logic in the review program. Similarly when two patterns (Method:Scale:Property) are semantically interoperable, all LOINC codes having these two patterns are considered semantically interoperable. In the following two examples, one code has a specified method and the other has not. Their meanings are not exactly the same, but they can support a degree of semantic interoperability.

<14336-2: Ethanol: **GC:** Pt: MCnc: Qn: Ser/Plas>

<5643-2: Ethanol: **Null:** Pt: MCnc: Qn: Ser/Plas>

or

<20405-7:Urobilinogen: **Test strip:** Pt: MCnc: Qn: Urine>

<3107-0: Urobilinogen: **Null:** Pt: MCnc: Qn: Urine>

The program discovered all similar pairs and generated a text report for manual review.

2.4.2 Discussion phase—After the discovery phase, there were two stages of discussion. In the first stage, small groups of experts reviewed the overall findings. Any confusing findings involving LOINC policy were distributed in the LOINC committee for additional discussion. The discussion phase can increase awareness of consistency and usefulness issues in the general public.

2.4.3 Analysis phase—After the discussion phase, we found some patterns that shared similar characteristics e.g. (Method vs. Methodless) or (Pre vs. Post-coordinated). Therefore, we developed formal descriptions (a taxonomy) for each group. Using those formal descriptions we can communicate consistency and usefulness of LOINC more efficiently.

3. Results

Table 5 shows the number of laboratory tests collected from three institutions and the number of version spaces for each institution. Table 6 shows the distributions of the size of pairwise version spaces between institutions. It also shows that many LOINC codes were used by all institutions and most version spaces contain less than 10 items. After reviewing consistency within each version space, reasons causing contradictory knowledge were classified into three categories (table 7): 1) Deprecated codes: LOINC codes were already deprecated in the LOINC distribution file but were still actively being used in laboratory systems. 2) Raw measurement versus interpretation: Both raw measurements (30 ng/ml) and their interpretations (e.g. Negative) are usually reported in laboratory systems. The LOINC committee created terms for both raw measurements and interpretations but most institutions only choose one code for mapping both kinds of results. 3) Ambiguous codes: There are LOINC codes for reporting similar things, but the institutions have chosen different styles of pre and post coordination.

Table 8 shows examples for each category. The ‘Raw measurement versus interpretation’ category is the most frequent reason for conflicting code use. Table 9 reveals results after the review for usefulness. All semantically interoperable pairs could be classified into 3 levels: 1) Level I: There is no meaning loss. Information is fully exchanged by identical codes, 2) Level II - there is no meaning loss, and information can be fully exchanged, but it requires conversion of units of measure or other kinds of additional processing. Level II interoperability has been further classified into another three sub-categories: II.a is ‘MCnc vs. SCnc’, II.b is ‘Pre vs. Post-coordinated’ and II.c is ‘value vs. log value’, and 3) Level III - there is some meaning loss and only partial information could be exchanged. We have listed examples and further explanation for each category in Table 10. From Table 9, we also learn that there are 956, 559 and 862 laboratory tests where results are exchanged using identical LOINC codes between ARUP & Intermountain, Intermountain & Regenstrief and ARUP & Regenstrief respectively.

4. Discussion

Using LOINC, or any standard coding system to support interoperable data exchange is not an easy task. We evaluated LOINC specifically on consistency and usefulness of use perspectives which revealed several important findings.

4.1. Creating LOINC codes should be consistent

LOINC is still being actively developed. Creation of new codes and updates could cause inconsistency. Ideally when new codes are created, the new code should follow previous design patterns. Currently, there is no way to automatically check consistency except by

using expert guidance from the LOINC committee. For long-term maintenance, there should be a standardized method to detect any contradictory knowledge within LOINC. When inconsistencies in LOINC codes are discovered, it leads to the complicated task of correcting the error. This leads to several suggestions about best practices: First, the LOINC committee should decide which codes should be deprecated or when a new code should be created for replacing existing contradictory codes. Second, after changes have been made to the LOINC codes, each institution should update any deprecated codes. Although the status for each LOINC code is clearly specified in the LOINC distribution and the RELMA program includes functions for finding local codes mapped to deprecated LOINC codes, we still observed deprecated codes in use in operational databases. This finding suggests that LOINC users are not always diligent in updating their systems with new LOINC releases.

