

Towards Automatic HL7-RIM and Relational Database Mapping

Shagufta Umer, Muhammad Afzal, Maqbool Hussain, Hafiz Farooq Ahmad, Khalid Latif
NUST School of Electrical Engineering and Computer Science (SEECS), Pakistan
{shagufta.umer, muhammad.afzal, maqbool.hussain, drfarooq, khalid.latif}@seecs.edu.pk

Abstract- Health Level Seven (HL7) is an emerging standard in the healthcare industry. HL7 version 3.0 (v3) targets to achieve semantic interoperability among the healthcare systems through HL7 Reference Information Model (RIM). In any healthcare system, information is ultimately stored in the database, which requires mapping of database to HL7 RIM. A key bottleneck in building HL7 systems is the laborious manual construction of mappings between the HL7 RIM and the local clinical databases. HL7 v3 messages, composed of classes, attributes and association identifiers, can be parsed to any relational database composite of tables, attributes and associated identifiers. Manually creating mappings is extremely tedious and error-prone. We propose an approach to automatically find such mappings. The technique exploits domain integrity constraints and user feed-back and performs simple as well as complex mappings. It extends our previous work utilizing different matching strategies and implements significant improvements and offers a comprehensive infrastructure to solve RIM to schema mapping problems.

I. INTRODUCTION

Healthcare has always been at the cutting edge of technology for patient care, yet little efforts are seen to support healthcare practices using ICT (Information and Communication) systems. Healthcare industry has made little use of Information and Communication systems like record keeping, but in real it can play an essential role in the formation of care chains between the healthcare sectors. Every country is currently engaged in making healthcare information at hands all the time. Some years ago, there was typically little need for clinical applications to exchange data; today, the need for applications to share clinical data is critical. The explosion in the number of clinical applications and the national push for a centralized electronic health record are only two factors driving the requirement for a common language between applications. Today's need is to address the issues of prompt availability of information at the point of care, reducing medical errors and avoiding unnecessary medical procedures. And through the use of Information and Communication Systems, we can achieve the said objectives.

In the last few decades a lot of efforts in provision of standard for interoperable communication are being made. Health Level Seven (HL7) is the leading healthcare standard, addressing the exchange of information among different healthcare organizations [1]. HL7 v3 adopts an Object Oriented (OO) approach using Unified Modeling Language (UML) principles. HL7 v3 strives to achieve semantic interoperability through introducing Reference Information Model (RIM) [2]. RIM is the source of all the information subjects used in HL7 specification in the form of classes,

attributes and relationships. The purpose of a RIM is to share consistent meaning beyond a local context.

Healthcare organizations mostly store their data in traditional relational databases. Relational databases store the information in the form of tables and its constituent fields. The data transfer between the systems in a meaningful manner is a critical aspect in the information sharing. RIM is an abstract model encompassing healthcare domains specified by HL7 specifications. RIM is a comprehensive UML model representing concepts in healthcare domain through classes containing their attributes and connected with associations. Therefore, there exist either one-to-one or one-to-many correspondences of fields mapping between clinical data model and RIM model. HL7 RIM and database mapping is an appealing but a complex problem.

In this paper, we are addressing the issues of aligning local clinical relational databases with RIM. Solving such mapping problems is of key importance to achieve interoperability and data integration in numerous application domains. To reduce the manual efforts required, this technique is developed to solve the mapping problem in a semi-automatic manner. The proposed approach typically exploits metadata including element names, data types and structural properties in the mapping process. Though RIM is the foundation of healthcare interoperability and covers a vast domain but there are problems with the RIM documentation which is not only disastrously unclear, and poorly integrated with those other parts of the V3 documentation for which the RIM itself is designed to serve as backbone [3]. The manual tracing out the classes and attributes not only requires lot of efforts in understanding the RIM concepts but also identifying the most appropriate match in the local clinical database

In summary, the proposed technique loads the schema of the clinical database into memory and then it automatically detects the mappings between clinical databases and the RIM by applying different mapping strategies. It reduces the overhead of manual mappings which is error-prone and time consuming.

