

Article

LUMA: A Mapping Assistant for Standardizing the Units of LOINC-Coded Laboratory Tests

Kai Vogl ^{1,*}, Josef Ingenerf ^{1,2} , Jan Kramer ^{3,4}, Christine Chantraine ⁵ and Cora Drenkhahn ^{1,2,*} 

¹ Institute of Medical Informatics, University of Luebeck, 23562 Luebeck, Germany; josef.ingenerf@uni-luebeck.de

² IT Center for Clinical Research, University of Luebeck, 23562 Luebeck, Germany

³ LADR Laboratory Group Dr. Kramer & Colleagues, 21502 Geesthacht, Germany; j.kramer@ladr.de

⁴ Medical Department 1, University of Luebeck, 23562 Luebeck, Germany

⁵ IT Services for Laboratory Medicine, Intermed, 21502 Geesthacht, Germany; c.chantraine@intermed.de

* Correspondence: kai.vogl@online.de (K.V.); c.drenkhahn@uni-luebeck.de (C.D.); Tel.: +49-451-3101-5623 (C.D.)

Abstract: The coding system Unified Code for Units of Measure (UCUM) serves the unambiguous electronic communication of physical quantities and their measurements and has faced a slow uptake. Despite being closely related to popular healthcare standards such as LOINC, laboratories still majorly report results using proprietary unit terms. Currently available methods helping users create mappings between their units and UCUM are not flexible and automated enough to be of great use in trying to remedy this. We propose the “LOINC to UCUM Mapping Assistant” (LUMA) as a tool able to overcome the drawbacks of existing approaches while being more accessible even to inexperienced users. By mapping LOINC’s *Property* axis to representations within UCUM reflecting its semantics, we were able to formalize the association between the two. An HL7 FHIR back-end provides LUMA with UCUM unit recommendations sourced from existing lookup tables simply by providing it with a LOINC code. Additionally, the mappings users created may be used to perform unit conversions from proprietary units to UCUM. The tool was evaluated with five participants from the LADR laboratory network in Germany, who valued the streamlined approach to creating the mappings and particularly emphasized the utility of being able to perform unit conversions within the tool.

Keywords: UCUM; LOINC; HL7 FHIR; mapping; standardization; REST; RELMA



Citation: Vogl, K.; Ingenerf, J.; Kramer, J.; Chantraine, C.; Drenkhahn, C. LUMA: A Mapping Assistant for Standardizing the Units of LOINC-Coded Laboratory Tests. *Appl. Sci.* **2022**, *12*, 5848. <https://doi.org/10.3390/app12125848>

Academic Editors: Robert Vander Stichele, Elena Cardillo and Dipak Kalra

Received: 30 April 2022

Accepted: 5 June 2022

Published: 8 June 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The aim of medical informatics is to deliver the appropriate information in the appropriate form at the appropriate place at the appropriate time. In order to satisfy these goals, the information communicated needs to conform to a high standard of data quality [1]. This is especially the case in the context of laboratory testing, where the unambiguity of results is of utmost importance for subsequent decision-making [2]. Logical Observation Identifiers Names and Codes (LOINC) is a universal standard that is used to identify observations and test results in both laboratory and clinical settings. Tools such as the Regenstrief LOINC Mapping Assistant (RELMA) [3] help users wanting to adopt the standard by aiding them in mapping their locally used codes for these observations and test results to their appropriate LOINC counterparts. Despite the extensive uptake of this type of mapping, the enthusiasm for dealing with the related coding system UCUM [4], intended to relay physical quantities and their values in an unambiguous manner, has remained lukewarm.

An example for the diverging emphasis can be found in the ongoing Medical Informatics Initiative (MII) in Germany, which shall improve the availability of routine data for clinical research in a national joint effort [5]. All participating university hospitals are required to map their laboratory data to LOINC, aided by a common subset of currently 935 LOINC codes for 300 different use cases (referred to as the “MII Top 300” list). Still, the

conjoined standardization of locally used units with UCUM is not mandatory and therefore not pursued wholeheartedly, although the idiosyncrasies of proprietary terms have already been shown to impede the aspired integration of otherwise interoperable lab data [6,7]. Additionally, there is still a divide in the way certain measurements are handled even after the reunification of Germany, with the former West Germany having adopted the SI system in the late 1970s, whereas East Germany did not [8]. Still, the lack of motivation to perform a time-consuming and seemingly optional mapping does not come as a surprise, seeing as currently available methods for doing so have noticeable drawbacks.

As for existing approaches, users may try consulting the LOINC data set via either RELMA or the web assistant SearchLOINC [9] in order to ascertain whether an example for a UCUM unit is provided for a specific LOINC code, but this is very inflexible considering the fact that there is often more than one way to report the unit of a laboratory result. For example, measuring a mass could be performed in grams or the non-metric pound, as is customary in the US. Additionally, metric units can differ in magnitude from the prefix they are preceded by. The other approach is manual in nature and requires the user to have at least some expertise in the matters of UCUM by using aggregated lists of UCUM units as guidance for finding or formulating an appropriate match for their proprietary unit. On the official LOINC web presence, users can find a reference to one such document with data gathered by the Regenstrief Institute [10] but there are also other resources such as the “UCUM-Common” ValueSet provided by Fast Healthcare Interoperability Resources (FHIR) [11]. While offering a broader range of possible units to choose from, these lists are not associated with specific LOINC codes in any way and users have to make an informed guess regarding the mapping they deem appropriate. Ideally, one would like to combine the flexibility of these extensive lists with the more automated approach of RELMA and SearchLOINC. In light of the lack of readily available support for such endeavors and a complete absence of formalized associations between LOINC and UCUM that could facilitate digital processing, we present the “LOINC to UCUM Mapping Assistant” (LUMA) as a tool to close this gap.

For this project, LOINC’s *Property* axis acts as a bridge between the proprietary units and UCUM. This approach requires a user’s proprietary units to already be associated with a LOINC code, as we mapped all quantitative LOINC *Properties* to a representative base unit combination in UCUM. This representation serves as a ground truth for verifying which UCUM units can potentially be associated with a given LOINC code, and thereby the proprietary unit it is associated with. Users inexperienced in either standard interact with the graphic interface LUMA in order to create mappings between their locally used proprietary units and the UCUM standard simply by providing the tool with a valid LOINC code and making a choice from UCUM units associated with it. After users have established such mappings between proprietary units and UCUM, they may perform unit conversions within UCUM based on them. Seeing as there are no official software tools that are associated with the UCUM standard, we used our own implementation from previous research efforts in this project.

2. Materials and Methods

The LOINC axis model is the foundation of all LOINC codes. In particular, there are two axes that play a major role in this project, these being the type of *Scale* and the kind of *Property* (also known as kind of quantity) associated with a quantitative observation or test result. A standard closely related to LOINC is UCUM. This coding system deals with the unambiguous communication of physical units and their associated quantities currently used in international science, business and engineering. Each UCUM unit can be represented via seven metric base units and a numeric value. This so-called canonical form is of vital importance when establishing the connection between LOINC and UCUM by mapping one such representation to each quantitative *Property*. Although there is a loose association between LOINC’s quantitative *Properties* and UCUM [6], the LOINC manual outlines no explicit way in which the two relate in detail [12]. Despite this, about half of all

LOINC codes have at least one example UCUM unit associated with them in the LOINC data set. The idea behind formalizing the connection between LOINC and UCUM was to make use of the fact that each LOINC code is always associated with a *Property*. Moreover, a subset of these *Properties* describe physical quantities which can be represented as UCUM units. In this way, the LOINC *Property* axis acts as the mediator between the two standards, by which an explicit association was able to be established. This knowledge was then formalized via a FHIR ConceptMap resource.

2.1. LOINC and the Property Axis

LOINC is a clinical terminology developed by the Regenstrief Institute. It is used, among other applications, to identify laboratory observations and test results. LOINC's adoption into relevant communication standards such as HL7v2 with its OBX (observation) segment played a significant role in overcoming the heterogeneity introduced by countless local identifiers used across different institutions. An important aspect that helped facilitate the acceptance of LOINC was the freely available tool RELMA, which aids users in mapping their locally used, proprietary identifiers to standardized LOINC codes [13,14].

