Computing (2020) 102:2409–2444
https://doi.org/10.1007/s00607-020-00837-2

REGULAR PAPER

Ubiquitous Health Profile (UHPr): a big data curation platform for supporting health data interoperability

Fahad Ahmed Satti[1] · Taqdir Ali[2] · Jamil Hussain[1] · Wajahat Ali Khan[3] · Asad Masood Khattak[4] · Sungyoung Lee[1]

Received: 29 September 2019 / Accepted: 31 July 2020 / Published online: 19 August 2020

Abstract

The lack of Interoperable healthcare data presents a major challenge, towards achieving ubiquitous health care. The plethora of diverse medical standards, rather than common standards, is widening the gap of interoperability. While many organizations are working towards a standardized solution, there is a need for an alternate strategy, which can intelligently mediate amongst a variety of medical systems, not complying with any mainstream healthcare standards while utilizing the benefits of several standard merging initiates, to eventually create digital health personas. The existence and efficiency of such a platform is dependent upon the underlying storage and processing engine, which can acquire, manage and retrieve the relevant medical data. In this paper, we present the Ubiquitous Health Profile (UHPr), a multi-dimensional data storage solution in a semi-structured data curation engine, which provides foundational support for archiving heterogeneous medical data and achieving partial data interoperability in the healthcare domain. Additionally, we present the evaluation results of this proposed platform in terms of its timeliness, accuracy, and scalability. Our results indicate that the UHPr is able to retrieve an error free comprehensive medical profile of a single patient, from a set of slightly over 116.5 million serialized medical fragments for 390,101 patients while maintaining a good scalablity ratio between amount of data and its retrieval speed.

Keywords Big data · Healthcare information systems · Data curation · Data interoperability

Mathematics Subject Classification 68U35 · 68P20

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00607-020-00837-2) contains supplementary material, which is available to authorized users.

✉ Sungyoung Lee
sylee@oslab.khu.ac.kr

Extended author information available on the last page of the article

1 Introduction

In the last decade, the digital healthcare space has witnessed a rapid technological expansion, which has led to the development and deployment of a plethora of policies, software and devices [45]. As a result, the quality and quantity of healthcare delivery, in terms of diagnostics, treatment, and follow-up has greatly improved [47,74]. Additionally, supplementary healthcare sources, such as whole-genome sequencing[73], precision medicine [57], Clinical Practice Guidelines (CPGs) [37], and medical Internet of Things (IoT), and others have added new dimensions, to medical data. Today, healthcare data is characterized [40] by its large Volume (number of patients, size of patient data, additional information), Velocity (production rate, which can range from seldom produced non-streaming data to streaming data from medical IoT, like continuous glucose monitor), Veracity (different quality), Variety (formal and/or non-formal standards), and Value (insights).

Consequently, new challenges have emerged in the domain of healthcare, including lack of interoperability, globalization, collaborative capacity gap, tele-medicine, and ubiquitous healthcare [24]. The scale and scope of these challenges, has pushed beyond the scope of traditional data mining and integration techniques. Expert driven solutions are no longer feasible, while machine learning approaches are not mature enough to guarantee complete conversions, every single time. Numerous endeavors have focused on resolving different aspect of the interoperability problem. Our review indicates that most interoperability tools and techniques, work under the assumption, that some form of standards are already in use by the participating medical platforms. On the other hand, the healthcare domain has many formal and even a larger number of non-formal standards(custom data representation, and exchange formats which are at-most used at institutional or regional levels) catering to different aspect of the interoperability problem. As a result, the technical aspect of the interoperability problem, can only be solved by applying semantic reconciliation at data, knowledge and process level. Current solutions are focusing on the use of two distinct approaches; a more formal and slower process of standard integration (to merge commonalities and novelties of numerous standards, producing only one universally accepted standard) and mediation based approaches (bridge the gap between all heterogeneous standards for a quick and dirty solution).

Our approach towards resolving this problem, is based on the Ubiquitous Health Platform (the platform's original acronym UHP is not being used in this paper to avoid any confusion with the acronym for the presented research work, UHPr), which is shown in Fig. 1. A key part of this large platform is the semi-structured data container, the Ubiquitous Health Profile (UHPr) which is used for storing, integrating and exchanging, multidimensional healthcare data. Going beyond the traditional longitudinal Electronic Health Records (EHR), the UHPr, represents a multi-dimensional data structure, which combines heterogeneous medical data, using a semi-structured storage engine. Utilizing, data produced from a variety of sources, ranging from very structured form (e.g. HMIS) to unstructured streaming data (Medical IoT), the UHPr is able to store and accurately retrieve archived patient information. While detailed description of the platform shown in Fig. 1 is outside the scope of this paper, it is important to understand the need for our presented methodology. As shown in

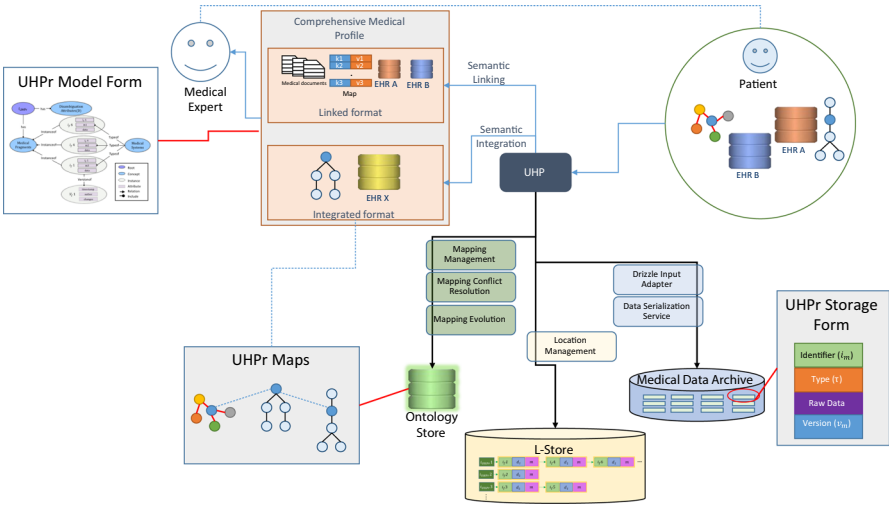


Fig. 1 Semantic reconciliation using the Ubiquitous Health Platform

Fig. 1, the Ubiquitous Health Platform acts as a bridge between a patient and medical experts/systems. On the one side of this bridge lies the big data archive service which consumes healthcare data from various sources, extracts meta information related to each patient, serializes the input to strip away its schema and converts it into a relatively flat/denormalized data structure, which is finally stored in a semi-structure form (UHPr Storage). The other side of this bridge is occupied by service consumers, which receive the healthcare data in graph form (UHPr model), containing either semantically linked or semantically integrated comprehensive medical profile of a patient (e.g. in Fig. 1, EHR A and B are transformed into EHR X, using semantic integration). The UHPr supports this bridge by providing data curation services for storing healthcare data, creating and storing the semantic reconciliation knowledge base (refered to as UHPr Maps), applying UHPr Maps to semantically match the attributes of each participating medical fragments, and finally to produce a semantically enriched version of the comprehensive medical profile of the patient. In this paper, we present the UHPr methodology, implementation and evaluation in terms of its timeliness, accuracy, and scalability. The contribution of this research work and paper is threefold. Firstly, we present the design and implementation of the UHPr engine along with its associated storage, management, and retrieval process as the major contribution. Secondly from our review of the current literature and practices, an amalgamation of Big Data curation technologies with healthcare data integration is a novel initiative, with very little implementation and evaluation proof. While we have previously introduced the concept behind such an amalgamation in [70], the implementation was very limited and not scalable. In this paper, we have extended that idea to formally prove it and provide results and evaluation of our system in detail. Thirdly, we describe the three main challenges of our approach (Patient Identification, Data Verification, and Security and Privacy). In the next sections we will discuss the following:

– Section 2 identifies some of the motivating factors behind the need for UHPr, including the requirement for a NoSQL based Big Data storage and processing engine, necessity for the semantic interoperability, especially taking into account the semantic matching and semantic integration aspects of the same, and data interoperability.
– Section 3 introduces the Ubiquitous Health Profile by presenting its novelty, theoretical foundations, and prototype implementation.
– In Sect. 4, we discuss the experimental setup and results of our evaluation towards proving the timeliness, scalability, and accuracy of the Ubiquitous Health Profile.
– Section 5 discusses some of the challenges and limitations, identified during our experiments and from peer review.
– Section 6 concludes the paper.

2 Big data and healthcare data interoperability

2.1 Big data in healthcare

One of the consequences of the changing healthcare environment is the production of heterogeneous, voluminous, medical data which necessitates the creation of comprehensive medical profile of the patient to improve healthcare service delivery. In particular Clinical Decision Support Systems(CDSS) require the combination of several data sources, such as diagnostic tests, patient's clinical history, CPG, vital signs, symptoms and others, to aid the decision making process [93].

Traditional healthcare systems have focused on using relational databases for persisting EHRs. Based on the idea of a well-structured storage solution, with the ability to uniquely store and identify tuples and their inter-relations, relational databases are beneficial for small to medium scaled medical systems, with little to no interoperability. Other research led initiatives are now turning towards NoSQL technologies [20,72,79] such as cloud based Column Oriented data store for storing healthcare data in HL7 v3 form by Celesti et al. [12], which provides very low query (with aggregation and filter operations over column data) execution times on very large amount of data, and Graph DB utilized by Balaur et al. [5] to integrate statistical data on molecular inter-dependencies from a manually curated and annotated relational database. The UHPr usecase, of retrieving related medical records for a patient, necessitates the use of a document oriented data store, which can hold each EHR record as a document. Since the target of UHP services is to provide a fast solution to the healthcare interoperability problem, and the participating schemas dont follow any formal messaging or communication standard, Relational databases, Column oriented data stores, and Graph data stores are not useful [79].

While the UHPr architecture is generic enough to run on any document based NoSQL engine, our current work utilizes hadoop distributed file system (HDFS), as a document store to archive medical fragments and Apache HIVE [82] (a structured query engine that runs on top of HDFS) as the data processing unit. The choice of these tools is based on four factors; firstly, the authors familiarity with the Hadoop ecosystem allowed for quick deployment, management, and customization. Secondly, Hadoop

[81] is able to run on commodity hardware and can scale very well, when resources are added or removed from the cluster. Additionally, both HDFS and Hive provide native Java API, which is useful for integration with the bigger platform. Thirdly, a plethora of literature and community based help is available for customization and resolving errors. Finally, HDFS and Hive both have a native Java API and command line access, which is useful for integration with the bigger platform. While a lot of effort has been put into developing proprietary solutions (like Essentia Health,[1] Omni MD,[2] and BlueEHR),[3] and some open source ones (openMRS[4] and openEMR[5]) which can capture heterogeneous data and create an EHR, there is a general lack of Big Data solutions for the healthcare market [3]. While there is no formal definition of the term "Big Data", any data will require a specialized storage and processing engine, if it has the following 5 properties (also known as the 5 Vs of Big Data), Volume, Velocity, Variety, Veracity, and Value [40].