4.2. Minimalism as a better approach

Complicated designs are hard for users to follow. The LOINC committee designed two styles (raw measurements versus interpretations) of codes for storing raw measurements and interpretations as distinct observations, which is a workable design. We have observed that sometimes when a new kind of test comes onto the market it is initially reported with interpretive codes, but over time the prevalent reporting pattern changes to sending the quantitative results. Therefore, users who aggregate longitudinal data or interface with many partners need extra efforts to accommodate both LOINC styles. Usually laboratories just choose one style for reporting data. Sometimes even within the same institution, we observed different styles due to different mappers. After reporting these issues to the LOINC committee, the LOINC committee suggested that these two different styles be condensed into one style that reports both pieces of information in a single data instance. For reporting the interpretation of raw measurements, the LOINC committee adopted the Value-Cutoffs-Interpretation style recommended by HL7. This style allows sending the value, cutoffs and interpretation in the same result segment of an HL7 message. This style indicates that results such as “POS” and “NEG” should go in the OBX-8 field for normalcy status in an HL7 message. Thus, LOINC codes with a scale of Qn can be appropriately used in cases where the “values” coming back are coded interpretations of the true numeric result value.

Another common issue we observed was differences in mapping to method-specific or methodless LOINC codes. In current laboratory systems, information about the ‘method’ is not always specified. For most practical purposes, users only care about the ‘method’ for billing purposes and only care about meaning of tests for analytic purposes. Using ‘methodless’ LOINC codes with the Value-Method-Interpretation style is one possible way to solve this problem.

The minimalist style of only keeping necessary information in the LOINC codes, can avoid knowledge overloading and overlap for LOINC users, and also avoids the situation where users choose different coding styles.

4.3. Measuring semantic interoperability of LOINC between two institutions

To determine whether LOINC has achieved its goal of improving semantic interoperability of laboratory tests is not an easy task. Any inconsistency in code definitions or use reduces semantic interoperability. If we can use consistent approaches then as more laboratory tests are exchanged we can achieve better semantic interoperability. By classifying semantic interoperability into three well-defined levels, we can better measure how we are doing in using data to its greatest potential. Overall, Level I statistics can be used to measure the semantic interoperability that can be achieved between two institutions by the use of current LOINC codes. It should also be possible to automate data conversions so that Level II interoperability is achieved in working systems. Using the definitions for the different levels

of interoperability enables the measurement of improvements in data representation as we improve LOINC code consistency and as systems implement better mapping strategies.

4.4. Using Version Space approach and EDs for auditing TSs

We demonstrated that using the version space approach can reduce the complexity of auditing TSs. It consists of three important steps: 1) Prepare a set of attributes for concept learning. The crucial step in concept learning is finding a set of attributes that can be used to partition the concepts. Perl and his colleagues used the “divide and conquer method” for constructing the abstraction network (area-taxonomy and P-area taxonomy) according to the roles of concepts, e.g. [has associated location], which is similar to utilizing attributes for representing concepts. Formal concept analysis (FCA) is another similar approach to concept learning, which decomposes compound medical concepts into several atomic concepts by constructing a concept-attribute table, e.g. tonsillitis could be represented by tonsils and inflammation process [17]. Both Perl’s approach and FCA could be used to create hierarchy of TSs. All three approaches, concept learning, Perl’s taxonomy analysis and FCA, provide methods for measuring the similarities between two concepts, which can be used for grouping or dividing concepts. 2) Grouping Terms by creating version spaces: There are general to specific and specific to general approaches for creating version spaces. For those TSs that already had a set of attributes (e.g. description logics), the above two approaches could be used. Our specific to general approach had one advantage in that it created smaller groups at the beginning compared to the “divide and conquer method”, which requires multiple steps to generate smaller groups. One limitation of this study was that we only audited the consistency and usefulness within single analyte version spaces. One possible extension of this study would be to use a semantic method for searching subspecies of analytes for creating version spaces, e.g. <Genetic tests,?, Time,?,?, System> or <Antibody,?, Time,?,?, System> [9]. Therefore, consistency and usefulness across a wider range of terms could be evaluated. 3) Using EDs for auditing: Our results revealed that using EDs alone for auditing may not be practical in the real world. Using EDs and version spaces together can provide a better understanding of the quality of TS implementations.

5. Conclusions

Min et al. concluded that auditing should be part of the terminology design life cycle, and LOINC is no exception [23]. There are variations in the way LOINC is used for data exchange that result in some data not being truly interoperable across different enterprises. To improve its semantic interoperability, we need to detect and correct any contradictory knowledge within LOINC using audit techniques and add computable relationships that can be used for making reliable inferences about LOINC encoded data. The LOINC committee should also provide detailed guidance on best practices for mapping from local codes to LOINC codes and for using LOINC codes in data exchange.