Each LOINC code corresponds to a combination of five to six axis elements, as shown in Figure 1. In this project, two of the axes play a major role: Firstly, the *Scale* type differentiates between, for example, qualitative, quantitative or nominal observations. Secondly, the type of *Property* differentiates between the kinds of quantities relating to the same substance. These *Properties* include concepts such as “Substance (Molar) Concentration”, “Mass per Area”, “Number per Volume” and many more. Quantitative entries make up 44,299 out of 98,268 total LOINC codes (roughly 45%) in the current LOINC version 2.72.

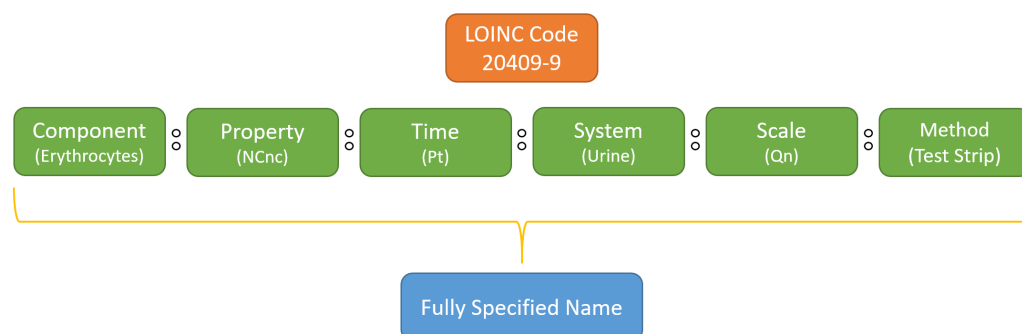


Figure 1. This figure shows all six LOINC axes that end up forming a specific LOINC code for “Erythrocytes [# / Volume] in Urine by Test Strip”. Concatenating the parts using colons yields the fully specified name of a LOINC code.

2.2. UCUM and the Base Unit Representation

The Unified Code for Units of Measure is a coding system developed by the Regenstrief Institute that aims to facilitate the unambiguous electronic communication of all physical units and their associated quantities currently used in international science [4]. Moreover, it plays an important role within the LOINC standard as well being the recommended way of reporting quantitative results [12]. All valid UCUM units are exclusively part of the seven-bit US-ASCII character set. The coding system consists of a set of simple terminal symbols, also called unit atoms, which can be combined by following the rules set out by the UCUM expression syntax (<https://ucum.org/ucum.html#section-Syntax-Rules> (accessed on 29 April 2022)), resulting in an infinite amount of possible unit terms with precise semantics. Unit atoms can be connected via the division (“/”) or multiplication operator (“.”). Metric units may be prefaced with a metric prefix (e.g., “mg” for milligrams) and integer suffixes signify the dimensionality of a unit (e.g., “cm2” and “s-1”). Additionally, there are two important types of brackets used. Round brackets retain their established usage of ordering operations, whereas curly brackets at the end of a valid unit are used to add annotations which may also be used in place of a unit symbol or integer. The content

of these annotations carries no semantic value within UCUM and is disregarded when processing units. The possibility to append such notes exists in order to accommodate traditions and habits in chemistry and biomedical sciences (e.g., mg{creat} conveying “milligrams of creatinine”) [15].

An important aspect of UCUM is the commensurability of units. Each quantity Q is the product of a measurement μ and its unit u . Moreover, the same kind of quantity can be expressed in a different unit u' with an accompanying measurement μ' . For example, one could use both meters or yards to determine a previously unknown length (e.g., $3.6 \cdot \text{m} = 3.937 \cdot \text{yd}$). In this way they would be described as being commensurable.

$$Q = \mu \cdot u = \mu' \cdot u' \quad (1)$$

One could also use the meter stick to measure the yard stick's length or vice versa. Therefore, the following equation holds:

$$u' = y \cdot u \quad (2)$$

where y is the magnitude of the unit u' measured as a quantity by unit u (e.g., $\text{yd} = 3600/3937 \cdot \text{m}$). It follows that any fixed quantity can be used as a unit to measure all other quantities that are of the same kind. The formula for unit conversion is then derived from Equations (1) and (2):

$$\mu' = \mu \cdot \frac{u}{u'} = \mu \cdot \frac{1}{y} \quad (3)$$

where $\frac{1}{y}$ can be described as a conversion factor between different but related units. UCUM proposes seven metric base units from which to derive all other measurements. A valid unit may also have no dimensions at all. In this case, it describes a numeric value. A special numeric value is being described by the default unit “1” (or “the unity”). A list of all base units is displayed in Table 1.

Table 1. This table contains all seven base units used within UCUM [15].

Base Unit	Kind of Quantity	Case Sensitive Symbol
Meter	Length	m
Second	Time	s
Gram	Mass	g
Radian	Plane Angle	rad
Kelvin	Temperature	K
Coulomb	Electric Charge	C
Candela	Luminous Intensity	cd

Each unit that does not already consist of only base units is classified as a derived unit, which in turn can always be broken down into its so-called canonical form that contains only base units and a magnitude (e.g., an hour turns into “s” and 3600). This canonical form is used to calculate conversions between units if the base unit dimensions are equal, and therefore also acts as a way to test if two units are commensurable [16].

2.3. Fast Healthcare Interoperability Resources

FHIR is a standard by Health Level 7 for exchanging electronic health records. Building upon the experience gained from trying to establish HL7v2 and HL7v3, it is intended to leverage existing IT infrastructure, especially in the form of internet technologies such as the Representational State Transfer (REST) architecture to represent information using the XML or JSON formats. In order to exchange data, FHIR defines a substantial amount of data structures, which are referred to as resources. Resources are the smallest units of information that are considered logically discreet and semantically unambiguous [17]. In addition, FHIR describes an interface to exchange this data in the form of the FHIR

API. There are three FHIR resources that play an important role in this project, all of which are sourced from the FHIR Terminology Module. This module is used for providing the terminology service functionalities needed to provide the use of coded data within FHIR [18]. The resources provided consist of CodeSystem, ValueSet and ConceptMap.

A CodeSystem generally identifies terminologies, ontologies and enumerations which are sources for various identifiers (codes) themselves. A CodeSystem contains various information pertaining to those systems, such as which concepts are defined by it; if it is defined by a compositional grammar; if it is case sensitive; and which FHIR filter operations can be performed on it [19]. ValueSets specify subsets of codes drawn from one or more CodeSystems. ValueSets are always intended for use within a particular context and, among other use cases, define the allowable content for coded elements within those [20]. Lastly, a ConceptMap describes how concepts from one CodeSystem relate to another CodeSystem. Mappings of this kind are only defined in the context of a particular use case (such as how concepts in LOINC relate to concepts in UCUM in the context of this project) and they are generally considered to be unidirectional from source to target. Each element that describes the relationship between two concepts additionally always has to define the nature of this mapping (e.g., broader; narrower; equivalent; somewhat related, though unclear; and not related at all). The concepts (or codes) for the source and target are drawn from the aforementioned ValueSet resources [21].

2.4. Bridging the Gap between LOINC and UCUM

LOINC would be an ideal proxy between the locally used units and UCUM. Firstly, LOINC has garnered widespread usage in the context of clinical and laboratory settings [22], which makes it an ideal foundation to piggyback off. Secondly, it supplies critical information about the proprietary units that can be processed in an automated way, namely in the form of the *Scale* and *Property* axis that each LOINC code is associated with. In the same vein, each valid UCUM unit is associated with a specific base unit representation that serves to answer questions regarding the nature of the physical properties of that unit and how it relates to other UCUM units. By trying to establish a link between all of LOINC's quantitative *Properties* and a base unit representation for each respective one, we could infer how any valid UCUM unit relates to the *Property* axis and, by extension, the proprietary unit. Staying true to the spirit of curbing the effects of heterogeneity induced by proprietary codes, this idea would need to be formalized in a way that makes use of existing healthcare standards. FHIR offers the necessary components in the form of the provided resources CodeSystem, ValueSet and ConceptMap that can be deployed on FHIR terminology servers. Moreover, there is the need for a UCUM tool that is able to canonize and convert units in addition to being able to generate human-readable display names for them. Along with allowing machines to make use of this formalized knowledge, a tool is needed to support users with little expertise in either LOINC or UCUM to establish mappings between their units and UCUM, simply by providing the software with a valid LOINC code that has already been mapped to a certain procedure and offering them a set of curated UCUM units to map their local units to. Keeping the target audience for this tool in mind, it would need to be very streamlined, providing only the most basic of functions needed to accomplish this mapping in a way that minimizes the room for error. The recommendations for these units would ideally be sourced from aggregations of UCUM units that are “tried-and-true” for common use-cases. Both the FHIR and the LOINC web presence provide such lists, and they can be turned into standardized FHIR ValueSets for the terminology server to make use of. In addition, there are also the aforementioned UCUM example units in the LOINC data set. The recommendation mechanism could then be realized through a ConceptMap acting as a filter that examines the recommendations of the ValueSet used, making sure they match the *Property* (and thereby the base unit representation) of the provided LOINC code.