The major challenges in RIM and clinical databases are as follows:

- RIM encompassing a vast domain is therefore multifarious and complex.
- Databases are highly heterogeneous with respect to the data models, the data schemas, the query languages they support, and the terminologies they recognize [4].

- Different databases having same real world semantics may have different schema structures.
- There is currently no tool available to partially automate the RIM and database schema mapping process.

II. PROPOSED ARCHITECTURE

Based on our previous work [8] of mapping between relational database and HL7 RIM, we enhanced the proposed architecture by refining the existing components as well as adding new components and mapping strategies which are discussed below in detail.

1) Mapping Knowledge Repository:

Mapping Knowledge Repository is a database of the possible mappings found in the clinical laboratory databases. This is done by the analysis of functioning laboratory databases and is kept to be extended with more mapping knowledge by community involvement. In the following section, we elaborate the structure and the major functions of the knowledge repository.

a. Structure of the Mapping Knowledge Repository:

The Mapping Knowledge Repository provides two essential functions, one is to store mapping knowledge and other is to accommodate various input/output functions to let users view, edit, and create new mappings. We selected XML due to flexibility and extendibility; we need to extend the Mapping Knowledge Repository during evolution. An XML repository retrieval system, could receive new documents containing new elements without breaking the system [9]. The Mapping Knowledge Repository also provides user friendly interface for browsing and adding the mapping knowledge.

b. Evolution of Mapping Knowledge Repository:

The evolution process of the Mapping Knowledge Repository involves the community through a registration process. The registered users are allowed to update the Mapping Knowledge Repository by adding new matching columns and the tables found in their clinical databases. Besides this, the Mapping Knowledge Repository automatically gets updated while the tool is performing mappings i.e. when the user edits or re-maps as explained in the 'Constraint Handler' section. The evolution of the Mapping Knowledge Repository will improve its matching results. The user can request for the updates made in the Mapping Knowledge Repository.

Mapping Knowledge Repository is flourished with RIM content representation and navigation. The repository also maintains visual interfaces to let the users create, edit, and use the mappings. As the system is being continuously refined, we plan to conduct tests and evaluations of the mappings in knowledge repository with real users' participation. We expect that the user evaluation will provide valuable input for the enhancement of the Mapping Knowledge Repository.

2) Schema Loader:

Schema Loader loads the data model of the target clinical database and the corresponding MIF as discussed earlier. Loading schema is a prerequisite step for mapping.

3) Mapping Controller:

This component is the core engine of the process. For every column in the database, it searches the most appropriate match through applying pattern matching algorithms and binds it to the RMIM accordingly. This component uses Mapping Knowledge Repository for its processing as shown in Figure 1.

- **Data Types Handler:** Most of the RIM data types are complex and different from that of RDBMS supported data types like ED (Encoded Data), CS (Coded String) etc. When local clinical data types are to be mapped with HL7, semantics are carefully considered to be preserved which is the ultimate goal of HL7 v3 for interoperability. For example, "Patient Name" in local clinical databases, is normally stored as a single column with corresponding data type varchar. HL7 on the other side adds semantic to handle the complexity by defining the person name parts like given name, family name, prefix etc having BAG data type. Data Type Handler addresses the data types' complexity by mapping the components of the name in the database to the defined parts in HL7 Name. Similarly, the other data types are handled using well-defined strategies.
- **Name Handler:** This tries to find out the corresponding patterns of the table/ column name in the RMIM for appropriate mapping. Different pattern matching rules are defined against which we have attached confidence values. These rules show that how much accurate is the mapping rule. If the accuracy is high than we perform the mappings confidently. In case the confidence value is low then we confirm the matching from the user to improve its accuracy. The matching rules and confidence values are briefly explained as follows:
 - Exact Match Rule:** We have assigned confidence value 1 for this rule. It finds the exact pattern in the Mapping Knowledge Repository i.e. For example, in the Knowledge Repository; we have mappings for patient name as *name*, *pname*, *patname* etc. During finding mappings, if an element is found in the database as *pname* or *patname* then this will be the exact match.
 - Pattern Match Rule:** This rule trace out pattern in the database element, i.e. the complete pattern is found in the element plus some other characters, and its confidence value is 2. For example, in the Knowledge Repository, we have mappings for patient name as *name*, *pname*, *patname* etc. Now while performing mappings, an element is found

in the database as *patientname*, which contains the pattern ‘*name*’ in it plus patient.