Acknowledgments

The authors would like to thank Brian Jackson, MD and Alan Terry, MS from ARUP for retrieving and processing the ARUP data set.

References

1. Ahmadian L, van Engen-Verheul M, Bakhshi-Raiez F, Peek N, Cornet R, de Keizer NF. The role of standardized data and terminological systems in computerized clinical decision support systems: literature review and survey. *International journal of medical informatics*. 2011; 80:81–93. [PubMed: 21168360]

2. Andrews JE, Richesson RL, Krischer J. Variation of SNOMED CT coding of clinical research concepts among coding experts. *J Am Med Inform Assoc.* 2007; 14:497–506. [PubMed: 17460128]
3. Baorto DM, Cimino JJ, Parvin CA, Kahn MG. Combining laboratory data sets from multiple institutions using the logical observation identifier names and codes (LOINC). *Int J Med Inform.* 1998; 51:29–37. [PubMed: 9749897]
4. Bodenreider O. Biomedical ontologies in action: role in knowledge management, data integration and decision support. *Yearb Med Inform.* 2008:67–79. [PubMed: 18660879]
5. Bodenreider O, Burgun A, Rindfleisch TC. Assessing the consistency of a biomedical terminology through lexical knowledge. *International journal of medical informatics.* 2002; 67:85–95. [PubMed: 12460634]
6. Bodenreider O, Mitchell JA, McCray AT. Evaluation of the UMLS as a terminology and knowledge resource for biomedical informatics. *Proc AMIA Symp.* 2002:61–65. [PubMed: 12463787]
7. Brandt, CA.; Lu, CC.; Nadkarni, PM. Automating identification of adverse events related to abnormal lab results using standard vocabularies. *AMIA ... Annual Symposium proceedings/AMIA Symposium.* AMIA Symposium; 2005. p. 903
8. Chute CG, Cohn SP, Campbell JR. A framework for comprehensive health terminology systems in the United States: development guidelines, criteria for selection, and public policy implications. ANSI Healthcare Informatics Standards Board Vocabulary Working Group and the Computer-Based Patient Records Institute Working Group on Codes and Structures. *Journal of the American Medical Informatics Association : JAMIA.* 1998; 5:503–510. [PubMed: 9824798]
9. Cimino JJ. Auditing the Unified Medical Language System with semantic methods. *Journal of the American Medical Informatics Association : JAMIA.* 1998; 5:41–51. [PubMed: 9452984]
10. Cimino JJ. Desiderata for controlled medical vocabularies in the twenty-first century. *Methods Inf Med.* 1998; 37:394–403. [PubMed: 9865037]
11. Cimino JJ, Min H, Perl Y. Consistency across the hierarchies of the UMLS Semantic Network and Metathesaurus. *Journal of Biomedical Informatics.* 2003; 36:450–461. [PubMed: 14759818]
12. Cornet R, Abu-Hanna A. Auditing description-logic-based medical terminological systems by detecting equivalent concept definitions. *International journal of medical informatics.* 2008; 77:336–345. [PubMed: 17693132]
13. Devanbu, PT.; Jones, MA. The use of description logics in KBSE systems: experience report. *IEEE Computer Society Press;* 1994. p. 23-35.
14. Dugas M, Thun S, Frankewitsch T, Heitmann KU. LOINC codes for hospital information systems documents: a case study. *J Am Med Inform Assoc.* 2009; 16:400–403. [PubMed: 19261942]
15. Huff SM, Rocha RA, McDonald CJ, De Moor GJ, Fiers T, Bidgood WD Jr, Forrey AW, Francis WG, Tracy WR, Leavelle D, Stalling F, Griffin B, Maloney P, Leland D, Charles L, Hutchins K, Baenziger J. Development of the Logical Observation Identifier Names and Codes (LOINC) vocabulary. *J Am Med Inform Assoc.* 1998; 5:276–292. [PubMed: 9609498]
16. ISO/TC215. Health informatics --Controlled health terminology --Structure and high-level indicators, Report NO.:17117. 2002.
17. Jiang G, Ogasawara K, Endoh A, Sakurai T. Context-based ontology building support in clinical domains using formal concept analysis. *International journal of medical informatics.* 2003; 71:71–81. [PubMed: 12909160]
18. Lau LM, Banning PD, Monson K, Knight E, Wilson PS, Shakib SC. Mapping Department of Defense laboratory results to Logical Observation Identifiers Names and Codes (LOINC). *AMIA Annu Symp Proc.* 2005:430–434. [PubMed: 16779076]
19. Lin MC, Vreeman DJ, McDonald CJ, Huff SM. A Characterization of Local LOINC Mapping for Laboratory Tests in Three Large Institutions. *Methods Inf Med.* :49.
20. Lin, MC.; Vreeman, DJ.; McDonald, CJ.; Huff, SM. Correctness of Voluntary LOINC Mapping for Laboratory Tests in Three Large Institutions. *AMIA ... Annual Symposium proceedings/AMIA Symposium.* AMIA Symposium; 2010; 2010. p. 447-451.
21. McDonald CJ, Huff SM, Suico JG, Hill G, Leavelle D, Aller R, Forrey A, Mercer K, DeMoor G, Hook J, Williams W, Case J, Maloney P. LOINC, a universal standard for identifying laboratory observations: a 5-year update. *Clin Chem.* 2003; 49:624–633. [PubMed: 12651816]