Figure 2 sketches the aforementioned ideas without going into detail about the actual implementation. Users utilize their proprietary units that are associated with LOINC as the

input for the mapping tool, receive a curated set of recommendations and make an informed choice based on them. Additionally, they may also specify a quantity and proprietary unit to convert to a UCUM unit of their choice. Internally, the software makes use of a powerful back-end containing both the theoretical knowledge stored on the terminology server and the capabilities of a UCUM tool for dealing with tasks related to the standard, such as verifying the compatibility of units and performing conversions between them.

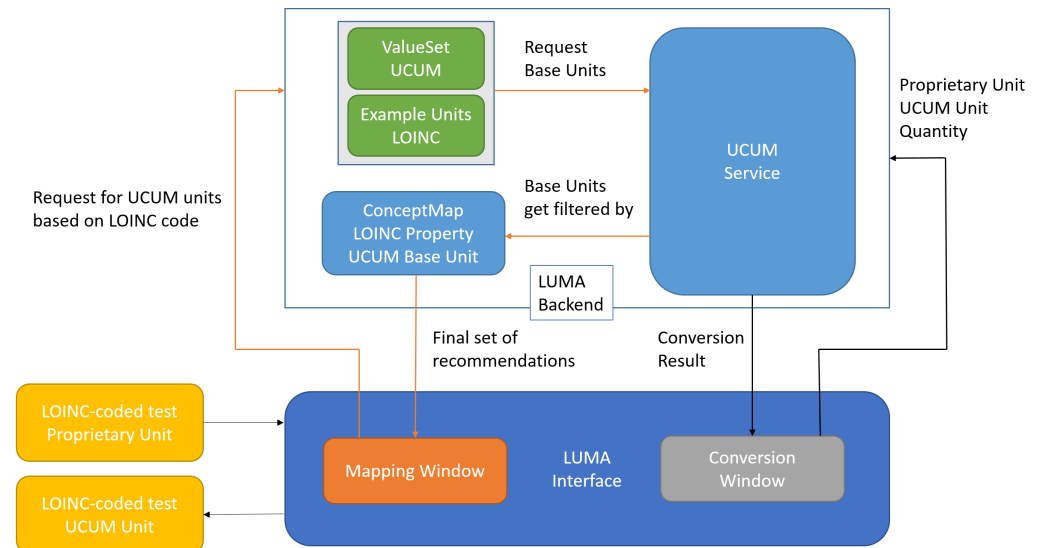


Figure 2. This figure outlines the rough interplay between the components. Specific implementation details have been abstracted.

3. Results

LUMA was able to be implemented based on the previous considerations. Firstly, generating the canonical form of a UCUM unit and extracting the base unit representation was of major importance when establishing the mapping between LOINC and UCUM. Initially, choosing an appropriate unit to generate the representation for all units that would later fit a *Property* is rather arbitrary, but there is a caveat in the form that the mapping between LOINC and UCUM is unidirectional in nature, seeing as there are multiple *Properties* that share the same base unit representation. Secondly, a ConceptMap was generated to formalize the associations between LOINC's property axis and the UCUM base unit representations.

Users interact with LUMA via a graphical user interface that consists of two components. In the first step, users establish mappings between their proprietary units and UCUM by inputting a LOINC code and finding an appropriate unit match from a subset of eligible choices sourced from a list containing roughly over 1300 entries in total. In the second step, users may then optionally perform unit conversions involving their proprietary units by making use of the previously established mappings to UCUM. The LUMA back-end consists of a FHIR terminology server providing the LOINC data set, FHIR CodeSystem, ValueSet and ConceptMap resources, as well as a RESTful UCUM service used for generating human-readable display names, converting units and verifying their base unit representation to match against LOINC's *Properties*. Both tools are coordinated via a Controller implemented independently from the details of the graphical user interface, making it flexible enough to accommodate different usage paradigms.

Of the 176 quantitative *Properties*, 159 were able to be mapped with confidence. Based on data gathered from three important lists of commonly used LOINC codes, the bulk of all observations and testing actually gathers around a very small subset that is entirely covered by our results. A user evaluation happened with five participants of the LADR laboratory network (<https://www.ladr.de/> (accessed on 29 April 2022)) in Germany.

3.1. The UCUM Parser

The UCUM web presence does not provide users with an official implementation of the standard. Instead, the specification document extensively describes the guidelines which a possible implementation has to adhere to without specifying how exactly the results are to be achieved. For this project, we revised our RESTful UCUM implementation from previous research efforts. By using our own implementation for the current LUMA project, we were certain of its capabilities and also easily able to make changes to it if necessary. We designed the RESTful UCUM service's response structure to comply with the FHIR Parameters [23] resource commonly used for exchanging information within the FHIR digital ecosystem and common denominator between all back-end software components used for LUMA.

In a more general sense, parsing describes “the process of structuring a linear representation in accordance with a given grammar” [24]. In the case of UCUM, this linear representation is a string that is analyzed in order to gain further insights into how it relates to the syntactic rules (grammar) and semantics laid out by the UCUM specification. In order to assess the meaning and validity of an input string in the context of UCUM there are only a handful of components that need to be evaluated:

- Operands containing:
 - A metric prefix (optional);
 - A unit symbol or an integer;
 - A dimension exponent (optional);
 - An annotation (optional).
- Arithmetic operators:
 - “.” for multiplication;
 - “/” for division.
- Parentheses further regulating the order of operations

These components, from here on referred to as tokens, are the smallest semantically meaningful bits of information that a UCUM unit can be decomposed to. Annotations themselves are meaningless for machine processing within UCUM, but they are allowed to contain the entire range of ASCII symbols ranging from 33–126, which includes both arithmetic operators, parentheses and strings that would otherwise be considered invalid. Due to the fact that we wanted to retain annotations for generating a unit's display name in full (e.g., “g{feathers}” to “grams of {feathers}”) we needed to account for this problem when initially performing syntax checks and separating the tokens afterwards. In the next step, the tokens are rearranged from an infix style notation to a postfix style notation, also known as reverse polish notation. In reverse polish notation, operators follow their operands. This style of notation is of particular interest in computer science, as it enables the stack-based processing of expressions [25]. The UCUM expression “L/(24.h)” is a “Volume Rate” in LOINC. After splitting the tokens and converting them into reverse polish notation, the output would look like this:

L 24 h . /

Now there is no need to make use of parentheses in order to clarify the order of operations. One of the ways to convert an infix notation to reverse polish notation is Dijkstra's Shunting-Yard algorithm [26]. For our use case, we can further simplify it. The reasons for the simplification are twofold. Firstly, all operators are left-associative, and secondly, the operators have equal precedence based on the fact that only multiplication and division are possible. After splitting the string into tokens and rearranging them via the Shunting-Yard algorithm, a binary tree reflecting the initial infix notation is generated. It is visualized in Figure 3.

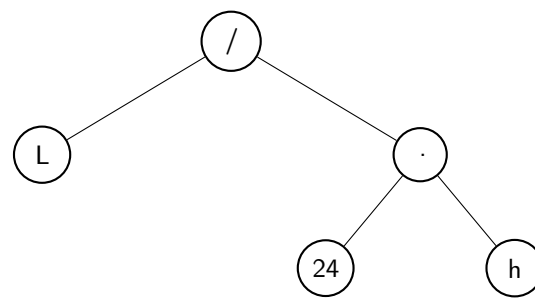


Figure 3. This figure shows how the infix notation for L/(24.h) would look if expressed in a binary tree. The tree itself was generated by evaluating the respective reverse polish notation.

Due to the fact that many UCUM units are derived units that are made up of other units, this binary tree will generally undergo further changes until only base units and integers remain, at which point it is traversed to generate its canonical form.