iii. **Synonyms Match Rule:** This rule is about synonym match with the database element and its confidence value is 3. For example in the Knowledge Repository, we have mappings for test description as desc, description, comments etc. Now while performing mappings, an element is found in the database as summary then this is the synonym match.

Sub-Pattern Match Rule: This rule further trace out a subset of the exact pattern of the stored mappings in the Knowledge Repository to the database element and its confidence value is 4. For example, in the Knowledge Repository we have mappings for patient name as *name*, *pname*, *patname* etc. Now while performing mappings, an element is found in the database as *p_nam* then this will be the sub-pattern match.

iv. **Locality Handler:** The Principle of Locality refers to use the data element within relatively close storage locations [10]. The location of the column in a particular table helps in the identification of the particular class in the RIM. The tool displays the corresponding attributes to the user with well- explanation so that it can help in identifying the mappings. This strategy is explained in detail by quoting an example of an attribute. For example, for a column named as age, tool cannot identify its corresponding

attribute, then it looks into the Parent table i.e. Patient. This helps us in identifying all possible mappings for the said attribute in the Person Class. Finally, the user selects the best suited mapping for the element.

4) **Constraint Handler:** This component improves the mapping accuracy in situation when the confidence value of the mapping is low.

- If two or more mappings are identified for a single concept then ask the user to decide which mapping is most appropriate.
- If any mapping is wrongly identified then the user can request to remap the particular element.

5) **Validation Process:** Once the user is finished with mappings, the next step is to validate the mapping. During validation, the tool displays any missing mapping to the user. The user has the choice to remap or save the mappings. The user can edit the mappings by selecting the mapped element and then request either to delete the mapping or re-map the element. On selecting the Delete option, the selected mapping is deleted and on selection of the “Re-Map” option, the tool deselects the mappings and search again for all possible best mappings. The user can then select the option which best suits the requirement. These new mappings specified by the user not only fulfill the user requirement, but also helps in updating the Knowledge Repository. The tool itself saves the new mappings suggested by the user during the mapping process, thus evolving the Mapping Knowledge Repository.