Volume Medical data can be classified into two types, primary data sources and secondary data sources [26]. Primary data sources require direct interaction with the patient for data creation. On the other hand, Secondary sources, represent the knowledge management systems, clinical research systems, Biobanks and other tools used by epidemiologists and medical experts, which provide supplementary diagnosis, treatment, and follow-up plans, based on indirect observations (e.g. environment and general living habits). Compounded by the number of patients (e.g. 500,000 participants in UK Biobank [78], 100 million for mendelian disorder risk [9], EHR4CR project with 45 partners in EU [18]) and medical IoT (producing streaming data using body sensors) the storage requirement for a comprehensive digital health persona has already grown beyond the scalability, and speed of traditional relational databases.

Velocity Healthcare data producers, emit data at different rates, pertaining to the use of information systems or medical devices. While medical information and knowledge systems, produce non-streaming data, which is seldom updated (relatively). Medical IoT can produce streaming data, which is continuously produced and has to be shared in real-time [46,60]. E.g. a heartrate monitor on a smart watch produces many instances of very shallow data, while the EHR is longitudinal and deeper, with infrequent instantiation. This requires the use of specialized hardware with low latency, high reliability, and rapid access to the data.

Variety Variety or Heterogeneity in healthcare data, stems from the existence of a large number of formal standards [45] and non-formal/custom standards [24]. This has led to the creation of several semantic reconciliation techniques and platforms which can resolve interoperability among the EHRs [44]. Medical systems also suffer from a variety of purpose, whereby they are created and used to serve the patient (e.g. smart watches), the medical experts, organizations (hospital and/or insurance companies), or environment (e.g. government, consortium) [21]. Consequently, the

[1] Essentia Health: http://www.essentiahealth.org.

[2] https://www.omnimd.com/.

[3] https://blueehr.com/our-services/electronic-health-records/.

[4] https://openmrs.org/.

[5] https://www.open-emr.org/.

data produced by these systems only conforms to their own abstraction level. This means, if an HMIS has to be used for running a small clinic, in a developing country like Pakistan, it would only work at the medical expert's level, leading to the usage of a cheap solution, creating non-standard, EMR.

Veracity Due to the heterogeneous nature of medical systems, EHRs suffer from a lack of universal quality. Universal quality is a made-up term, which is used to identify a golden set of features that an ideal EHR storage and processing system should have. In the real world, EHRs do usually conform to some (standard) schema, making them accurate, true and valid in a given context. However, as the (standard) schema is changed, the existing data becomes stale and often loses its usefulness as well. Additionally, the mere presence of schema would not enhance the quality of data. Additional enrichment information in the form of linked medical records and supplementary knowledge bases are necessary for achieving this aim. LinkedEHR has presented a good approach to partially resolve the data veracity problem, by identifying and building a common platform for primary and secondary data [19], leading to actionable insights into diagnosis, risk stratification and treatment [30]. Yet another key factor to consider here is the fact, that high volume does not always translate to veracity. While it is possible to dilute the gaps in data, when doing quantitative research, the same is not really possible in qualitative research [10]. One way of verifying the truthfulness or veracity of medical data is to measure the data quality in terms of its timeliness (e.g. When did it happen?), completeness (e.g. Did we capture/record everything?), uniqueness (e.g. Is this a duplicate entry?), validity (e.g. Does the data correspond to its schema?), consistency (e.g. Is there any conflicting data?), and accuracy (e.g. Was the medical data recorded accurately, mirroring the real world events?) [3].

Value The main driving force behind the creation of UHPr is to ease the process of converting high volumes of diverse healthcare data, being produced at ever increasing velocity and of varying quality into information and knowledge. Due to its nature as an integrated healthcare record, the UHPr is able to provide value, to the patient, the medical experts, organizations, and the environment. The UHPr complements the benefits from traditional healthcare systems [31] by enriching each patient record with supplementary data from secondary sources and medical IoT.

2.2 Healthcare interoperability

As defined by IEEE 610.12, interoperability is the ability with which, two or more participating information systems or components can not only exchange information but also use it [25]. Building on this basic definition, Health Level Seven International(HL7), a healthcare standard management body, divides interoperability into functional and semantic types; where the former relates to reliable exchange of information, while the later allows the receiver to interpret and use the information. Additionally, CEN ISO/IEEE 11073, is a multi-part standard, developed in collaboration with other standards development organization, that defines the communication standards, enabling real-time, efficient exchange of data produced by (plug-and-play supported) medical care devices [14]. HIMSS, provides a more comprehensive defi-

nition of healthcare interoperability by defining it as the ability to exchange data, at foundational (only relates to exchanging data, without the need to interpret it), structural (an intermediate level, that takes the schema of the data into account as well), and semantic (takes, schema and meaning of the information into account) levels, within and across organizational boundaries [39].

Ubiquitous healthcare can be formalized using these definitions. However, achieving interoperability, in the presence of voluminous, heterogeneous, low quality healthcare data, produced at different rates [40], is an uphill task. This is compounded due to the development of a plethora of messaging, terminological, decision support and other standards [44,45]. Besides the well-defined and developed standards, practical healthcare informatics also suffers due to the existence of non-formal standards, which are used to build specialized small-to-medium scaled systems. Healthcare organizations tend to move towards standards that are easy to use and cost effective [15]. While, this is usually not a problem when medical components have to be made interoperable within the organizational boundary, interoperability between different, often competing, healthcare organizations is a major challenge [49].

Data Interoperability, is a part of the general interoperability problem, which represents the set of policies and guidelines, and their application towards building systems and services that can help create, exchange and consume data while maintaining its contents, context and meaning. These tasks require the use of schema matching/mapping approaches, to map (transform) source data into a consumable form [63]. The main approaches to data interoperability can be categorized as standard based and mediation based approaches. Whereby the former, is focused towards creating and using agree-able standards, which all participating organizations must conform to, while the later, more autonomous approach, creates data translations from descriptions of the data in participating schemas [66]. Linked Data is a well-known example of standardization based data interoperability approach [8], while semantic information layer (SIL) [76] is an ontology mediation approach for data interoperability among enterprise information systems (EIS).

In healthcare, data interoperability can greatly enhance the financial and administrative aspects by reducing overhead and redundant costs, saving time at both the patient and physicians end, preventing operational waste, and allow policy makers to employ the best accountability and privacy services across the board [7].

Overall, healthcare Interoperability (when achieved), will additionally enable the healthcare organizations to increase the data and service delivery quality [74] and remove gaps between healthcare providers and patients [68].

In order to resolve the heterogeneity problem in healthcare, we have to look at the use cases, where an interoperability service can be utilized. In the case of Ubiquitous Health Platform, as shown in Fig. 1, input medical fragments can either be transformed from a source schema to a target schema, or it can be amalgamated into a comprehensive model for the patient's medical history. The former, challenge can be resolved using semantic matching algorithms while the later requires semantic amalgamation. These techniques are further discussed in the following sub sections.

2.2.1 Semantic matching

While there can be many ways to cater for bridging the ever growing gap between heterogeneous medical systems and bringing them on the same connected platform, two primary strategies are standard based and mediation based semantic reconciliation [66]. Here, the former aims to develop a central standard, which all medical systems can comply with [18], while the later, uses mediating ontologies, which can semantically transform data from one format to another [76].

In coming up with a central standard, Clinical Information Modeling Initiative (CIMI) [13] has shown great promise, by integrating the best features of Health Level Seven Version 3 (HL7v3) [35] and openEHR. This endeavor is especially important, given the fact that both HL7v3 and openEHR provide structurally distinct templates (and archetypes) for medical data representation and exchange [17]. In the same way systematized nomenclature of medicine—clinical terms (SNOMED CT [75]), is a terminological standard representing systematically codified clinical nomenclature, while and logical observation identifiers names and codes (LOINC) [51] is a terminological standard for laboratory tests and other measurements. Until 2013, both of these standards had some overlapping, leading to problems in using them together. However, efforts are now underway to link LOINC and SNOMED CT, removing any overlapping, leading to healthcare interoperability at the terminological level. In terms of achieving some automation for this semantic reconciliation process, a lot of state-of-the-art ontology matching tools have been presented using the ontology alignment evaluation initiative (OAEI) [62] platform. However, apart from few matching tools, most have limited extendibility, reusability, and expressive mapping representations, leading to their low adoption rates. Semantic reconciliation, using mediation based approaches, require the usage of similar ontology matching and transformation techniques, which can bridge the gap between heterogeneous systems. Over the years, several methods have been proposed and implemented for achieving the objective of interoperability. These methods, include but are not limited to, the use of standards, mediation via third parties, specification-based interaction, and mobile functionality [64]. Semantic Mediation Systems, represent a formal transformation process, which can provide coupling and cohesion between different data sources [66,85], using Model-Driven Engineering [6].

A plethora of medical platforms have achieved some form of interoperability by mediating between healthcare standards, and extending the benefits of formalization and systematic definitions. One of the most prominent and active semantic transformation tools is the LinkEHR [84], which provides transformation between between HL7 clinical document architecture (CDA) [33], openEHR, CEN/ISO 13606, CIMI reference model, and others [52]. LinkEHR, uses archetypes which contain definitions of clinical information models and a mapping specification generated by the knowledge engineer which is then used for converting legacy data into one of the supported standard types, finally producing a normalized XML file. This conversion is based on a common ontology which provides both syntactic and semantic relationships between the two participating schemas [53,55]. The knowledge engineer, with ample knowledge on informatics can use a purpose built UI for matching the schemas. Application of LinkEHR have also proven effective to achieve interoperability between CDSS and

EHR, which correspond to different levels of abstraction in terms of patient information (usually CDSS is a more abstract representation than EHR) [54]. The LinkEHR platform doesn't provide native data storage services but can be integrated with other similar implementations(including other LinkEHR deployments) and can also act as a semantic transformation engine for other healthcare interoperability platforms.

2.2.2 Semantic integration

Traditionally, healthcare solutions have focused on the use of well-structured storage for resolving interoperability. However, with a variety of medical platforms becoming widely available the interoperability problem now requires the use of ontologies and semantic maps which can identify and create relationships between various data elements from various sources [23]. Semantic Integration, provides a solution to the interoperability problem, by utilizing standardized models, in the form of resource description framework (RDF) [90] and web ontology language (OWL)[89]. Three main methodologies to achieve semantic integration are discussed as follows.