A token initially falls either into the category of operator or operand. In order to evaluate the expression, all created nodes also contain various other pieces of information, such as their value, a metric prefix exponent, their dimensionality or annotations. Creating such a node for an operator is trivial; operands, on the other hand, require a more elaborate approach. Operands are initially examined starting from the end of the string in order to trim annotations and the dimension exponent. What is left is a possible UCUM symbol with an optional metric prefix. A node (and with it the entire input) is declared invalid if the unit symbol or prefix can not be found in the UCUM database or if the prefix is attached to a non-metric unit symbol. If the unit symbol the node is associated with is not a base unit or integer, we recursively call the function for generating a binary tree on it, replacing the old node in the process. The idea behind the evaluation of the binary tree is to multiply the final value of all its nodes. In order to make sure that operators do not skew the result, they behave similarly to neutral elements. Their value and dimension is set to one and their metric prefix exponent amounts to one as well (as in 10^0). Moreover, all division tokens are treated as multiplications internally by making use of the reciprocal relationship between the operations. By inverting the dimension exponent of a division node's right child, we cascade the effect down the entire subtree that may follow it later.

In the given example, there are two nodes that can be further dissolved. On the left side, there is a liter, which is represented by a cubic decimeter. On the right side, there is the symbol for hour, which in turn can be unfolded into a minute and then into seconds, respectively. Integers are always completely unfolded just like base units. The root of this tree is a division operator. In order to calculate the final value later on, its right subtree was inverted. The final form consists of only multiplication operators, base units and integers, as can be seen in Figure 4.

After the entire tree has been unfolded, it can now be traversed to calculate its value, but also to determine its base unit representation by adding up the dimension exponents of each respective base unit node. In this way the entire canonical form of the UCUM unit in question is generated consisting of the base unit representation “m3.s-1” and the associated magnitude $(10^{-1} \cdot 1)^3 \cdot (10^0 \cdot 24)^{-1} \cdot (10^0 \cdot 3600)^{-1} = 1.1574074074074076 \cdot 10^{-8}$.

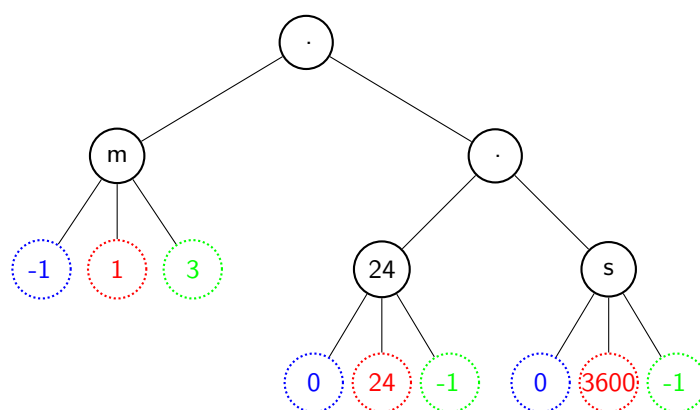


Figure 4. The figure shows the final unfolding of the binary tree. The dotted nodes are not actual children. Instead, they signify the metric prefix exponent (blue), value (red) and dimension (green) associated with the respective UCUM symbol. Note that the initial division operator and its subtree have been inverted by multiplying the dimension exponent with minus one.

3.2. Mapping LOINC to UCUM

Having explained the fundamentals of generating the base unit representation for a given UCUM unit, we want to now present the results and some more thoughts on formalizing the relationship between LOINC and UCUM. In order to establish a mapping between the two standards, the subset of *Properties* that conformed to the quantitative *Scale* axis was used to formulate a base unit representation for each entry. The process of formulating such a representation incorporated the following steps:

1. Check the official LOINC data set to see which example units are associated with a certain *Property*;
2. Verify if they are consistently commensurable across all entries;
3. Compare the example units to keywords in the LOINC guide to see if they line up semantically.

If there were no example units given for a *Property*, we formulated an example based on keywords in the LOINC guide. A “Mass Concentration” should reflect the fact that it describes a mass divided by a volume, for example. The LOINC guide also served as the authoritative point of reference when deciding on how to deal with small inconsistencies in the data set. Strong inconsistencies that were not able to be reconciled were disregarded. They are mentioned in the evaluation.

It is important to note that there was a lot of wiggle room when choosing an appropriate unit to generate the base unit representation from initially. That is due to the fact that picking one such unit is arbitrary as long as it reflects the intent of the *Property*. A gram, kilogram and pound all describe the concept of a mass, and it does not matter which one of these is chosen when generating the base unit representation. There is a caveat to this though: when going from instance level to the level of the underlying model (the canonical form), we have to be mindful of the loss of semantic precision. A gram describes a mass, but not every mass is measured in grams. In the context of LOINC and UCUM, this means there are multiple *Properties* represented by the same combination of base units. For example, both “Length” and “Volume/Area” share the same base unit meter, but the meaning they convey on the instance level appears rather different. In other words, this mapping is unidirectional. This is even more so the case for dimensionless *Properties*, the most prominent example being the default unit “the unity” with a value of one. These types of units also make a common appearance as numeric values such as “24” or “10⁻³”, but there are also quite a few unit symbols such as “%” and “mol” that are nothing but dimensionless values. In addition, units are also able to annul each other, which can result in a dimensionless unit, as is the case with “g/g”. While these expressions and their conversions into one another are all legitimate within UCUM, there is the question of how this

aspect should be accounted for in the context of this project, seeing as there are 78 LOINC *Properties* that are dimensionless. They are often related to ratios or certain “scores” and “counts” expressed as annotations such as “{ct_score}”. For this initial implementation of LUMA, we have not yet opted to decrease the flexibility offered by a broad range of possible choices until we receive more feedback on how to handle them. Table 2 shows an excerpt of the created mappings.

Table 2. This table shows some examples that detail how a LOINC *Property* might be represented on the instance level (a UCUM unit that can be used for measurements related to that *Property*) and its representation in the underlying model.

LOINC Property	Possible UCUM Unit	UCUM Base Unit Representation
Mass	[lb_av]	g
Volume	mL	m3
Volume Content	mL/[lb_av]	m3.g-1
Time Mass Concentration	ug.h/L	m-3.g.s
Frequency	Hz	s-1
Length	m	m
Volume/ Area	L/m2	m

Something that becomes apparent when studying the table is how simple formulating a base unit representation often is. Many keywords are used in a consistent way across the entire *Property* table, which helps to increase our confidence in the created mappings even when no example unit is given or if there are conflicts relating to them.

3.3. Generating the ConceptMap

The results of the formalization are stored in a ConceptMap. The tuples defined in a ConceptMap are a unidirectional association between source and target. The first code marks the source, in this case a code pertaining to LOINC’s *Property* axis. “MRat” is the official abbreviation for a “Mass Rate”, a measurement of a mass over a time interval (e.g., iron in 24 h urine). This code from the source gets associated with a code in the target. In this case, “MRat” is represented by the base unit combination “s-1.g” ergo gram per second or, in a more abstract fashion, mass over time. FHIR ConceptMaps generally require there to be an equivalence statement to describe the nature of the mapping. We decided to go with the description “narrower” in order to reflect the loss of precision going from *Property* to base unit representation. The ConceptMap was generated based on a CSV file that can be found in the supplementary materials. The tool Snapper (<https://ontoserver.csiro.au/site/technical-documentation/snapper-documentation/snappermap-guide/> (accessed on 29 April 2022)) was used to create the ConceptMap itself. The following is an excerpt of it:

```
{
  "code": "MRat",
  "target": [
    {
      "code": "s-1.g",
      "equivalence": "narrower",
      "comment": "Properties can share the same base units"
    }
  ]
}
```

In our implementation, this ConceptMap accomplishes the important tasks of one, filtering out UCUM recommendations that do not fit the given *Property*; and two, minimizing the effects of invalid example UCUM units in the LOINC data set by being able to detect them.

3.4. The Software Architecture

The software back-end around LUMA was designed with the Model-View-Controller (MVC) design pattern in mind. The FHIR terminology server Ontoserver acts as one part of the model containing all data pertaining to LOINC as well as the standardized FHIR resources that describe how each LOINC *Property* relates to UCUM. Ontoserver was developed with the intent of further increasing the uptake of clinical terminologies such as LOINC and the Systematized Nomenclature of Medicine (SNOMED) by supporting them out of the box and providing users with a powerful search mechanism for dealing with the complex and vast amounts of content in an effective and responsive manner. The stored terminologies are kept up to date via Ontoserver's syndication mechanism, which enables each instance of the tool to act as a client or a server for fetching or distributing data from other Ontoserver instances [27]. Possible UCUM units to map to are sourced from a ValueSet that may be specified beforehand. We ended up using the FHIR "UCUM-Common" ValueSet (<https://build.fhir.org/valueset-ucum-units.html> (accessed on 29 April 2022)) as it contained about 500 entries more than an aggregation provided by LOINC with the exception of roughly 45 units. The other part of the model consists of a RESTful UCUM service that deals with all requests regarding UCUM units, namely the canonization, conversion and display name generation for units. Ontoserver and the UCUM service never interact with each other. Instead there is a Controller, also implemented as a RESTful service, that encapsulates requests involving the model. Users do not interact with the back-end directly. Instead, they use the graphical user interface LUMA that communicates with the Controller. It should be noted that LUMA is only one imaginable way of making use of the back-end. The Controller is perfectly capable of being used for machine-to-machine communication, having been designed first. The LUMA interface was implemented later as one possible use case. A high-level abstraction of the component interactions is shown in Figure 5.