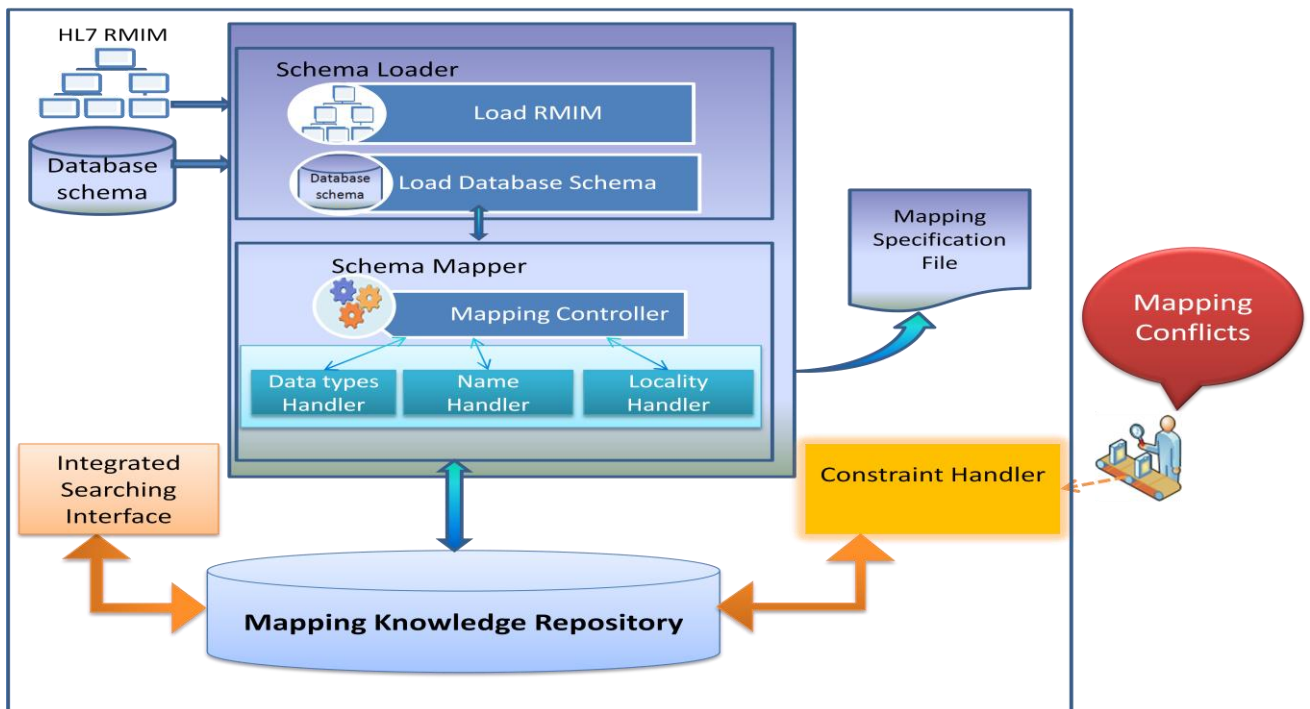


Figure 1: Proposed Architecture

III. ARCHITECTURE SCENARIO

In this section, the working process is elaborated with the help of real scenario. First of all, the target clinical database is loaded along with the corresponding RMIM. All the tables and the corresponding columns are on one panel and RMIM object model on the other panel. After the loading process, all of the tables and corresponding columns of the schema are scanned one by one. In our scenario, the clinical database table Patient is the first entity to be mapped as shown in Figure 2. Next the Mapping Controller searches the most appropriate class in RMIM. Here, Mapping Controller makes use of matching techniques to accomplish the task. *Name Handler* is responsible for the identification of the corresponding match through pattern matching algorithms and confidence values. It exploits the database element name and tries to find out the appropriate mapping for it. In the Patient Table, the corresponding match for Name attribute is found and mapped accordingly. Name column is mapped to the name attribute in the Person Class as shown in Figure 2. The next concept is about Data Type Handler; the name has data type Entity Name Part (ENXP) [6]. Typical name parts for person names are given names, family names, titles, etc, but there are lots of variations across the world. To handle this variation, we have provided flexibility in the architecture which is discussed in Section V.

IV. MAPPING VARIATIONS ACROSS THE WORLD

People only have one name, but that name may be written down in many different ways. The final dimension of complexity occurs when names from many cultures are to be handled. Different countries have different cultures for naming. This complexity is handled by collecting variety of possible naming conventions around the world [10] and allowing the user to select it as per its requirement.

We have used variety of naming formats to address this naming issue; few of them are summarized in Table 1. The default settings in the tool for the name are <Given Name><Family Name>, which is most frequently used convention. However, the tool facilitates the user to reset the naming convention using drop down menu. HL7 name parts are standardized as described in Table 2. The tool handles different naming formats and associated mappings