22. McDonald CJ, Overhage JM, Dexter P, Takesue BY, Dwyer DM. A framework for capturing clinical data sets from computerized sources. *Annals of internal medicine*. 1997; 127:675–682. [PubMed: 9382379]
23. Min H, Perl Y, Chen Y, Halper M, Geller J, Wang Y. Auditing as part of the terminology design life cycle. *Journal of the American Medical Informatics Association : JAMIA*. 2006; 13:676–690. [PubMed: 16929044]
24. Mitchell, TM.; Michell, T. *Machine learning*, McGraw-hill series in computer science. WCB McGraw-Hill; 1997.
25. Rubin DL, Shah NH, Noy NF. Biomedical ontologies: a functional perspective. *Briefings in bioinformatics*. 2008; 9:75–90. [PubMed: 18077472]
26. Steindel, S.; Loonsk, J.; Sim, A.; Doyle, T.; Chapman, R.; Groseclose, S. Introduction of a hierarchy to LOINC to facilitate public health reporting. *American Medical Informatics Association*; 2002. p. 737
27. Vreeman DJ, Stark M, Tomaszefski GL, Phillips DR, Dexter PR. Embracing change in a health information exchange. *AMIA Annu Symp Proc*. 2008:768–772. [PubMed: 18999242]
28. Wang Y, Halper M, Min H, Perl Y, Chen Y, Spackman KA. Structural methodologies for auditing SNOMED. *Journal of Biomedical Informatics*. 2007; 40:561–581. [PubMed: 17276736]
29. Zhu X, Fan JW, Baorto DM, Weng C, Cimino JJ. A review of auditing methods applied to the content of controlled biomedical terminologies. *Journal of Biomedical Informatics*. 2009; 42:413–425. [PubMed: 19285571]

Table 1

Evolution of LOINC model

	LOINC model	Explanation
1	<analyte>:<specimen>:<precision>:<method>.	Initial model
2	<analyte>:<timing>:<specimen>:<precision>:<method>	Adding 'timing' axis
3	<analyte>.<subspecies>:<property>:<timing>:<system>: <precision>:<method>	Adding chemical subspecies and kind of property
4	<analyte>.<subspecies>^<chall>:<property>:<timing>:<system>:<precision>:<method>	Adding 'challenge' information

Table 2

Extensional Definitions (EDs) included local description, mean, standard deviation, units of measure, coded variables and frequency.

Extensional definitions	Example	Containing information for review
Local description	“Creatinine, 24 hr urine”, “Sodium urine”	Local description - mainly provides analyte information. In some cases, it also provides method (e.g. EIA), scale/property (e.g. titer), time (e.g. 24 hr) or system information (e.g. urine)
Mean	1.46, 137	Mean - provides scale/property information (e.g. SCnc/Qn). This is mainly for numeric tests.
Standard deviation	0.54, 7.02	Standard deviation - provides scale/property information (e.g. SCnc/Qn). This is mainly for numeric tests.
Units of measure	g/24 h, mmol/L, mg/dl	Units of measure - provides scale/property information. This is mainly for numeric tests.
Coded variables and their frequencies	1:8 (109), Negative (900), Positive (899), M1M1 (75)	Coded variables - provides scale/property (e.g. Titr/Qn or ACnc/Qn). For example, M1M1 is a reported value for the genetics test 'ALPHA-1-ANTITRYPSIN PHENOTYPE' and its frequency was 75.
Frequency	50, 184	Frequency - implies whether tests are frequent

Table 3

Example EDs from two different institutions. Two genetic tests are shown along with their local names, LOINC code mapping and reported values. By examining the EDs, we can determine how these two LOINC codes were used in each institution.