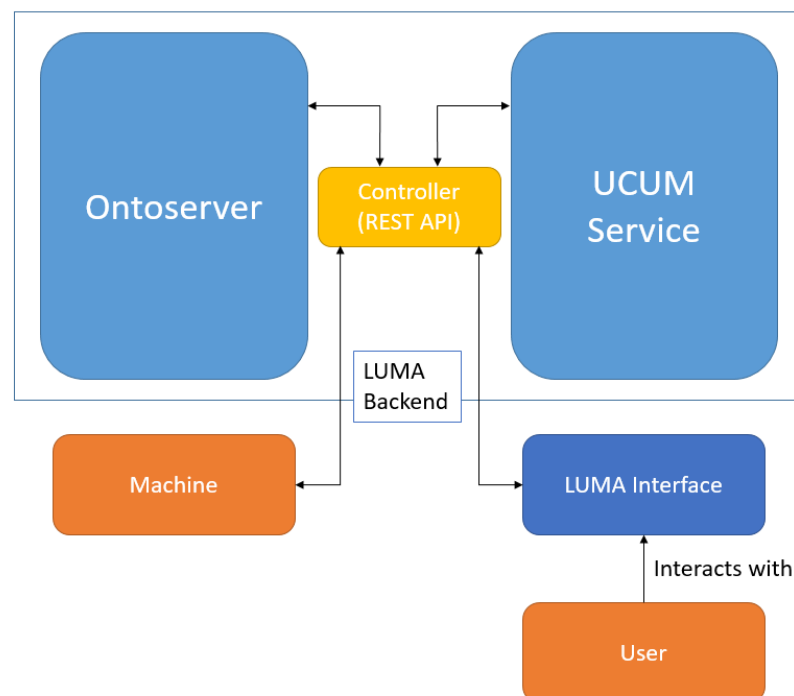


Figure 5. This figure shows the interactions of each component. The Controller manages the flow of information and triggers the terminology server Ontoserver and the UCUM service in order to generate responses related to mapping tasks. Machines interact with the Controller via its REST interface, while humans do so via the graphical user interface LUMA. The communication happens via HTTP calls and the JSON file format is used for data interchange.

3.5. The LUMA Interface

The graphical user interface is split into two components. In the first step, users create mappings between their proprietary units and UCUM via a valid LOINC code, and afterwards they may optionally carry out unit conversions based on the mappings they created. Some of our ideas behind streamlining the interface in this way can be found in the discussion section.

3.5.1. The Mapping Window

In the first step, users create mappings between their proprietary units and UCUM. In order to formulate these rules there are three requested inputs:

- The proprietary unit term;
- A descriptor of the laboratory test;
- A valid LOINC code.

After verifying whether the given LOINC code is valid, this incomplete tuple is added to the mapping table below. By selecting an incomplete mapping, users can opt to look up UCUM units associated with the LOINC code provided. On the right side of the LUMA interface, users may then find several possible matches for the proprietary unit. These recommendations are sourced from the LOINC data set in the form of a UCUM example unit (should it exist and be appropriate) and the ValueSet specified beforehand. As some UCUM units can appear rather cryptic, we also opted to add a human-readable written-out description of the unit that is generated automatically. When users find a match they deem appropriate, they may map the selected unit to their incomplete tuple. A completed table entry contains these additional values:

- The UCUM unit selected for the mapping;
- Its written-out display name;
- The mapping status (Complete).

Users may additionally manage these mappings by deleting entries. Once one or more mappings have been created, users can export them into a simple TSV file from which they can be recreated by loading them on the next start or during a running session. An exemplary view can be seen in Figure 6.

3.5.2. The Conversion Window

After having created at least one complete mapping rule, users may then optionally perform unit conversions from their proprietary units to UCUM based on these. The process is rather simple, as they only need to provide a numeric value as an input and choose the mapping they want to work with. Once a complete mapping rule is selected, all appropriate UCUM units from the ValueSet are provided as conversion targets. By selecting one such entry and executing the unit conversion, the final result is displayed adjacent to the input. An exemplary view can be seen in Figure 7.

3.6. Evaluation

The evaluation happened on two levels. In the first step, we generated quantifiable results by determining how many *Properties* were able to be mapped to a base unit representation. For this purpose, we calculated the mapping coverage we were able to achieve compared to all quantitative *Properties*. In order to estimate how thorough the coverage for real-world data would be, we matched our mappings against the Top 2000 US, Top 2000 SI and Top 300 documents, which are intended to aid users in mapping their laboratory identifiers to commonly used LOINC codes. In the second step, data on LUMA itself were gathered with five participants of the LADR laboratory network in Germany via a video conference and a subsequent discussion. Additionally, a questionnaire concerning UCUM and previous mapping efforts was handed to the participants.

In terms of raw numbers, the current 210 LOINC *Properties* in Version 2.72 can be categorized as follows for the purpose of this project:

- Thirty-four did not belong to the quantitative subset.
- Of the remaining one hundred and seventy-six *Properties*:
 - One hundred and fifty-nine were mapped with confidence;
 - Three were excluded based on ambiguous usage;
 - Ten were excluded based on specifying ranges, dates or time via annotations, effectively serving no purpose within UCUM;
 - Four were excluded based on being deprecated, their use being discouraged or not existing in the LOINC data set without any further clues on how they may be represented.

Guided Mapping
UCUM Convert
About

Proprietary Unit Term

Proprietary Laboratory Descriptor

LOINC Term

Add LOINC Term to Mapping Table

Proprietary Unit Term	Proprietary Laboratory D...	LOINC Term	UCUM Term	Human-readable UCU...	Mapping Status
mltr	Blutvolumen Szintigrap...	30899-9	mL	(milliliter ^ 1)	Complete
mlg/24std	Laktose in 24h Urin	2554-4	mg/(24.h)	(milligram ^ 1) per (24 ...	Complete
mgramm/kgramm	Caesium im Haar	5615-0	mg/kg	(milligram ^ 1) per (kilo...	Complete
g/Liter	Eisen in Wasser	17025-8	g/L	(gram ^ 1) per (liter ^ 1)	Complete
millgr	Dosis Nikotin	4350-5	mg	(milligram ^ 1)	Complete

Look up possible UCUM Terms for the selected entry

Delete Selected Row

Load Table from TSV File

Save Table as TSV File

LOINC Term
30899-9

Recommended UCUM Term for this measurement **LOINC**
From Regenstrief

mL

Common UCUM Terms used for similar measurements **HL7 FHIR**

UCUM Term	Human-readable UCUM Term
[cin_i]	(cubic inch ^ 1)
[cup_us]	(cup ^ 1)
[drp]	(drop ^ 1)
[fdr_us]	(fluid dram ^ 1)
[foz_br]	(fluid ounce ^ 1)
[foz_us]	(fluid ounce ^ 1)
[gal_br]	(gallon ^ 1)
[gal_us]	(Queen Anne's wine gallon ^ 1)
[min_us]	(minim ^ 1)
[pt_br]	(pint ^ 1)
[qt_br]	(quart ^ 1)
[qt_us]	(quart ^ 1)
[tbs_us]	(tablespoon ^ 1)
[tsp_us]	(teaspoon ^ 1)
cl	(centiliter ^ 1)
cm3	(centimeter ^ 3)
dL	(deciliter ^ 1)
fL	(femtoliter ^ 1)
hL	(hectoliter ^ 1)
kL	(kiloliter ^ 1)
L	(liter ^ 1)
mL	(milliliter ^ 1)
mm3	(millimeter ^ 3)
nL	(nanoliter ^ 1)
pL	(picoliter ^ 1)
uL	(microliter ^ 1)
[pt_us]	(pint ^ 1)
[bbl_us]	(barrel ^ 1)
[bf_i]	(board foot ^ 1)
[bu_br]	(bushel ^ 1)
[bu_us]	(bushel ^ 1)
[cft_i]	(cubic foot ^ 1)
[cr_i]	(cord ^ 1)
[crd_us]	(cord ^ 1)
[cyd_i]	(cubic yard ^ 1)
[dpt_us]	(dry pint ^ 1)
[dqt_us]	(dry quart ^ 1)
[fdr_br]	(fluid dram ^ 1)
[gal_wi]	(historical winchester gallon ^ 1)
[gil_br]	(gill ^ 1)
[gil_us]	(gill ^ 1)
[min_br]	(minim ^ 1)
[pk_br]	(peck ^ 1)
[pk_us]	(peck ^ 1)
st	(stere ^ 1)

Map the selected UCUM term to your proprietary term

Figure 6. The left side of the mapping window deals with the user's instance data while the right one is based on the formalization we postulated.