Ontology-based data access (OBDA) framework represents such a solution that is dependent on well-defined domain ontologies, which can map concepts from several data sources. The OBDA model consists of data elements and their semantic relationships build using a terminological service. When a user queries for some selected variables associated with the patient data, it is converted into SPARQL [88] which identifies the semantic relations between participating systems and creates native subqueires, which are executed in a federated manner. The results from these queries are finally integrated using unique identifiers from their data tuples.

The usefulness, of this framework to semantically integrate medical data for cancer patients is proved in [92]. The authors used a top-down approach to first construct an ontology for cancer research variables (OCRV), which contains the semantic relationships between the concepts in virtual RDF graph forms, from four different relational data sources. This ontology contains well-defined terminologies which are based on the National Cancer Institute (NCI) Thesaurus [61]. For converting SPARQL queries into native SQL queries the Ontop OWL API [11] is used which relies on the Ontop model, containing both semantic axioms and the data configuration necessary for connecting with the data sources. The results from each data source is then integrated using the unique identifiers for all records, and presented to the client.

Some multi domain semantic integration strategies, have focused on the development and/or usage of enterprise service bus (ESB), which provides a loosely-coupled, highly distributed, communication channel for software applications and modules in a service-oriented architecture (SOA). In general, several services can connect with this shared communication channel as a consumer or a producer. Each producer converts the messages into an internal format understood by all services, especially consumers. Using a publish/subscribe model, the services are able to communicate with each other using event driven paradigm. In healthcare IBM provided an early implementation of the ESB to create the IBM Healthcare Service Bus [56] which enables the integration of multiple services by using web services description language (WSDL) [86], simple object access protocol (SOAP) [87], and HL7 Standards. The service has now been upgraded [38] to become completely deployable on the cloud and to provide

support for many healthcare standards suchs as HL7v2.X, Fast Healthcare Interoperability Resources (FHIR) [36], Digital Imaging and Communications in Medicine (DICOM), and others. It also now supports several types of message flows including eXtensible stylesheet language transformations (XSLT), extended structured query language (ESQL), file transfer protocol (FTP), java message services (JMS), and others.

Health service BUS (HSB) [67] is an implementation of the Mule ESB [58], which uses a native XML-database and XSLT to provide semantic translation services from HL7v3 to HL7v2 [34] and openEHR. The patient EHRs are stored using OpenEHR database, while HL7v2 and HL7v3 are used for sharing messages belonging to a particular patient. The HSB uses SNOMED CT for providing terminological services, which are also embedded into the ESB as XML messages and used with a custom ontology mapping tool called OWLmt to provide semantic interoperability between patient records. In [56] an event-based HSB based on the JBossESB is presented which converts heterogeneous data into RDF quads, before utilizing the health and lifelogging data (HLD) Ontology for building a semantically linked graph of health and lifelog data. The authors have used LOINC as the terminological handler, which is used to provide semantically annotated versions of input sensory data from wearable devices, before creating the RDF quads and applying semantic integration using the HLD ontology. Internally, the bus is able to provide point-to-point communication between any two services, and a publish/subscribe broadcast model using JMS queues. The overall platform can be used to push notifications to the users, using event-driven paradigm and can also provide query services for executing SPARQL queries.

Yet another interesting initiative is the Yosemite Project [43], which aims to bridge the gap between healthcare standards and the data. The main driving force behind this initiative is the conversion of messaging standards like HL7v2 and FHIR into RDF graph for semantic representation. It is also concerned with resolving the ambiguity in the human language by using Natural Language Processing technologies for processing unstructured medical data (such as Clinical Notes, Clinical Practice Guidelines, and others). Their methodology consists of two related process, standardize the healthcare standards and using crowdsourcing for translations. Here the former task has been undertaken to find and create semantic links between 30 most used vocabularies amongst over a 100 listed by unified medical language system (UMLS) [59].

Using a custom tool, iCat, which currently only support international classification of diseases (ICD)-11 [91], the yosemite group provides an easy to use interface to the medical experts. Over 45,000 concepts with 17,000 links to external terminologies have been defined by the medical experts, which are converted into RDF form for creating computable data resources. The latter task of translation using crowd sourcing resolves the problem of standard complexity, evolution of technologies and methodologies in computing and healthcare, and finally change in the standards themselves. This translation process is an extension of the inference process which can identify implicit relations, RDF assertions and localization between languages to enrich the existing semantic maps. This semantic integration is language/tool agnostic and can be used with any other platform.

While many paradigms have been introduced to resolve the semantic matching and integration problem, it is clear that the difficulty in creating ontologies and seman-

tic bridges between various standards and terminologies is greatly hampering any functional interoperability solution. Additionally, the initiatives for standardizing the standards are still about a decade away from becoming implementable. Meanwhile, the healthcare data is growing beyond the management abilities of traditional data curation engines. Moreover, the top-down approach necessitates the use of medical experts for initially creating a rule base and/or ontology, which is not always possible. It is therefore necessary that a novel methodology is used to archive the existing medical data and keep it available to create and test semantic integration methodologies. Additionally, due to the various methodologies involved in this semantic reconciliation process, it is important to store this data, while maintaining most of its original schema. Conversion to RDF quads, XML, relational or other methodologies can lose the original schematic information.

3 Ubiquitous Health Profile (UHPr)

3.1 Platform novelty

As evident from the discussion above, healthcare interoperability, presents a major challenge towards achieving ubiquitous healthcare. Many factors influence this challenge, including availability of a large number of standards, evolution of standards, privacy concerns around patient data, lack of access to healthcare data, large number of healthcare information management and support systems, and others. In aiming to resolve these problems, one crucial question has been left unanswered in literature, relates to, how do we provide interoperability support to the large number of small and medium scaled HMIS and other healthcare platforms, which are not currently complying with any formal standard?

The Ubiquitous Health Platform, aims to provide a solution to this problem by providing a large medical archive and transformation platform, which can evolve and apply the semantic reconciliation process with changing organizational needs. It is also imperative to mention here, that while initiatives to standardize the standards, like CIMI and Yosemite project are slow in their development, they are necessary for any healthcare interoperability solution to evolve and generalize in future. Essentially, the UHP and its underlying UHPr engine, together with a semantically integrated standardized communication, messaging, and storage mechanism would act as mutually enabling healthcare interoperability enabling technologies of the future.

On the other hand, technologies and platforms such as LinkEHR, OBDA, and HSB provide an alternate to the UHPr approach, which have been discussed above and briefly compared in the Table 1. The platforms have been compared in terms of their features during data acquisition and data retrieval. This comparison has been based on the available literature only and what has been achieved so far and not in terms of their capabilities which is beyond our scope. In particular, LinkEHR has focused on using well defined archetypes to provide a semantic and syntactic transformation engine, with large input from the knowledge engineer, leading to high dependency on well-defined standards such as HL7 CDA, OpenEHR, and others. Once the mapping has been provided, there is little to no chance of data loss during data acquisition

Table 1 Comparison with existing platforms

	Data acquisition			+	Data retrieval		
	Dependency on well-defined standards	Required intermediate data conversion	Effort to add a new data source	Data loss	Dependency on well-defined standards	Required intermediate data conversion	Traceability
LinkEHR	✓	✓	✓	✓	✓	✓	X
OBDA	✓	X	✓	X	✓	X	✓
HSB	✓	✓	✓	X	✓	✓	X
Proposed (UHPr)	X	✓	X	X	X	✓	✓

since LinkEHR does not natively store the data, rather it provides transformation on the fly. However, based on how complete the metadata and bridging ontology is, LinkEHR may lose data while data retrieval. Based on the user query, the internal XML representation may require an additional conversion to the requested form, with the help of two way semantic relations defined by the knowledge engineer. Finally, from the current literature, there is no evidence to suggest that LinkEHR manages the traceability of healthcare records, beyond what may be present in the standard itself.

On the other hand, OBDA is dependent on a well formed ontology, to which all participating systems must comply. As a result adding a new source can become problematic if it does not comply with the structure in the current ontology. Similar to LinkEHR, OBDA does not lose healthcare data owing to its use of only remote data source connections. During data retrieval, data loss by OBDA is dependent on the accuracy of the reasoner and how well the custom API is able to transform the SPARQL queries into native SQL queries. This may lead to some data loss, in terms of the number of retrieved healthcare records. There is no data conversion during acquisition or retrieval by OBDA, however there is very limited traceability in terms of unique identifiers from various healthcare sources, participating in the result set.

Both OBDA and LinkEHR utilize the federated query model to resolve interoperability during data retrieval and are based on well-defined semantic bridges between participating healthcare sources. While LinkEHR uses a one-to-one model, where each pair of systems have a supporting archetype and metadata, OBDA uses a central ontology and thesaurus to bridge many systems together. HSB also uses a semantic interoperability paradigm similar to OBDA, however in HSB, the various healthcare systems, as producers and consumers are only loosely coupled with each other and require transformation services from well-defined standard form to an an internal format for exchanging data. Additionally intermediate conversion at both data acquisition and retrieval phases is required to convert from one standard form into another. Data Loss in HSB is mitigated through the use of buffering queues. Finally, the current implementations do not show any traceability at the data source level.

Finally, the proposed engine UHPr, is not reliant on any well-defined healthcare standard but requires serialization of the data and its conversion into a semi-structured format before storage. This process is explained, in some detail, in the next sections. Adding a new data source to the UHPr is relatively a trivial process and is dependent on writing a simple java class which can read the data, extract meta patient information(name and date of birth), serialize the data as a single string. The UHPr on retrieval requires extensive conversion to convert the same data string into a computable and semantically enriched form. Since UHPr archives the medical data and semantic maps for bridging schema it does not suffer from data loss. UHPr also provides traceability for identifying the patient and the source medical system.