Source	Examples of EDs
A	[Local name]: Alpha 1 antitrypsin phenotyping
	[LOINC]: 6770-2: Alpha 1 antitrypsin phenotyping: Immunofixation: Prid: Pt: Nom: Ser/Plas
	[Coded Variables and their frequencies]: M1M1 (75), M1M2 (31), M1S (12), MM (8), M1Z (6)
B	[Local name]: Alpha 1 antitrypsin phenotyping
	[LOINC]:32769-2: Alpha 1 antitrypsin phenotyping : none: Imp : Pt : Nom : Ser/Plas
	[Coded Variables and their frequencies]: M1M1 (887), M1M2 (278), M1S (91), M1Z (88), MM (86), SEE NOTE (60)

Example of version spaces. Multiple local tests can be mapped to a single LOINC code, because even though the local tests are different, the difference is not significant according to LOINC naming rules.

Table 4

Index	Source/Local Name	LOINC	Analyte	Method	Time	Scale	Property	System
Version space: <Hemoglobin,?, Pt,?,>, Bld >								
1	Intermountain/Hemoglobin, Blood Quantitative Calculated	20509-6	Hemoglobin	Calculated	Pt	Qn	MCnc	Bld
2	Regenstrief/Hemoglobin Bld QN	20509-6	Hemoglobin	Calculated	Pt	Qn	MCnc	Bld
3	Regenstrief/Hemoglobin, POC	20509-6	Hemoglobin	Calculated	Pt	Qn	MCnc	Bld
4	Regenstrief/Hgb	718-7	Hemoglobin	None	Pt	Qn	MCnc	Bld
5	ARUP/Hemoglobin, inst.	718-7	Hemoglobin	None	Pt	Qn	MCnc	Bld
6	ARUP/Hemoglobin	718-7	Hemoglobin	None	Pt	Qn	MCnc	Bld
7	Intermountain/Hemoglobin, Blood Quantitative	718-7	Hemoglobin	None	Pt	Qn	MCnc	Bld
Version space: <Alpha-1-fetoprotein,?, Pt, Ser/Plas,?, Ser/Plas>								
1	ARUP/PATIENT~S AFP	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
2	ARUP/AFP (TUMOR MARKER)	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
3	ARUP/MEDIAN	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
4	Intermountain/Alpha- 1-Fetoprotein, Serum Quantitative~AFP TUMOR MARKER ONLY	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
5	Intermountain/Alpha- 1-Fetoprotein, Serum Quantitative	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
6	Regenstrief/AFP	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
7	Regenstrief/AFP SerPl QN	1834-1	Alpha-1- fetoprotein	None	Pt	Qn	MCnc	Ser/Plas
Version space: <Acyl carnitine,?, Pt,?,>, Ser/Plas>								
1	ARUP/CARNITINE, EASTERIFIED	14282-8	Acyl carnitine	None	Pt	Qn	SCnc	Ser/Plas
2	Intermountain/Carnitine Esters, Plasma Quantitative	1717-8	Acyl carnitine	None	Pt	Qn	MCnc	Ser/Plas

Table 5

The number of laboratory tests and the number of version spaces for <Analyte,?, Time,?,?, System> in each institution

Source	# of laboratory tests	# of version spaces
ARUP	1917	1601
Intermountain	1267	1089
Regenstrief	1693	1440

Table 6

Pairwise comparison of numbers of tests in each version space between institutions. A-I: ARUP & Intermountain, I-R: Intermountain & Regenstrief, A-R:ARUP& Regenstrief

# of tests	A-I	I-R	A-R
1	1240	1444	1435
2	589	414	631
3	117	96	134
4	41	42	55
5	20	12	23
6	3	12	9
7	4	5	4
8	5	2	4
9	0	0	2
10	0	1	2
11	0	1	1
13	1	0	1
15	1	0	0
28	0	1	0
37	1	0	0
Total	2022	2030	2301

Table 7

Numbers of pairs having contradictory knowledge and their classifications

	Deprecated codes	Raw measurement versus interpretation	Ambiguous codes	Total
ARUP & Intermountain	3	84	17	104
Intermountain & Regenstrief	0	108	1	109
ARUP & Regenstrief	2	106	4	112