Even when excluding 17 *Properties* from the overall result, we managed to achieve a coverage of roughly 90% for the quantitative subset. When encountering inconsistencies in the way example units were used, we weighted keywords and descriptions in the LOINC guide highest, seeing as their usage was more coherent across all entries, especially if the example units were obviously inappropriate. For the three *Properties* we excluded based on their ambiguous usage, we were not able to reconcile the inconsistencies. They include "Fluid Conductance", "Fluid Resistance" and "Resistance". "Fluid Conductance" is currently used to measure fluid conductivity, electrical conductivity and diffusion capac-

ity. “Fluid Resistance” is currently used to measure airway resistance and *specific* airway resistance. “Resistance” is currently used for measuring hemodynamic resistance and impedance. Although formulating a base unit representation for each of these use cases by themselves is not an issue, they are incompatible when assuming that *Properties* should be unambiguous.

The screenshot shows the 'UCUM Convert' window. On the left, the 'Measured Quantity' is '1000'. Below it is a table with columns: 'Proprietary Unit Term', 'Proprietary Laboratory Desc...', 'LOINC Code', 'UCUM Term', 'Human-readable UCUM Te...', and 'Mapping Status'. The table lists several units like 'mltr', 'mg/24std', 'mgramm/kgamm', 'g/Liter', and 'millgr'. On the right, the 'Conversion Result' is '1.0'. Below that is a list of 'Similar UCUM Terms' including '[cin_i]', '[cup_us]', '[drp]', '[fdr_us]', '[foz_br]', '[foz_us]', '[gal_br]', '[gal_us]', '[min_us]', '[pt_br]', '[qt_br]', '[qt_us]', '[tbs_us]', '[tsp_us]', 'cL', 'cm3', 'dL', 'fL', 'hL', 'kL', 'L', 'mL', 'mm3', 'nL', 'pL', 'uL', '[pt_us]', '[bbl_us]', '[bf_i]', '[bu_br]', '[bu_us]', '[cft_i]', '[cr_i]', '[crd_us]', '[cyd_i]', '[dpt_us]', '[dqt_us]', '[fdr_br]', '[gal_wi]', '[gil_br]', '[gil_us]', '[min_br]', '[pk_br]', '[pk_us]', and 'st'.

Figure 7. In the conversion window, users may now perform unit conversions by selecting one of their proprietary units and the UCUM unit they want to convert it to.

In order to simplify the uptake of mapping proprietary laboratory codings to LOINC, two data sets are provided on the official LOINC web presence. Each contains roughly 2000 LOINC codes that are supposed to cover 98% of use cases for typical lab result volumes. The data were gathered from three large organizations by the Regenstrief Institute and the data sets are split into a US and an SI version. The US version is intended for users who favor reporting in mass units, whereas the SI set favors molar units. In addition to these internationally provided data sets, Germany’s Medical Informatics Initiative (MII) has produced a similar effort for national adoption and mapping guidance. As mentioned before, a set of 300 general laboratory test scenarios was envisioned to cover the most clinically relevant use cases. As a basis, LOINC-coded analyses were collected from five university laboratories in Germany and ranked according to their frequency of usage [22,28]. Based on their principal characteristics—mainly the analyte in the Component axis (e.g., glucose) and the specimen in the System axis (e.g., blood)—these codes were aggregated into the aforementioned 300 groups. For each group, one primary LOINC code was chosen that represents the respective test scenario in the most general or common way. Any other LOINC codes placed in the same group show slight differences in the axes *Property*, *System* and/or *Method* and are marked secondary. Like this, a varying amount of (currently) one to

fifteen possibilities per use case ensures the list's flexibility and accounts for the variability among laboratories and their favored procedures. All in all, the MII's Top 300 list includes a total of 935 LOINC codes at the moment.

In Table 3, we ranked the ten most common quantitative LOINC *Properties* of each data set in descending order by absolute occurrence. It is apparent that there is a substantial amount of overlap between each of the sets, and even if we only mapped these *Properties* to a base unit representation in UCUM, we would have already achieved a coverage of at least 80% for these typical real-world data.

Table 3. This table shows how the first ten quantitative *Properties* of the Top 2000 SI, US and Top 300 LOINC codes are distributed by absolute occurrence in descending order.

Rank	Top 2000 SI	Top 2000 US	Top 300 (MII)
1	Substance Concentration	Mass Concentration	Mass Concentration
2	Arbitrary Concentration	Arbitrary Concentration	Substance Concentration
3	Mass Concentration	Substance Concentration	Number Concentration
4	Number Fraction	Number Fraction	Number Fraction
5	Number Concentration	Number Concentration	Arbitrary Concentration
6	Mass Fraction	Mass Fraction	Mass Fraction
7	Dilution Factor (Titer)	Dilution Factor (Titer)	Catalytic Concentration
8	Substance Ratio	Substance Ratio	Time
9	Number per Area	Number per Area	Pressure (Partial)
10	Time	Time	Dilution Factor (Titer)

In the last step of the evaluation, we presented the initial version of LUMA to five participants of the laboratory network LADR. The evaluation took place as a video conference in which the capabilities of LUMA were demonstrated. For this purpose, we were provided with real-world data by the LADR that were mapped to UCUM during the presentation. The participants were able to ask questions and a discussion took place afterwards. Additionally, a questionnaire that dealt with UCUM-related mapping endeavors was handed to each participant.

- Do you have any prior experience regarding UCUM? (Purpose of UCUM, Context in which you used it or have seen it used, etc.)
 - In general, the involvement with the UCUM standard was theoretical at best, which is not surprising considering the current lack of strong standardization efforts regarding units of measure. The standardization of laboratory identifiers via the use of LOINC is only getting started in small steps through projects such as the ongoing Medical Informatics Initiative that enforces the mapping to LOINC for participating locations in Germany and the additional important step of also unifying the communicated units and measurements via UCUM is made optional.
- Are you aware of any tools or other means that could help you in creating appropriate mappings between proprietary units and UCUM?
 - The participants were not acquainted with any tools or other means for creating the mappings.
- Have you already created mappings between proprietary units and UCUM prior to this? If so, how did you find or formulate those mappings? If not, what was the biggest hurdle when trying to do so?
 - Currently, there were two aspects that majorly shaped the way proprietary units were handled. Seeing as there are no binding large-scale governmental efforts regarding the establishment of a unifying standard for communicating measurements, the problem of using proprietary units is a widespread issue that increases in magnitude the more work is put into interconnecting the healthcare system. With no standard to mediate between the different actors, a lot of effort is needed simply to maintain the conversion tables needed to communicate on a case-by-

case basis. Moreover, there is a lot of room for error during every step of the communication process. This effect is multiplied by a significant magnitude for laboratory networks such as LADR, which are in contact with an enormous amount of other locations and doctors that all have certain idiosyncrasies in reporting style to be accounted for.

In the discussion concerning LUMA itself, it became apparent that the participants saw the potential not only for using the UCUM standard, but also in our approach to the associated mapping efforts in such a streamlined way that minimizes the room for error as best as possible without overwhelming the inexperienced user. Displaying the example unit associated with a LOINC code in the official LOINC data set was found to be helpful to get an idea of an initial reference value to work with or deviate from. In this context, the participants suggested further emphasizing this aspect. Firstly, the example unit should be marked in the list of possible recommendations or be able to be mapped directly, and secondly, there was a suggestion to highlight completed mappings that deviated from the example unit. Initially only planned as an additional feature, the unit converter was of great interest to the participants currently working with extensive conversion tables. In this context, it would make sense to at least think about adding a functionality to our UCUM service that would specify the conversion factor between units. Currently, it is only used internally to calculate the final conversion value without being explicitly displayed itself. Without any widespread standardization efforts of UCUM, this could at least help generate more precise local conversion tables, especially for complex unit terms.