Figure 2 shows the comparison between various interoperability paradigms. Figure 2a shows the standards based approach, whereby the semantic reconciliation process is used to transform non-compliant data sources. After this process, the interoperable medical data is in one standard form, which enables the user to execute one query and get the results. An alternate to this approach is shown in Fig. 2b and as an example the pipeline typically followed by the federated query approach is shown. In this approach a controller is used to generate separate queries for each of the data sources, these are

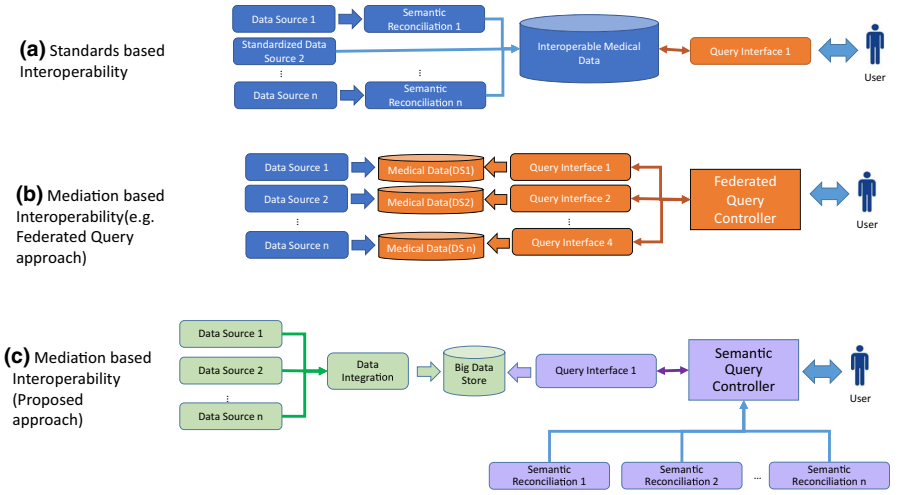
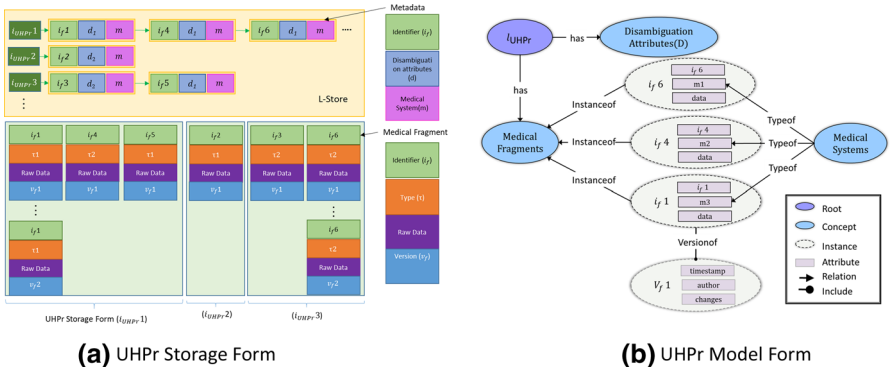


Fig. 2 Novelty of the proposed approach in terms of the semantic reconciliation pipeline

executed on the corresponding Medical data, the results of which are then combined and shown to the user. Figure 2c shows the pipeline of our proposed approach which archives all medical data after integration (conversion to semi-structured form) and then uses UHPr maps to apply semantic reconciliation on subsets based on their individual schema and relationship to the inquired schema. The proposed approach is able to efficiently deal with data volume (using well established Big Data tools and technologies), variety (unlike the other two approaches, requiring less intervention for each integrating new data source), and velocity (by separating the data acquisition and semantic reconciliation process like other mediation based approaches). This platform then provides the foundation for identifying new values from the integrated medical data and enhance its veracity. On the other hand, most interoperability initiatives are tightly bound with existing standards and data exchange interfaces [32]. The novelty of our approach, towards solving the Interoperability problem, lies in delaying the semantic reconciliation process, and thereby moving it away from the data and closer to the user. As a result, UHPr has been optimized for acquisition, storage and minimal processing of the medical data.

UHPr, provides data curation services for heterogeneous medical data, utilizing two distinct but related forms; UHPr storage form and UHPr model form. The UHPr storage form, utilizes the concept of minimal changes before insertion routine, which is used for dumping data into an archive. While the UHPr model form represents a graph data structure that holds the comprehensive digital profile of a patient. In the following sub-sections, we will first present the theoretical representation of the UHPr, followed by our prototype implementation, which is focused on creating the infrastructure for storing and processing of the UHPr, to finally produce a comprehensive medical profile of the patient. The use of heterogeneous data models in hospital management and information system (HMIS) obstructs the communication and integration of the systems in clinical workflows. The diverse medical concepts diminish the systems'



(a) UHPr Storage Form **(b)** UHPr Model Form

Fig. 3 UHPr data representation

interoperability. The aforementioned barrier is overcome using semantic reconciliation model, which is proposed in our previous work [1]. The proposed mapping model maps diverse localized concepts, called domain clinical model (DCM), with standard and non-standard medical terminologies. The existing mapping algorithms only focus on the internal semantics of terminologies such as parents, childs, and siblings similarity matching within the source and target terminologies. While in our proposed model, we include the external semantics of the source and target concepts in the form of concepts and relationships provided by the semantic libraries such as UMLS [59] and ConceptNet5 [50].

3.2 Theoretical representation

The UHPr storage form, as shown in Fig. 3a represents the medical fragments acquired from a variety of healthcare sources and stored in semi-structured form, characterized by a data and a metadata component. The data part of UHPr storage form, also called the Medical Data Archive, is represented by a 4-ary Cartesian product of the set of Identifiers(I), Types(τ), Serialized Fragments(F), and Versions(V). Where set of Identifiers, as defined in Eq. 1, is used to uniquely identify an individual record in the UHPr storage engine. The identifier is dynamically generated using any fixed or dynamic length technique [41]. It is not dependent on any features related to the patient or the source medical system and is used for linking the data components with their respective patient's metadata component, which are in-turn identified by their own unique identifiers. Universally Unique Identifier(UUID) as defined in RFC 4122 [48] version 4 (pseudo-random) with its 128 bit encoding can be used to uniquely identify upto 5.3×10^{36} objects and is well qualified for use as identifier for both data and metadata components.

$$I = \{i_f | f \epsilon F\} \tag{1}$$

The set of Types, defined as in Eq. 2, holds a unique identifier for the participating medical fragment schema that a particular medical fragment corresponds to. For practical purposes, the name of the medical fragment schema(such as OpenEMR, HL7

CDA, KrsiloEMR, or other) is sufficient to be used as an identifier. In case of collisions, the name can be augmented by other differentiating features, such as the organization name, country code and so on. This meta information is used to select the appropriate UHPr map for semantic linking or transformation, during retrieval. Consequently, a medical fragment is considered unique, and becomes a candidate entry in the sent τ, if it has a different schema than the ones already participating. Essentially if a two medical fragments, coming from two different organizations, but following the same schema τ_1, would result in one unique entry in the set τ.

$$Type(\tau) = \{\tau_1, \tau_2, \tau_3, \dots\} \tag{2}$$

The non-empty set F, defined in Eq. 5, represents the serialized form of the medical fragment, provided by a connected medical system M and identified by a type τ. This serialization, de-normalizes the data into key:value form, where each key belongs to and is unique within the schema τ. For disambiguation, keys can be prepended with the database name and table name, if they come from a relational data source. This is used to provide disambiguation between the keys, which enables correct semantic matching and transformation, at retrieval.

$$F = \{f_m | f_m : \tau \& m \epsilon M\} \tag{3}$$

The set of versions V, in Eq. 4, represents a ternary of author, timestamp, and the changes to a previous version of the fragment f_m. Version control in UHPr is provided only for handling minor errors in existing medical fragment data. Any change to the metadata(such as patient's name or date of birth) should be managed by creating a new medical fragment and handling this corner case at the consumer's end. In line with the Big Data architecture, the UHPr curation engine discourages any update or deletion of records, which would require a deletion of the entire archive fragment containing many records and reinsertion of the same(a very expensive operation in terms of data consistency and availability).

$$V = \bigcup \{(t, a, v_f) | v_f \subset f_m \& t = timestamp \& a = author\} \tag{4}$$

UHPr storage metadata, also known as the location store (L-Store), contains meta elements of the UHPr data structure. This store, as shown in Eq. 5, provides a logical indexing service, by storing references to the global identifier i_{UHPr}. These references, in turn refer to the medical fragment identifier from Ipatient's meta information and the medical system sourcing the health record.

$$L = \{i_U HPr \Rightarrow (i_f, d, m) | i_f \epsilon F \& d \epsilon D \& m \epsilon M)\} \tag{5}$$

In addition to the medical schema type τ, some information is also required to uniquely identify the source medical system. This information is available in the metadata component of the UHPr storage form, and is defined in Eq. 6. This information is also kept as a single string to keep the overall data structure largely denormalized. Since this information is a part of the metadata, it can become a part of the UHPr engine,

only if there is a medical fragment in the archive, sourced from the medical system (m).

$$M = m | hasFragment(m) \tag{6}$$

Additionally, some disambiguation attributes(D) are necessary to keep the global identifiers unique. The selection of appropriate attributes to uniquely and universally identify a patient, across medical systems is a big challenge, which is discussed briefly in Sect. 5.1. A naïve implementation can use the patient's name and date of birth for this purpose. This is shown in Eq. 7.

$$D = \left\{ d | \exists d_x \epsilon D : (\forall d_y \epsilon D \rightarrow d_x = d_y \right\} \tag{7}$$

UHPr model represents the structurally integrated output of the UHPr storage engine. This particular data representation is used to zip together the most important aspects of the user's record and to provide an iterate-able data structure to the consumer. The resulting data structure can be in the form of a well-defined standard(such as HL7 V2, HL7 V3, HL7 FHIR, CIMI archetypes, or others), or in a graph data structure, shown in Fig. 3b. This data structure is obtained by structurally transforming the UHPr storage form. UHPr storage conversion to a well-defined standard form requires a supporting schema map, however out of box support for the UHPr model graph form is provided by the UHPr engine.

This graph data structure contains the i_{UHPr} as the root node. The root node is linked to patient's disambiguation attributes (D), which can be used by the consuming agent to identify the patient. Additionally, it is linked to the set of all the medical fragments instances belongs to the patient. Each instance is identified by its unique identifier i_f. It also contains the medical system m (from Eq. 6) and the data element, which unlike UHPr is semantically enriched to contain semantic relations or transformed into a target schema, based on the retrieval query. Changed versions are linked with their respective data elements for supporting traceability of medical records. The version elements contain the timestamp of change, author information, and the changed data, corresponding to the data element. In this way, the UHPr model is able to re-build a comprehensive medical profile of the patient. This theoretical representation provides the foundational elements of the UHPr engine. It provides the necessary infrastructure for providing data level interoperability, in particular and supporting healthcare interoperability, in general. In the next section we present the implementation details for building the UHPr engine.