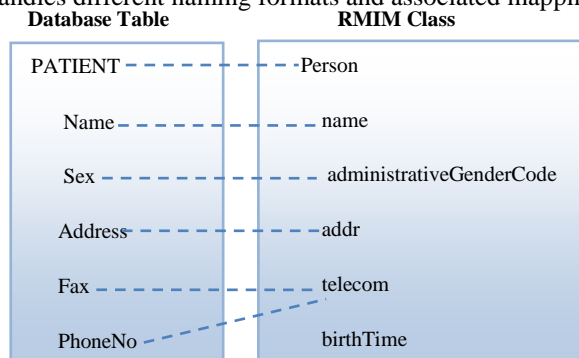


Figure 2: Mapping Workflow Example

TABLE 1: NAMING CONVENTION AROUND THE WORLD

Country	Naming Convention
Japan	<family name > <given name>
Germany	<first name> <family name>
India/Pakistan	<first name> <middle name (Optional)>< last name>
Assyria	Personal Name> <Father's Personal Name> <Grandfather's Personal Name>
China	<Family Name> <Generational Name> <Given Name>
Indonesia	<Given Name><Clan Name>
Hungry	<Family Name> <First Given Name> <Second Given Name>
Fiji	<Baptismal Name> <Given Name> <Mother's Family Name> <Father's Family Name>
And others.....	

to the HL7 name parts. The user selects the naming convention as per the requirement only once. The data is mapped as per the selection, thus achieving semantics in the name. If the user selects <Given Name> <Family Name>, then this will be mapped to Given Name and Family Name in the HL7 name parts as shown in Figure 3.

V. RELATED WORK

Anhoi Doan et al [11] work focuses on building data integration systems that provide a uniform query interface to the sources. The problem in building such systems has been the laborious manual construction of semantic mappings between the query interface and the source schemas. They have proposed a multi-strategy learning approach to automatically find such mappings. The approach applies multiple learner modules, where each module exploits a different type of information either in the schemas of the sources or in their data. It then combines the predictions of the modules using a meta-learner. They describe the LSD (Learning to Match Schemas of Data Sources) system, which employs this approach to semantic mappings. They have tested LSD experimentally on several real-world domains. The experiments show that LSD semantic mappings have a high degree of accuracy. It

TABLE 1: HL7 NAME PARTS

Name Part	Description
Family Name	This is the name that links to the genealogy. In some cultures (e.g. Eritrea) the family name of a son is the first name of his father.
Given Name	This is the given name, don't call it "first name" since this given names do not always come first
Prefix	A prefix has a strong association to the immediately following name part. Note that prefixes can be inverted.
Suffix	A suffix has a strong association to the immediately preceding name part. Suffices cannot be inverted.
Delimiter	A delimiter has no meaning other than being literally printed in this name representation.

is an effective framework strategy that could be used for any domain. It helped in polishing our efforts in making mapping strategies for RIM and clinical schema mappings.

caAdapter[12] is an open source tool set developed by National Cancer Institute US. It aims to facilitate data mapping and transformation among various kinds of data sources. Using caAdapter, the user can perform object model to data model mapping through drag & drop facility. Classes are mapped to the corresponding tables, attributes are mapped to the corresponding columns and the object relationships are mapped to table relationships by the user. caAdapter provides graphical user interface for mapping without automation. Its mapping service requires human intervention for manually tracing out all attributes in RIM, which is laborious and time consuming task.

VI. DISCUSSION

RIM and clinical schema mappings would be helpful in sharing and exchange of the information among different healthcare systems and making prompt decisions. People don't adopt new standards due to the fact that they are not easily integrated with the existing infrastructure of the organization. This tool will help in bridging the existing relational databases with the HL7 messaging. Thus development of such automation tools will serve in addressing such kind of issues. Majority healthcare organizations around the world are not using international standards for information exchange and interoperability rather using either ad hoc ways like faxes, mails and emails or national standards produced by specific international standards country/region. If, for example, they are convinced to use like HL7, still they face number of issues due to some underlined facts i.e. database and application are not in compliance to standard. These non-compliances lead to consideration of a complex and challenging activity called standard-to-database mapping. Recently, HL7 v3 is getting more popular due to the RIM. Although RIM is credible, clear, comprehensive, concise, and consistent; but its understanding and mapping with clinical schemas require a thorough and deep understanding of each and every concept. Similarly, clinical schemas are developed having no single structure and representation. So mapping of RIM concepts and database schema always requires not only studying the RIM concepts but also the individual database schema understanding. For automating the mapping, we need to analyze multiple databases in order to create comprehensive knowledge base. The knowledge base must have enough information require to, at least, partially automating the mapping rather as a whole, which is practically impossible.