4. Discussion

Seeing as we are not aware of any solutions that aid users in their standardization endeavors regarding UCUM in the way that this project outlines, we cannot draw any direct comparisons. Users motivated to carry out such mapping ideas are currently faced with only two possibilities to do so, either by using the LOINC data set via SearchLOINC or RELMA or by making use of aggregated lists of UCUM units. By assessing the respective strengths and weaknesses of each approach, we drew conclusions regarding the design of this project. It was important to not only view it from a technical standpoint but also in terms of the obstacles an average user with little expertise would be faced with when trying to map proprietary units to UCUM.

4.1. The Evaluation

For our initial evaluation, we invited five participants of the LADR laboratory network to partake in a video conference where we demonstrated the capabilities of LUMA. Presenting the software in this way meant that our evaluation focused on the general premise of LUMA and not primarily on aspects of usability, although some input regarding that was generated nonetheless. It additionally highlighted the fact that the development of a web-based interface is the right way forward for future developments. Although one could bundle the software components and data in a way that does not rely on a FHIR terminology server and capable UCUM tool to exist as a requirement for using LUMA, it is a much more sensible approach to encapsulate all of it in a platform-agnostic way that is also easier to maintain due to having one central back-end to take care of. Additionally, although the participants were only loosely familiar with UCUM, we nonetheless inquired what would make adopting the standard more attractive on a user basis. One requirement was deemed to be very important. There was an interest in a verified and easy-to-use reference that would ideally also be able to be used in a standardized way within a digital context. Having worked with the UCUM standard extensively, there is something to be said about the accessibility as presented on the official web presence. The specification document is quite dense and very detailed, which makes sense considering the fact that its main purpose is to lay out the framework for software implementations. There is no doubt that end-users wanting or needing to acquaint themselves with the standard without the intention of implementing it would likely be intimidated by this. As of March 2022, the

official web presence is undergoing changes regarding its appearance and composition, so it remains to be seen what future developments are going to occur in this regard.

4.2. Current Approaches

Users wanting to map their proprietary units to UCUM currently have two ways to do so. The first solution relies on accessing the LOINC data set either via RELMA or the web-based SearchLOINC application. Here, users can look up specific LOINC codes in order to find information on them that ranges from a full description of all its axes to an example UCUM unit. The biggest drawback of this approach is that not all LOINC codes have an example unit associated with them. Secondly, the examples displayed do not necessarily fit the unique circumstances of each laboratory and mapping scenario. One also has to be mindful of the fact that not all given examples are inherently correct. Although smaller in the total amount of choices compared to the second solution, this approach is automated. By being able to look up specific LOINC codes, users are already pointed in the correct direction by drastically decreasing the possible search space.

The second approach that users may take into consideration is making use of aggregations of UCUM units as a reference. One such document pointed to on the LOINC and UCUM web presence consists of common UCUM units aggregated by Intermountain Healthcare, a joint effort of the National Library of Medicine and the Regenstrief Institute. It consists of 848 UCUM units commonly used in electronic reporting sourced from the raw units of more than 23 laboratory sources and the HL7 table of units. At the time of writing, the latest version 1.5 was published in June 2020 (<https://ucum.org/trac/wiki/adoption/common> (accessed on 29 April 2022)). The explicit goal of this effort was to aid users who felt intimidated by UCUM in standardizing their proprietary unit terms [10]. Users wanting to utilize resources of this kind have to do so by combing through the documents manually. There is no software tool or explicit connection to LOINC codes associated with them which could help automate the process. On the other hand, they are more extensive in scope compared to the LOINC data set.

Contributing to enhancing the Data Quality of LOINC

During our initial assessment of LOINC's 210 *Properties*, we took notice of the fact that not every example unit in the LOINC data set is an appropriate match for the *Property* it is associated with, thereby lowering the overall data quality. Our results can contribute to automatically finding such errors before they get included into future iterations of the standard; secondly, it can be used to sanitize the current data set. By matching the example units given against our base unit ground truth, we are able to verify if they represent a valid pairing.

4.3. Drawing Conclusions for Designing LUMA

In summary, we can conclude that existing approaches regarding the mapping of proprietary units to UCUM for laboratory testing are noticeably less automated and formalized than LUMA. We argue that the most important hurdle for more active mapping efforts lies in the lack of an official and explicit outline of how UCUM and LOINC relate to one another, and we proposed the idea of creating such a connection via formulating a UCUM base unit representation for LOINC's *Property* axis. Our approach improves upon the capabilities of the currently available methods in every way.

By turning the initially mentioned aggregations of UCUM units into FHIR ValueSets we leverage them for use within the LUMA framework. Although these lists are available in a digital format, they are not officially embedded within any software tools that aid users in their mapping efforts. Users have to rely on manually scanning them for a unit that might fit their intended use, and there is no way to involve LOINC codes to simplify the process as it lacks a direct association with the standard. A slightly more automated approach makes use of the LOINC data set via either RELMA or SearchLOINC, where users can look up specific LOINC codes. Although some LOINC codes are associated

with one or sometimes two example UCUM units, not all of them are. Our approach improves upon existing solutions, firstly by being at least as good as the LOINC data set itself meaning that if an example unit exists, LUMA highlights it separately. By additionally having a way to automatically compare these example units to the expected UCUM base units we postulated, there is a way to detect invalid entries. Moreover, LUMA not only provides the example units themselves, but it can also leverage existing lists of UCUM units in order to source recommendations from them. We demonstrated this by including aggregations of UCUM units into the framework which were not intended for this purpose without having to extensively modify them beforehand. Even though the usability aspect of LUMA was not the main focus of our evaluation with the five participants of the LADR, we nonetheless received feedback on two aspects that we want to incorporate in future iterations. Firstly, example units should be highlighted in our list of recommendations or be able to be mapped directly as a default option when possible. Secondly, if a user decides to deviate from the example units, this aspect should be reflected in the mapping table.

When designing the back-end for LUMA, we built upon locally pre-existing infrastructure in the form of the FHIR terminology server Ontoserver. In doing so, our idea was to create and make use of standardized FHIR resources that we would be able to share easily via its syndication mechanism and that might be useful even outside the scope of this application. That being said, the framework itself—meaning a FHIR terminology server, a UCUM service and the Controller handling the machine-to-machine communication between them—is not very portable. In theory, it is possible to set up a dedicated service with the same capabilities, but due to the modular approach we took when initially sketching ideas on how to design our solution, it is much easier to encapsulate the back-end via numerous possible views. For this purpose, we created a graphical user interface for internal testing and are now shifting our efforts towards a web interface that can easily be exposed and flexibly accessed even via mobile devices capable of connecting to the web. Our intended target audience was people with only very little knowledge about the intricacies of LOINC and UCUM. Regarding the quality of mappings created, we are faced with the inherent issue of not knowing the exact semantics of the proprietary units users are trying to map to UCUM. As the user is the final bottleneck in this situation, we tried to minimize the amount of unsound associations that could be created. We therefore opted to heavily streamline LUMA's capabilities by offering users to search for a specific LOINC code and to carry out unit conversions based on the mappings created that way. Additionally, we made sure to provide users with a written-out human-readable description of each UCUM unit provided and to filter flawed example units and unfitting recommendations based on the *Property* of a specific LOINC code. Due to the fact that the set of valid UCUM units is infinitely big, we only provide recommendations based on a previously aggregated finite set of UCUM units. Although this does not mean it can not be added onto, we believe that the current amount of units should be sufficient for initial usage.

Moreover, there is another idea that could increase the flexibility of the curated recommendations even more. Each unit symbol in UCUM is associated with numerous additional information. Units can be grouped by classes such as “SI-Units” or “US-Lengths”, for example, but they may also be grouped by property. This property (not to be confused with LOINC's *Property* axis) allows one to group units by their physical makeup. Second, minute and hour obviously describe time whereas gram, ton and pound are masses. Another way of describing the relationship between LOINC's “Mass Rate” *Property* and UCUM might therefore be described as “[UCUM unit that is of the property ‘mass’] / [UCUM unit that is of the property ‘time’]”. What this would enable us to do is to provide users with the ability to make controlled modifications to the curated UCUM recommendations within LUMA. Each use case and laboratory setting is different, and while the ValueSet from which the recommendations are sourced from is quite extensive, it does not account for all possible related units, metric prefixes, specific scalars or numerous combinations of the three in the form of “ug/(12.h)” and “mg/(30.min)”, for example. Our flexible UCUM implementation allows us to analyze the specific operands of each of these terms in order

to provide information on how they may be altered without breaking compatibility with their base unit representation.