3.3 Implementation

3.3.1 UHPr storage

Implementation of the prototype UHPr storage form has been achieved by consolidating information from three medical systems (M), OpenEMR patient reports, 100,000 patient data set from EMRBOTS [42] and our custom implementation of expert driven medical diagnostic system (Krsiloemr). This platform is based on Hadoop, with HDFS acting as the main storage medium, while Apache Hive is used to temporarily create

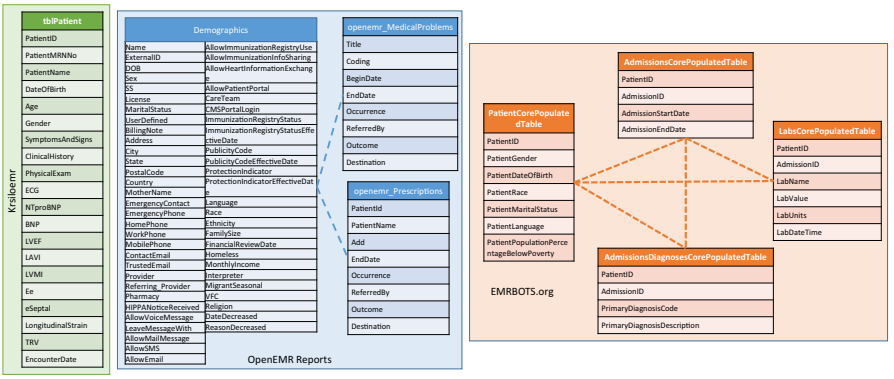


Fig. 4 Schemas for medical fragments participating in UHPr

the UHPr schema (shown in Fig. 4) and fetch all records for the patient. The UHPr hadoop deployment is composed of 1 master and 2 slave nodes with 1.8TB HDFS size, 20 MB block size, Block Replication of 3. The master node has 64GB ram, while the slave nodes have 32GB ram. Each unit of this cluster has 4 core AMD Ryzen 3 2200G processor ([2]), and has CentOS ([83]) 7.5 as the operating system.

The UHPr storage form as shown in Fig. 3a, is stored in form of text files in HDFS, which in turn, contain various medical fragments in semi-structured form. With Hive we temporarily create a schema, utilizing the semi-structured elements (the identifiers) and perform complex queries, which are then converted into MapReduce operations. Each patient is assigned a global identifier (i_{UHPr}) using a 128 bit UUID which maps each patient's firstname, lastname, and date of birth with a related medical fragment (f_m). Medical Data Archive, stores the medical fragment in block form, where many medical fragments are combined together into one file (identified by the global id). The medical fragments, in turn, contains, the unique identifier, as available in the L-Store (different versions of the same medical fragment, will have the same identifier). Additionally, it contains a type element, which is used to identify the schema of the medical fragment and will be used later on for using the correct, ontological map for transformation. Each fragment also contains a locally unique version identifier, which is used for managing instance evolution and verification purposes. Starting with 40 real patients in Krsiloemr and 12 patients for openEMR with various medical problems, we generated medical fragments for 80,000 patients. Each patient has 1 openemr-Demographic Report, and is randomly assigned another 29 medical fragments amongst Krsiloemr, openemr-MedicalProblems, and openemr-Prescriptions. After 7 iteration and including the 100,000 patient dataset from EMRBOTS, the data store now contains 115,737,428 (a little over 115 million) records, corresponding to 390,101 patients. The dataset from EMRBOTS was slighlty modified to include 'PatientName' (since this is required for our approach), before being serialized into a UHPr compliant format. The schema for our three participating medical systems is shown in Fig. 4.

Through our experiments, we were able to determine that the most feasible strategy to store these fragments, along with L-Store metadata, in HDFS is by using a 1 file-per-transaction strategy [70]. In this strategy, we consolidate various medical

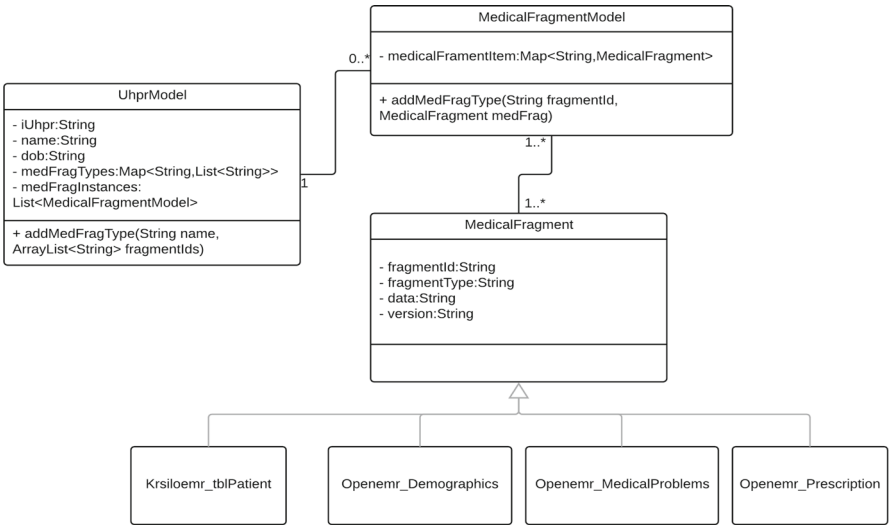


Fig. 5 Class diagram, representing the UHPr model building application

fragments, from 1 transaction(similar to data buffering) into 1 metadata, 1 data, and 1 connector file. The metadata file, contains the meta information for the L-Store, the connector file contains index entries for mapping i_{UHPr} identifiers to i_f identifiers, and the data file contains the medical fragment, corresponding to each i_f identifier.In this way we can store a large amount of data in relatively smaller number of files. This strategy enables the most preferred way of data processing using MapReduce operations, with small number of large sized files [27,28]. As a result of this process, the UHPr is able to achieve transactional consistency. For data processing we then move the relevant records into memory by employing a temporary external table (schema-on-read), created using Hive. Using simple Hive Query Language (HiveQL) based queries (as shown in Table 2) we are able to retrieve the medical fragments belonging to a particular user (Fig. 5).

3.3.2 UHPr maps

As discussed in 2.1, the mapping of various attributes in the participating schemas is a challenging task. In our previous work we have developed a semantic reconciliation model that insets explicit semantics into word vectors of a SNOMED-CT following schema and a non-formal schema model similar to Krsiloemr [1]. However, for mapping two non-formal schema we need some initial tweaking, especially in terms of identifying the correct stop words that can separate multiple 'word' strings from an 'Attribute' string (such as PatientLanguage, PatientMRNNo, MartialStatus have multiple words which are not identifiable using traditional whitespace based separation techniques). An initial prototype of this scheme has been presented in our previous work [71]. Our extended methodology now utilizes UMLS concepts to enrich and improve the semantic matching. The current approach is shown in Fig. 6.

Table 2 QUERIES

Ine Id	Query	Description
Ine Q1	Select medicalfragmentidx.fragmentid, uhpridx.firstname, uhpridx.lastname, uhpridx.dob, uhpridx.gid from medicalfragmentidx,uhpridx where medicalfragmentidx.gid=uhpridx.gid AND uhpridx.firstname="Harry" AND uhpridx.lastname="Potter" AND uhpridx.dob="19880708"	Selects the fragment id, patient's first name, patient's last name, patient's date of birth, and global identifier, from the L-Store, for user named "Harry Potter" who was born on 19880708
Ine Q2	Select * from uhpr where fragmentid in (select fragmentid from medicalfragmentidx where gid=(select gid from uhpridx where firstname="Harry" AND lastname="Potter" AND dob="19880708"))	Selects the medical fragments from UHPr storage form, by matching the global identifier for the patient named "Harry Potter" , who was born on 19880708
Ine Q3	Select fragmentid from medicalfragmentidx where gid=(select distinct(gid) from uhpridx where firstname="Harry" AND lastname="Potter" AND dob="19880708")	Select the fragment id from L-Store for the patient named "Harry Potter" who was born on 19880708, selecting only distinct global identifiers first
Ine Q4	Select distinct(fragmentid) from medicalfragmentidx where gid=(select gid from uhpridx where firstname="Harry" AND lastname="Potter" AND dob="19880708")	Select only the unique fragment id from L-Store for the patient named "Harry Potter" who was born on 19880708
Ine Q5	Select * from uhpr where fragmentid in (select fragmentid from medicalfragmentidx where gid=(select distinct(gid) from uhpridx where firstname="Harry" AND lastname="Potter" AND dob="19880708"))	Selects the medical fragments from UHPr storage form, by matching the distinct global identifier with the fragment id for the patient named "Harry Potter" , who was born on 19880708
Ine Q6	Select * from uhpr where fragmentid in (select distinct(fragmentid) from medicalfragmentidx where gid in (select gid from uhpridx where firstname="Harry" AND lastname="Potter" AND dob="19880708"))	Selects the medical fragments from UHPr storage form, by matching the distinct fragment identifier with the global id(s) for the patient named "Harry Potter" , who was born on 19880708
Ine		

The process is split into two phases; an offline phase, which generates the semantic maps between the participating schemas, and an online phase which is used to apply the schema for semantic linking or transformation. In the offline phase, all participating medical fragment schemas are provided to the tool in form of a text file which contains the list of attributes in the form of 'SchemaName_IgnoredText, AttributeName'. This list is then used to apply 7 steps, for semantically enriching each attribute (A) by its concept (C) and its corresponding word (W), syntactic comparison between a pair of attributes and their corresponding words, and semantic comparison between the attribute and word pairs. In step 1, a simple case insensitive match between attribute

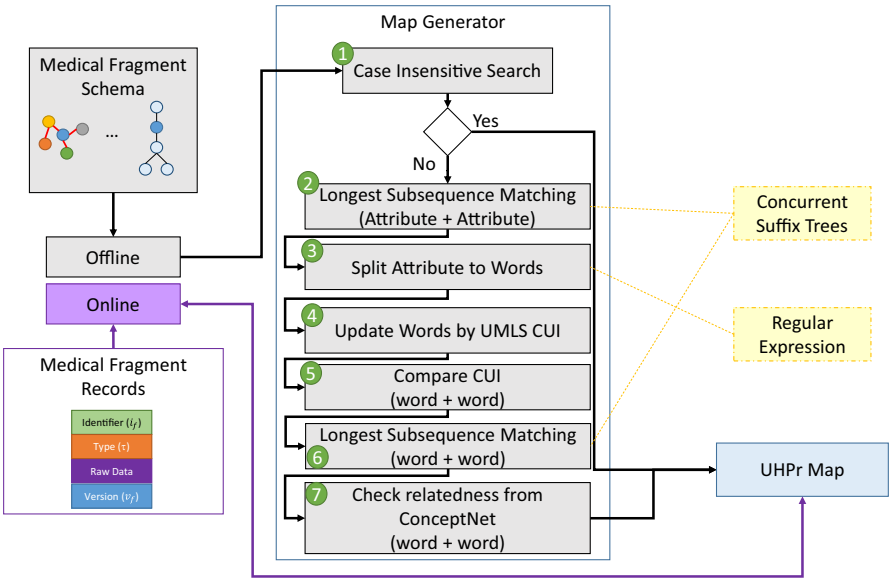


Fig. 6 UHPr Map generation process

strings is applied ($A - A$). If the two attributes match, they are not processed further. If the attributes do no match step 2 is performed. In Step 2, we search for the longest common subsequence between the two attributes $A - A$ using the Suffix Tree method. Using Concurrent Suffix Tree implementation [22], the two attribute strings are combined into one Suffix Tree, which contains character sequences of all possible lengths between the two strings. The longest subsequence which is common between the two strings is placed in the leaf node of this tree, which can be queried quickly.