VII. CONCLUSION AND FUTURE DIRECTION

The paper presents the proposed architecture for dynamic mapping of HL7 RIM to clinical schema. We have discussed the components involved in automating the RIM and database schema mappings. This scheme will reduce

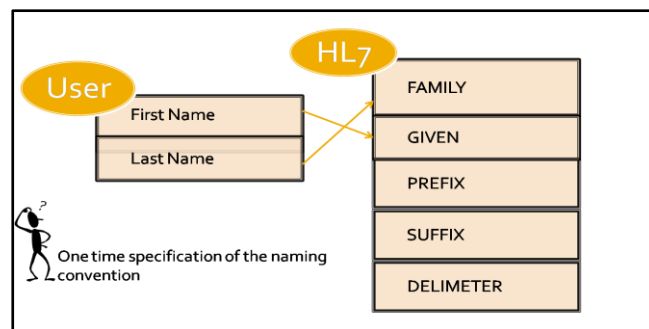


Figure 3: Naming Convention Support

the overhead of manual mapping the local clinical schemas to RIM. The mapping strategies are intelligently handled to make the system robust. The evolution of the tool with the time will help in serving the community more effectively by providing maximum number of mappings.

The future work involves making the proposed scheme intelligent enough in diagnosing the RIM to schema mapping knowledge through ontology based mapping. We can extract any database schema and build automated ontology of database. This will overcome the first step of analyzing and storing the mapping knowledge in advance and automate the generation of Knowledge Base. Later on, we can use this ontology for the dynamic RIM to clinical schema mapping. Further, this work will be exposed as web-service so that community can benefit from this tool.

ACKNOWLEDGMENT

This work is part of Health Life Horizon Project (HLH), funded by ICT R&D Ministry of IT, Pakistan. More details of this project can be seen from <http://hl7.seecs.edu.pk>.

REFERENCES

- [1] Health Level 7, <http://www.hl7.org>, June 2008.
- [2] HL7 Reference Information Model, ANSI/HL7 V3 RIM, R1-2003, 12-3- 2003.
- [3] Berry Smith, Werner Ceuster, "HL7 RIM: An Incoherent Standard", Studies in Health Technology and Informatics, 133-138, 2006.
- [4] Walter Sujansky, "Heterogeneous Database Integration in biomedicine", Journal of Biomedical Informatics, 285-298, 2001.
- [5] "Laboratory Domain", HL7 V3 Lab, R1, Last Ballot: January 2008.
- [6] "Laboratory Domain", HL7 UK V3 Lab, R1, Affiliate Ballot: May 2008.
<http://www.hl7.org/v3ballot2008JAN/html/help/V3guide/v3guide.htm>
- [7] The HL7 MIF - Model Interchange Format, http://www.ringholm.de/docs/03060_en_HL7_MIF.htm, June 2009
- [8] Shagufta Umer, Muhammad Afzal, Maqbool Hussain, Hafiz Farooq Ahmad, Khalid Latif, "Design and Implementation of an Automation Tool for HL7 RIM-to-Relational Database Mapping", 10th International HL7 Interoperability Conference (IHIC), pp 111-116, May 8-11, 2009, Kyoto Japan.
- [9] XML database, <http://lists.xml.org/archives/xml-dev/200010/msg00211.html>, June 2009.
- [10] Wikipedia, <http://en.wikipedia.org>, June 2009.
- [11] Anhai Doan, Pedro Domingos, Alon Halev, "Learning to Match Schemas of Data Sources : A multi-strategy Approach". Machine Learning, Volume 50, Number 3, pp. 279-301(23), March 2003.
- [12] caAdapter, <https://cabig.nci.nih.gov/tools/caAdapter>, January 2009.
- [13] caAdapter Model Mapping Service Hands-On Training, November, 2007.