5. Conclusions

In creating a formalized association between LOINC and UCUM, we established an important milestone for dealing with problems related to the standardization of units of measure used in laboratory testing. Of the 176 quantitative LOINC properties, we were able to match 159 of them to a UCUM base unit representation. Building upon this, we applied this knowledge in the context of a software framework represented by the graphical user interface LUMA. We intended for it to lower the barriers of entry for standardization efforts concerning UCUM in laboratory settings. LUMA was therefore specifically meant to aid users inexperienced in either standard to map their proprietary units to UCUM via the use of LOINC codes. The drawbacks of currently available approaches with similar intentions informed the design of our project, and we were able to fully integrate the example units of the LOINC data set and various existing aggregations of UCUM units into our solution. Additionally, we outlined how the formalization of the connection between LOINC and UCUM could be leveraged to increase the data quality of the LOINC data set by verifying existing and future example unit entries. The main takeaways of the user evaluation were that future developments of LUMA should be focused on a web interface and that the ability to perform controlled unit conversions is not a peripheral feature but actually very much a point of interest. In this way, the logical next step comes down to presenting our findings to the Regenstrief Institute. Not only do we need to report our discoveries concerning the data set but we would also be very interested in feedback regarding our formalization efforts. Ideally, our results could become part of the official LOINC canon.

Supplementary Materials: The following are available online at <https://www.mdpi.com/article/10.3390/app12125848/s1>.

Author Contributions: Conceptualization, C.D., K.V. and J.I.; methodology, K.V. and C.D.; software, K.V.; formal analysis, K.V.; investigation, K.V.; resources, C.D., J.I., J.K. and C.C.; data curation, K.V. and C.D.; writing—original draft preparation, K.V.; writing—review and editing, C.D.; visualization, K.V.; supervision, J.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the German Federal Ministry of Education and Research (BMBF) as part of the Medical Informatics Initiative Germany, Grand ID 01ZZ1802Z.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FHIR	HL7 Fast Healthcare Interoperability Resources;
HL7	Health Level 7 International;
LOINC	Logical Observation Identifiers Names and Codes;
LUMA	LOINC to UCUM Mapping Assistant;
RELMA	Regenstrief LOINC Mapping Assistant;
REST	Representational state transfer;
UCUM	Unified Code for Units of Measure.

References

1. Pommerening, K.; Deserno, T.; Ingenerf, J.; Lenz, R.; Schmücker, P. Der Impact der Medizinischen Informatik. *Informatik-Spektrum* **2014**, *38*, 347–369. [CrossRef]
2. Hauser, R.; Quine, D.; Ryder, A.; Campbell, S. Unit conversions between LOINC codes. *J. Am. Med. Inform. Assoc. JAMIA* **2017**, *25*, 192–196. [CrossRef] [PubMed]
3. Regenstrief Institute Inc. About RELMA. Available online: <https://loinc.org/relma/> (accessed on 30 April 2022).
4. Schadow, G.; McDonald, C.J. The Unified Code for Units of Measure. 2016. Available online: <https://ucum.org/trac> (accessed on 30 April 2022).
5. Semler, S.C.; Wissing, F.; Heyder, R. German medical informatics initiative. *Methods Inf. Med.* **2018**, *57*, e50–e56. [CrossRef]
6. Drenkhahn, C.; Ingenerf, J. The LOINC Content Model and Its Limitations of Usage in the Laboratory Domain. *Stud. Health Technol. Inform.* **2020**, *270*, 437–442.
7. Rajput, A.M.; Ballout, S.; Drenkhahn, C. Standardizing the Unit of Measurements in LOINC-Coded Laboratory Tests Can Significantly Improve Semantic Interoperability. In *Integrated Citizen Centered Digital Health and Social Care*; IOS Press: Amsterdam, The Netherlands, 2020; pp. 234–235.
8. Breuer, H.W.M. Laborwerte: Ohne Umrechnungstabelle läuft nichts. *Dtsch. Ärzteblatt* **2004**, *101*, A24.
9. Regenstrief Institute Inc. About SearchLOINC. Available online: <https://loinc.org/search-app/> (accessed on 30 April 2022).
10. Regenstrief Institute Inc. Common UCUM Units. Available online: <https://loinc.org/usage/units/> (accessed on 30 April 2022).
11. Health Level 7. Valueset-Ucum-Common. 2019. Available online: <https://www.hl7.org/fhir/valueset-ucum-common.html> (accessed on 30 April 2022).
12. McDonald, C.; Huff, S.; Deckard, J.; Armson, S.; Abhyankar, S.; Vreeman, D.J. *LOINC Users' Guide*; Regenstrief Institute: Indianapolis, IN, USA, 2017; pp. 19–51.
13. McDonald, C.J.; Huff, S.; Suico, J.; Hill, G.; Leavelle, D.; Aller, R.; Forrey, A.; Mercer, K.; DeMoor, G.; Hook, J.; et al. LOINC, a Universal Standard for Identifying Laboratory Observations: A 5-Year Update. *Clin. Chem.* **2003**, *49*, 624–633. [CrossRef]
14. Bodenreider, O.; Cornet, R.; Vreeman, D. Recent Developments in Clinical Terminologies—SNOMED CT, LOINC, and RxNorm. *Yearb. Med. Inform.* **2018**, *27*, 129–139. [CrossRef] [PubMed]
15. Schadow, G.; McDonald, C.J. Unified Code for Units of Measure Specification. 2015. Available online: <https://ucum.org/ucum.html> (accessed on 30 April 2022).
16. Schadow, G.; McDonald, C.J.; Suico, J.; Föhring, U.; Tolxdorff, T. Units of Measure in Clinical Information Systems. *J. Am. Med. Inform. Assoc. JAMIA* **1999**, *6*, 151–162. [CrossRef]
17. Nizamov, S. *Unofficial Developer's Guide to FHIR on Mirth Connect*; Shamil Publishing, 2016. pp. 19–20. Available online: <http://shamilpublishing.com/> (accessed on 7 June 2022).
18. Terminology Module. 2019. Available online: <https://www.hl7.org/fhir/terminology-module.html> (accessed on 30 April 2022).
19. HL7 FHIR Resource CodeSystem—Content. 2019. Available online: <https://www.hl7.org/fhir/codesystem.html> (accessed on 30 April 2022).
20. HL7 FHIR Resource ValueSet—Content. 2019. Available online: <https://www.hl7.org/fhir/parameters.html> (accessed on 30 April 2022).
21. HL7 FHIR Resource ConceptMap—Content. 2019. Available online: <https://www.hl7.org/fhir/conceptmap.html> (accessed on 30 April 2022).
22. Semler, S.C. LOINC: Origin, development of and perspectives for medical research and biobanking—20 years on the way to implementation in Germany. *J. Lab. Med.* **2019**, *43*, 359–382. [CrossRef]
23. HL7 FHIR Resource Parameters—Content. 2019. Available online: <https://www.hl7.org/fhir/valueset.html> (accessed on 30 April 2022).
24. Grune, D.; Jacobs, C. *Parsing Techniques: A Practical Guide*; Monographs in Computer Science; Springer: New York, NY, USA, 2007; p. 1.
25. Hamblin, C.L. Translation to and from Polish Notation. *Comput. J.* **1962**, *5*, 210–213. [CrossRef]
26. Dijkstra, E.W. *Algol 60 Translation: An Algol 60 Translator for the x1 and Making a Translator for Algol 60*; Technical Report 35; Mathematisch Centrum: Amsterdam, The Netherlands, 1961.
27. Metke-Jimenez, A.; Steel, J.; Hansen, D.; Lawley, M. Ontoserver: A syndicated terminology server. *J. Biomed. Semant.* **2018**, *9*, 1–10. [CrossRef] [PubMed]
28. Bietenbeck, A. Der TOP 300 Datensatz. 2018. Available online: https://www.medizininformatik-initiative.de/sites/default/files/2020-07/TOP%20300%20Datensatz_Bietenbeck.pdf (accessed on 30 April 2022).