The attributes are then split into words in stage 3. For identifying the words, we utilized a regular expression to split the attribute string, on any case change (this would lead to PatientMRNo becoming 'Patient', 'MR', and 'No'), any occurrence of digits, or special characters ($A \rightarrow W$). In step 4 each word ($WinA$) is queried on UMLS to collect its associated concept identifier and source ($C[W]$). Utilizing over 213 terminology services, UMLS returns a list of concepts which may belong to a word. Due to result size limitations in the UMLS REST API call, multiple queries are often necessary to collect all corresponding concepts against a word.

Then for each distinct pair of words, step 5, 6, and 7 are performed. First, in step 5, the UMLS concepts from each word pair is compared to identify any similar concepts ($C_1[W_1] - C_2[W_2]$). This is done by intersecting the list of concepts from both words in the word pair. Next, in step 6, Longest Subsequence Matching is applied to identify the syntactic similarities between an attribute and the words from ($A - W$). Here we again use the Concurrent Suffix Tree implementation to identify the longest substring common between the two strings. Finally, in step 7, the words are checked on ConceptNet for their relatedness ($C_1[W_1] = C_2[W_2]$) measure, obtained from numberbatch.h5 embeddings [77]. The final results of these steps are then placed into SchemaMap in JSON form.

The glue between these two phases are the model classes, which are shown in Fig. 7. The base model here is the UHP_MAP of type 'SchemaMap', which represents a HashSet(for holding only unique values) of 'AttributeMap'. The 'AttributeMap' holds two attribute nodes, leftNode and rightNode. Additionally, it contains the schemaRelation, which corresponds to the semantic and syntatic relationship between the two attributes. Finally, it contains comments, method and confidence strings for holding additional details of the mapping. The 'Attribute' contains, the name of the table, complete name of the attribute, and a list of words, corresponding to the attribute. 'Word' holds the title of the word, and a string representation from ConceptNet which can be optionally filled. Finally it contains a list of Concepts which are collected from UMLS, in the form of concept unique identifier (CUI), name, and source (such as SNOMED CT, UMLS Metathesaurus) of the underlying concept. A supplementary 'WordMap' entity has class attributes similar to the 'AttributeMap', (except for wordLeft in place of attributeLeft, and wordRight in place of attributeRight) for holding the semantic relationship between the word pairs belonging to a pair of attributes. This map becomes a part of the comment string, for the corresponding 'AttributeMap' and is used for explanations and mapping traceability. In the online phase, each medical fragment provides the name of the schema, which is used to search for the appropriate SchemaMap. From the SchemaMap, using attribute names, the related schema maps are selected. These are then converted into appropriate object models and loaded in memory, while the UHPr model generator is running. It is then used to either enrich the attribute name or to transform it into a target format. The offline process was executed on a workstation with AMD Ryzen 3 2200G CPU with 4 cores, and a maximum memory allocation of 10Gb(by using the -Xmx Java Virtual Machine argument). A dedicated machine with ConceptNet was also set up with 4 core AMD Ryzen 3 2200G processor, 32Gb RAM, and 10 dedicated virtual processing threads. The Schema Maps for 144 attributes as shown in Fig. 4, were generated in a little under 97 mins. It contains a total of 17913 relationships with step 1 generating 10 relations, step 2 generating 0 relations, step 5 generating 8980 relations, step 6 generating 69, and step 7 producing 8854 relations.

3.3.3 UHPr model

Structured output of the UHPr storage engine is presented in the form of UHPr model. This representation is retrieved after applying UHPr Maps, as semantic bridges, between the various attributes of the participating schema. In order to implement the UHPr Model, a java based application reads the medical fragment data from a csv file, generated from Hive. It also reads the UHPr Maps JSON file and loads the Schema Maps in memory. Then based on the name of the schema for each medical fragment, it reads the appropriate schema map, and generates the graph form of the UHPr model. Based on user request, the UHPr Model generator can either add the all AttributeMaps belonging to an attribute in the output graph, or it can read the use the AttributeMaps linking the source and target attributes and apply transformation, if the confidence score of the mapping is above some user defined threshold. The class diagram for this model is shown in Fig. 5. It uses the UhprModel as the base class holding the root node, and MedicalFragmentModel, holding a map of fragmentId and the medical fragments. The MedicalFragment itself, is a parent class of our 8 specialized medical

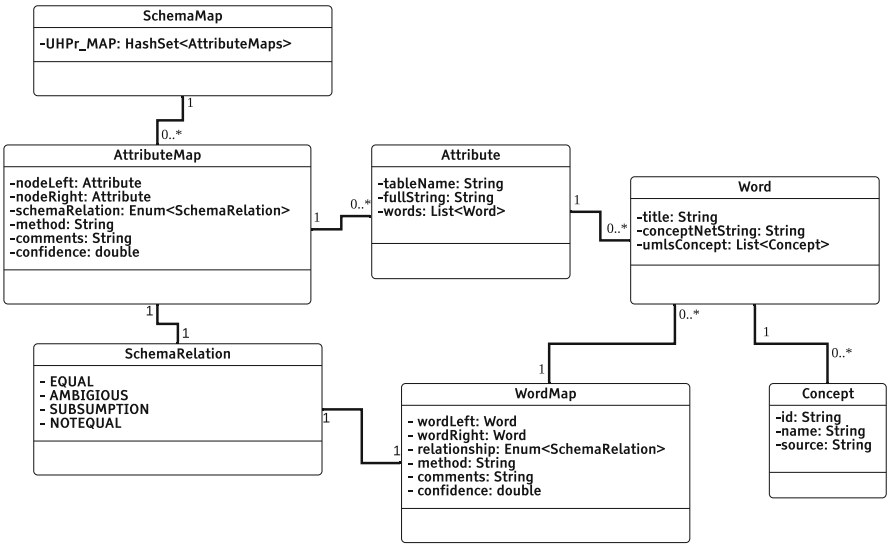


Fig. 7 Modeling UHPr maps

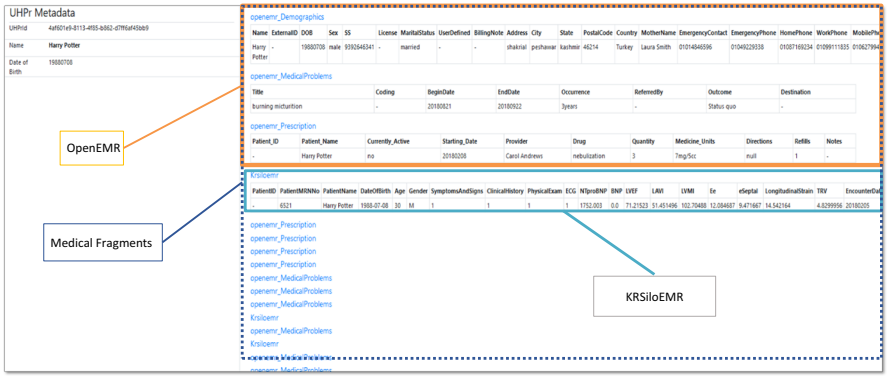


Fig. 8 UHPr results for selected user

systems: ‘ Krsiloemr_tblPatient ’, ‘ Emrbots_PatientCorePopulatedTable ’, ‘Openemr_Demographics’, ‘Openemr_MedicalProblems’, ‘Openemr_Prescription’, ‘Emrbots_LabsCorePopulatedTable’, ‘Emrbots_AdmissionsCorePopulatedTable’, and ‘Emrbots_AdmissionsDiagnosisCorePopulatedTable’.

The same application then transforms the UHPr object form into JSON based graph form. Here, loading the SchemaMap into memory and its deserialization into object form took 988 ms. For 1 patient with 30 records, the semantic linking process took under 3 s. While the semantic transformation process for the same user too 404 ms. The UHPr model, in this graph form, is then transformed into a user friendly format, as shown in Fig. 8.

3.3.4 Availability of data and software

All code(for creating, transforming, and view), some sample data(minus the EMRBots data set), and results related to this version of the UHPr are available in a public GitHub repository(https://github.com/desertzebra/UHPr).

4 Results and evaluation

Building upon the initial results from UHPr, an experimental setup was created, with the aim to evaluate the timeliness, scalability and accuracy of this platform. Based on the data quality definition by [3], the modified evaluation metrics of the UHPr are as follows:

1. *Timeliness* The medical fragments are archived in the storage medium and retrieved from it within some soft time-bound.
2. *Scalability* All the medical fragments are recorded completely.
3. *Accuracy* Each medical fragment is retrieved accurately.

Here timeliness and scalability are related to the performance of the storage engine, while accuracy is related to the data retrieval and transformation process.

4.1 Experimental setup

Starting with a set of 2.4 million synthesized medical fragments against 80,000 patients, we performed 7 iterations to increase the data and test the three metrics. Data for the first 6 iterations is based on 40 real patients in Krsiloemr and 12 patients for openEMR. In iteration 7, we used the EMRBOTS dataset of 100,000 patients. In each iteration, we evaluated 8 timeliness criteria to evaluate the performance of data insertion into HDFS (corresponding to UHPr storage form), creation of temporary schema in Hive, and timeliness of data retrieval (corresponding to the transformation process from UHPr storage form to model form). These are shown in Table 3. In order to test the actual transformation of medical fragments from UHPr storage form to the model form, we executed Q1 and Q2 in iteration 1–5, while Q3, Q4, Q5, and Q6 in iteration 6 and 7, to retrieve medical fragment ids and medical fragments, respectively. The queries and their description is shown in Table 2. These were repeated 10 times to strengthen the results. The evaluation results of these iterations and the relationship of the evaluated criteria across them is discussed as follows:

In the first iteration we started by generating medical fragments for 100 patients, with 20 medical fragments per patient. Total number of medical fragment instances for the user “Harry Potter” , who was born on 19880708 were 30 (same as iteration 0). The results for executing Q1 and Q2 10 times, for criteria C7 and C8 respectively, is shown in Fig. 9. The average time taken for C7 is 28.8528 s and for C8 is 119.1014 s. In the second iteration, the number of new patients was increased to 10,000, with each having 20 medical fragments. Executing Q1 and Q2, 10 times, for criteria C7 and C8, respectively, yielded the results shown by Fig. 9. The average time taken for C7 is 28.4869 s, while for C8 is 121.4805 s. The total number of rows returned by these

Table 3 Evaluation criteria for each iteration

Ine Id	Description	Metric
Ine C1	Time taken to insert UHPr medical fragment file into HDFS	Time
Ine C2	Time taken to insert medical fragment bridging information, linking global id($g_i d$) with fragment id($f_id) into HDFS	Time
Ine C3	Time taken to insert UHPr patient index part of L-Store into HDFS	Time
Ine C4	Time taken to create UHPr table schema in Hive	Time
Ine C5	Time taken to create medical fragment bridging table schema in Hive	Time
Ine C6	Time taken to create UHPr patient index table schema in Hive	Time
Ine C7	Time taken to retrieve all fragment ids for 1 user	Time
Ine C8	Time taken to retrieve all medical fragments for 1 user	Time
Ine		