Table 8

Examples of pairs having contradictory knowledge

Source	LOINC	Analyte	Method	Time	Scale	Property	System
ARUP	7650-5	Ambrosia elatior Ab.IgE	None	Pt	Qn	ACnc	Ser
Regenstrief	6085-5	Ambrosia elatior Ab.IgE	None	Pt	Qn	ACnc	Ser
Explanation: The LOINC code 7650-5 was already deprecated.							
Intermountain	44014-9	Porphyrins	None	Pt	Nom	Imp	Urine
Regenstrief	2818-3	Porphyrins	None	Pt	Ord	ACnc	Urine
Explanation: Both codes were used to report 'Interpretation', such as 'Negative, Positive or Normal'.							
ARUP	21654-9	CFTR gene mutation analysis	Molgen	Pt	Nom	Prid	Bld/Tiss
Intermountain	38404-0	CFTR gene mutation analysis	Molgen	Pt	Nar	Prid	Bld/Tiss
Explanation: Although 'Nom' and 'Nar' were designed to store different types of information, in practice they were used to store similar information.							
Regenstrief	49291-8	Porphyrins	None	Pt	Nar	Imp	Urine
Intermountain	44014-9	Porphyrins	None	Pt	Nom	Imp	Urine
Explanation: Although 'Nom' and 'Nar' were designed to store different types of information, in practice they were used to store similar information.							

Table 9

Numbers of pairs having semantically interoperable knowledge, which were classified into three categories. II.a is 'MCnc vs. SCnc' II.b is 'Pre vs. Post-coordinate' and II.c is 'log'

	I	II (II.a+II.b+II.c)	III	Total
ARUP & Intermountain	956	92(4+84+4)	117	1165
Intermountain& Regenstrief	559	17(3+14+0)	189	765
ARUP & Regenstrief	862	73(6+65+2)	186	1121

Table 10

Degree of semantic interoperability between two LOINC codes.

Source	LOINC	Analyte	Method	Time	Scale	Property	System
Intermountain	6085-5	Ambrosia elatior Ab.IgE	None	Pt	Qn	ACnc	Ser
Regenstrief	6085-5	Ambrosia elatior Ab.IgE	None	Pt	Qn	ACnc	Ser
Level I: For all relationships in level I, there is no meaning loss. Information is fully exchanged by identical codes.							
ARUP	14282-8	Acyl carnitine	None	Pt	Qn	SCnc	Ser/Plas
Intermountain	1717-8	Acyl carnitine	None	Pt	Qn	MCnc	Ser/Plas
Level II.a: For all relationships in the level II, there is no meaning loss. Information could be fully exchanged, but it requires additional processing. In the above case, 'SCnc' measures numbers of molecular of 'Acyl carnitine' and 'MCnc' measures the weight of 'Acyl carnitine'. Therefore, the value for 'MCnc' equals to 'SCnc' multiplied by the molecular weight of 'Acyl carnitine'.							
Intermountain	17096-9	Lymphocytes.k apps/100 lymphocytes	None	Pt	Qn	NFr	Bld
ARUP	20617-7	Lymphocytes.k apps/100 lymphocytes	None	Pt	Qn	NFr	XXX
Level II.b: ARUP uses the code in a post-coordinated way. When the value of 'Bld' is included in a second part of the message to further specify the value of 'XXX', the meanings of these two codes are equivalent.							
Intermountain	11011-4	Hepatitis C virus RNA	Probe.amp.tar	Pt	Qn	ACnc	Ser/Plas
ARUP	38180-6	Hepatitis C virus RNA	Probe.amp.tar	Pt	Qn	LaCnc c	Ser/Plas
Level II.c: 'LaCnc' is a logarithmic scale. So the value of log (ACnc) is equal to 'LaCnc'.							
Intermountain	20570-8	Hematocrit	None	Pt	Qn	VFr	Bld
ARUP	4545-0	Hematocrit	Spun	Pt	Qn	VFr	Bld
Level III: For all relationships in level III, there is some meaning loss. Partial information could be exchanged. Since one institution is using a methodless term (which could represent a value from a spun hematocrit, impedance, or hematology analyzer) it is not possible to know from the LOINC name whether these results are exactly equivalent. Although these two codes are not identical, but they do share identical 5 other axes. The two codes can be used interoperably in any context where the method is not considered important.							