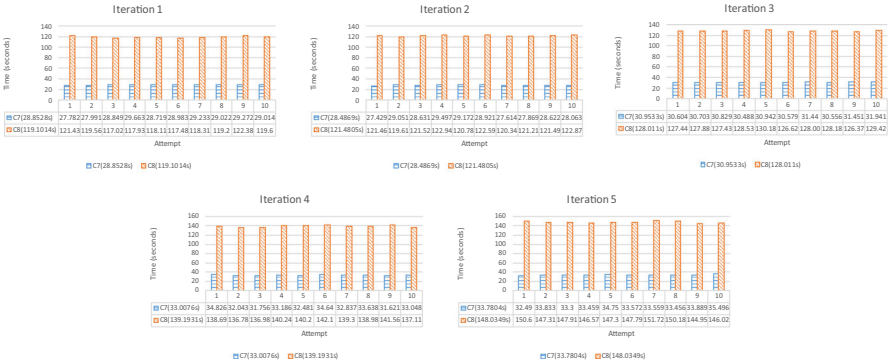


Fig. 9 Iteration 1–5 results for C7 and C8 after executing Q1 and Q2

operations were 30 (same as iteration 0). In the third iteration, 40,000 new patients with 20 medical fragments each was generated. The results for this iteration are shown in Fig. 9. The average time for C7 is 30.9533 s and for C8 is 128.011 s. In the fourth iteration, 80,000 new patient records were generated, with 30 fragments for each. As shown in Fig. 9, the average time for C7 is 33.0076 s and for C8 is 139.1931 s. Similar to the previous iteration, 80,000 new patients with 30 fragments each were added as a new UHPr storage file in the HDFS. As indicated by the results shown in Fig. 9 there is only a slight increase in the amount of time consumed by Hive. With an average time of 33.7804 s for C7 and 148.0349 s for C8, there is a slight increase of 0.7728 s for parsing the medical fragment identifiers and a relatively larger increase of 8.8418 s for retrieving the UHPr storage forms. Here, the former can be explained by the small size of each row, while the latter is the result of processing a large amount of text, especially in the raw data column.

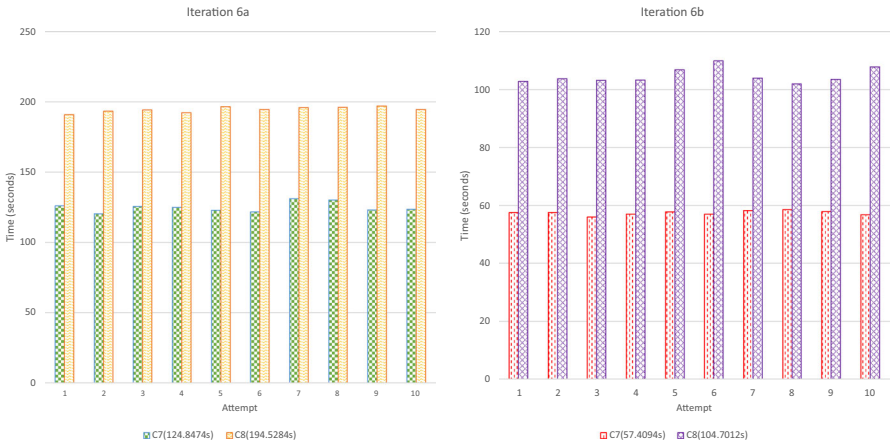


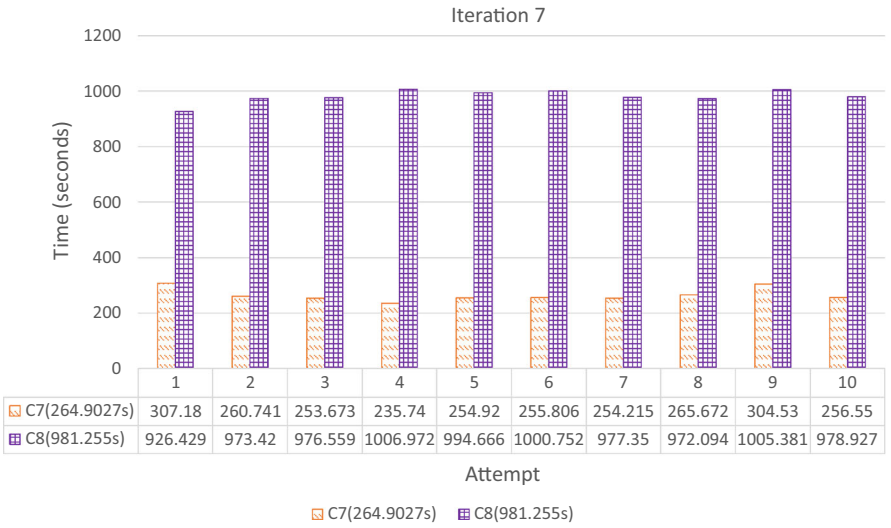
Fig. 10 Iteration 6 **a** results for C7 and C8 after executing Q3 and Q5 and **b** for C7 and C8 after executing Q4 and Q6

The main aim behind iteration 6, was to evaluate the accuracy of the UHPr when new medical fragments for a particular patient are added. In this iteration we generated an additional 40 medical fragments for our selected patient. It is also important to point out here that while the UHPr platform and the selected queries, allow for non-unique gid (i_{UHPr}), the theoretical model is based on these being unique for individual patients. As a result, the gid of the new fragments was also matched with the already existing one. The number of medical fragments for the selected patient were increased to 70 (These exist in two distinct files for UHPr, and L-Store with the split 30–40).

On executing Q1, the total number of rows returned were 140 in 32.918 s. The results indicated that each fragment id was repeated twice, which is the result of multiple "Map" operations, converging without consolidating their records. While this is not an erroneous execution, it is still undesirable for our use case. As a result, we switched the queries to Q3, Q4, Q5, and Q6. Executing in two sets of 10 repetitions each, we first calculated the results of Q3 and Q5, followed by 10 repetitions of Q4 and Q6.

In the first case, used the distinct function on the inner most query, which would produce a set of unique gid (which is 1 only), further used to retrieve the fragment ids and eventually the medical fragments. The results for this case are shown in Fig. 10a. On the other hand, Fig. 10b, shows the results of the second case, whereby the distinct function was applied on the outer query in Q4/middle query in Q6, to produce the unique set of fragment ids, eventually used for retrieving the medical fragments. The distinct operation is executed via the "Reduce" operation in Hive, which consolidates the results, leading to 70 correct records, every single time.

In iteration 7 we introduced the large dataset from EMRBOTS into the platform after tweaking it to include randomly generated patient names(a requirement for our platform). The dataset contains 100,000 new patients, along with their corresponding record of 107,535,388 fragments(from Admission, Admission Diagnosis, and Labs table). Average time for C7 is 264.9 s and for C8 s. Here again, there was a substantial



	1	2	3	4	5	6	7	8	9	10
C7(264.9027s)	307.18	260.741	253.673	235.74	254.92	255.806	254.215	265.672	304.53	256.55
C8(981.255s)	926.429	973.42	976.559	1006.972	994.666	1000.752	977.35	972.094	1005.381	978.927

Fig. 11 Iteration 7 results for C7 and C8 after executing Q4 and Q6

increase in the query execution time, as shown in Fig. 11. However the returned results were error free and conform with the platform scalability, discussed in the following section.

4.2 Evaluation

The evaluation results are presented in the following sections.

4.2.1 Timeliness

In order to evaluate the timeliness aspect of the UHPr, we analyzed the time taken in each iteration to store the medical fragments and their associated metadata into HDFS. As shown in Fig. 12a, there is a general increasing trend in the amount of time consumed, in relation with an increase in the amount of records. In iteration 1 and iteration 6, the time consumed by C1 and C3 is almost the same. For iteration 2 there is approximately 200% increase, while in iteration 3, 4, and 5 there is a 300% increase. For C2, in all iterations the difference remains within 0.402 s. This variation is explained by the increasing file size involved in each iteration, as shown in Table 4. For criteria C4, C5, C6, all six iterations showed similar execution time. This is due to the fact that in creating a table, Hive only performs basic indexing operations, thereby creating a logical schema, which is unaffected by the amount of actual data or files in the system. Figure 12b. Shows this trend, with only one corner case in iteration 1, which is most likely, an outlier. Finally, for C7 and C8, we took the average time of 10 queries, as discussed earlier and analyzed the results, which also showed a general increasing trend, till Q4 and Q6, dramatically changed the results. This trend can be explained as a by-product of an unintended optimization. The result of this analysis is

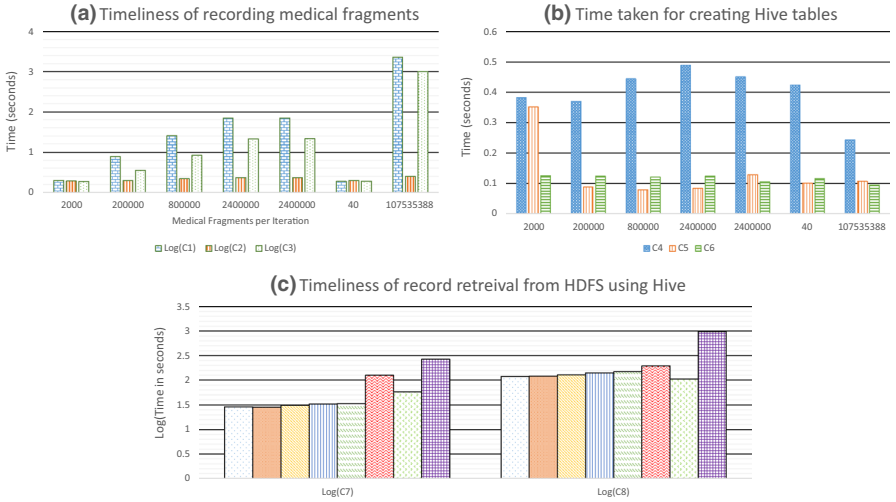


Fig. 12 **a** Timeliness of recording medical fragments and their metadata in HDFS. **b** Time taken by Hive to create tables [C4, C5, C6]. **c** Timeliness of retrieving medical fragments

Table 4 HDFS file size comparison

Ine iteration	Total medical fragments	File size for C1 (Kb)	File size for C2 (Kb)	File size for C3 (Kb)
Ine 1	2000	659	6	181
Ine 2	200,000	66,260	580	18,059
Ine 3	800,000	264,923	2320	72,242
Ine 4	2,400,000	755,295	4639	216,617
Ine 5	2,400,000	755,417	4639	216,608
Ine 6	40	13	1	4
Ine 7	116560948	25752400	7263	11118380
Ine				

shown in Fig. 12c. Summarizing these results, it is evident that the rate of increase in file size and medical records has a very small impact on the rate variations of C1, C2, C3, C7, and C8. While there is no impact on C4, C5, and C6 criteria. This indicates that the UHPr platform is able maintain timeliness of data storage and retrieval, while also supporting high scalability.

4.2.2 Scalability

As discussed earlier, from our 7 iterations, we have been able to stress test the storage platform, eventually recording over 116 million medical fragments for slightly over 390,000 patients. The storage strategy here, is very important as Hadoop and by extension Hive are really good at processing a small number of large sized files. As shown in Fig. 13, the platform is not only able to scale up when adding new patients and their associated medical fragments but has also proved successful in scaling the

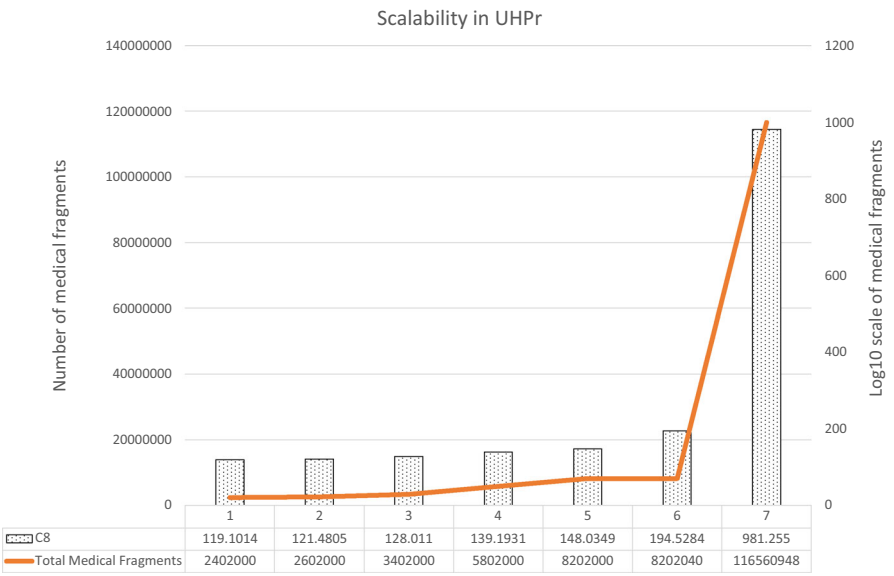


Fig. 13 Scalability in UHPr

medical fragments of an already existing patient. In particular between iteration 6 and 7, when there was a 14-fold increase in data, only 9-fold increase in querying time was observed.

4.2.3 Accuracy

For our test case of retrieving records of the user "Harry Potter" born on "19880708", the UHPr has shown 100% accuracy in all 7 iterations, albeit with some adjustment in the 6th iteration. However, even in the case where our particular query returned more results than expected, it did only double up every correct value. This has been explained earlier as a lack of consolidation for the results, which once applied, returned the correct results. Another associated caveat here is the somewhat tightly controlled nature of the sampled data. Even though the data was synthesized (partially based on 52 real patient data), producing over 116 million records, no patient with the same name and data of birth was repeated. However, in real world that may not be the case leading to the challenge of sparse data, which we will discuss in the next Sect. 6.

5 Challenges

The UHPr platform provides the necessary underlying infrastructure for resolving the interoperability problem in the field of Healthcare Informatics. However, during this long journey we were able to identify new challenges via, experimentation and peer review, some of which will be discussed henceforth.

5.1 Patient identification

Patient identification number is considered one bottleneck for cross sharing of the patient information among different medical organizations. The proposal of a single identifier across the country to identify patients in every medical organization would be one restricted solution. But the implementation of this strategy worldwide, still needs to be seen. Many covid 19 fatalities that were having underlying medical conditions could have been saved, if patient identification was performed properly across the countries. The problem of unique indexing can be explained by a simple question, raised by one of the reviewers of our work, "What happens when there are two individuals named Harry Potter and born on 19880708?" . This is one of the key research in the field of information systems. Also known as the entity resolution problem, in a Big Data environment, this problem is especially important, given the schema less storage and the large volume of items, qualifying as an entity [4].

There are two perspectives of this particular challenge. Firstly, the problem of disambiguation, whereby two different individuals from the real world, must remain so in the digital world as well. Secondly, due to sparse data, we may not always have the complete picture leading to one real world user, having multiple digital profiles. The problem might look trivial with an obvious solution to incorporate some more unique attributes like patient's address, or a hash of the patient's demographics, or an email or a phone. However, for one thing this would lead to a cyclic argument, whereby no amount of extra attributes would be enough for a universal solution. Pattern recognition technique such as the one presented in [80], which performs a similarity analysis, while keeping the computation with-in database can prove to be useful in our setting as well.

5.2 Data verification

Another challenge towards achieving complete data interoperability is the lack of a comprehensive and easy to use data verification platform. This is partially due to the veracity of medical data. As discussed in the motivation section, it is not possible to expect the over-worked medical experts to provide complete information. Instead a system of incentive based verification along with distributed voting, crowd sourcing or Blockchain could prove to be successful here. Another related aspect of this problem, is verification of semantic matching and semantic integration, in order to provide semantic reconciliation at data, process, and knowledge levels.

Data verification is made more complicated because of the occurrence of duplicate records in the patient index. Similarly, duplicates can occur when multiple records of the same patient are created in a medical system. This will not provide full medical history to the medical staff, restricting the quality of care. Confusions can also occur when same ID is provided to multiple patients. This can be very risky as the history of one of the patients should be the combination of the two patients with similar IDs. In addition, inaccuracy of data can be another challenge for data verification. For example, inaccurate data collection at the current or previous registration process at same or different medical organizations.

5.3 Security and privacy

Due to the very sensitive nature of the healthcare domain, data privacy is a major challenge, which requires implementation of very precise and comprehensive methodologies and policies for preventing any unauthorized access [16]. This includes providing an authentication and authorization procedure, maintaining integrity of the data, keeping the patient records confidential, maintaining availability, and disallowing non-repudiation [65]. Security and privacy is one of the most critical factors for any information system in general and an interoperable one in particular. This involves the questions such as whom to share, how to share, why to share, and how much to share? This also is related with another debate about who is real owner of the data (patient, one of the participating medical organizations, or all of the medical organization).

While ample solutions do exist which can help resolve this problem, identifying and using the one with least impact on the timeliness and scalability is the main concern, here. Additionally, depending upon the abstraction level at which the platform, like UHPr, is deployed, it may be necessary to take into account multiple legislation and organizational policies. e.g. compliance with Health Insurance Portability and Accountability Act of 1996 is required in the US, while in the EU medical record management systems must comply with General Data Protection Regulation 2016/679.

6 Conclusion

In this paper we presented realization of the proposed UHPr platform. The platform covers various dimensions of curating healthcare big data with big data management tools. The results showed, the platform achieving data level interoperability, by evaluating the timeliness, accuracy and scalability aspects. In summary, we have created and evaluated the UHPr platform and its associated management tools, as a proof of concept for applying semantic mediation, on semi-structured healthcare data. As a future work, we intend to extend the platform and resolve the challenges addressed in this paper.

Acknowledgements This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program(IITP-2017-0-01629) supervised by the IITP (Institute for Information & communications Technology Promotion)", by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-00655), by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2020-0-01489) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) NRF-2016K1A3A7A03951968 and NRF-2019R1A2C2090504. This research work was also supported by Zayed University RIF research fund # R18052. The authors would like to thank Uri Kartoun and EMRBots.org for generating and providing a publicly accessible large synthezied dataset (100,000 patients) to test the accuracy of the presented work. Finally, we would like to thank Professor Tae-Choong Chung (Kyung Hee University, Global Campus, Yongin, South Korea), for his valuable feedback and guidance in directing this research work.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ali T, Lee S (2017) Reconciliation of SNOMED CT and domain clinical model for interoperable medical knowledge creation. In: Proceedings of the annual international conference of the IEEE engineering in medicine and biology society, EMBS. pp 2654–2657. https://doi.org/10.1109/EMBC.2017.8037403
2. AMD (2018) AMD Ryzen 3. https://www.amd.com/en/products/apu/amd-ryzen-3-2200g
3. Askham N, Cook D, Doyle M, Fereday H, Gibson M, Landbeck U, Lee R, Maynard C, Palmer G, Schwarzenbach J (2013) The six primary dimensions for data quality assessment defining data quality dimensions. In: DAMA UK working group
4. Axelsson LE (2006) Identify user profiles in information systems with unknown users—a database modelling approach. Int J Public Inf Syst 2006(2):19–32
5. Balaur I, Saqi M, Barat A, Lysenko A, Mazein A, Rawlings CJ, Ruskin HJ, Auffray C (2017) Epigenet: a graph database of interdependencies between genetic and epigenetic events in colorectal cancer. J Comput Biol 24(10):969–980. https://doi.org/10.1089/cmb.2016.0095
6. Benaben F, Mu W, Boissel-Dallier N, Barthe-Delanoe AM, Zribi S, Pingaud H (2015) Supporting interoperability of collaborative networks through engineering of a service-based mediation information system (mise 2.0). Enterp Inf Syst 9:556–582. https://doi.org/10.1080/17517575.2014.928949
7. Berryman R, Yost N, Dunn N, Edwards C (2013) Data interoperability and information security in healthcare. In: Transactions of the international conference on health information technology advancement, vol 26
8. Bizer C, Heath T (2009) Linked data—the story so far. Int J Semant Web Inf Syst 5(3):1–22
9. Blair DR, Lyttle CS, Mortensen JM, Bearden CF, Jensen AB, Khiabanian H, Melamed R, Rabadan R, Bernstam VE, Brunak S, Jensen LJ, Nicolae D, Shah NH, Grossman RL, Cox NJ, White KP, Rzhetsky A (2013) A nondegenerate code of deleterious variants in mendelian loci contributes to complex disease risk. Cell 155(1):70–80. https://doi.org/10.1016/j.cell.2013.08.030
10. Boyd D, Crawford K (2011) Six provocations for big data. SSRN Electron J. https://doi.org/10.2139/ssrn.1926431
11. Calvanese D, Cogrel B, Komla-Ebri S, Kontchakov R, Lanti D, Rezk M, Rodriguez-Muro M, Xiao G, Corcho Ó (2016) Ontop: answering SPARQL queries over relational databases. Answering SPARQL queries over relational databases M4—Citavi. Semant Web 8(3):471–487. https://doi.org/10.3233/SW-160217T4
12. Celesti A, Fazio M, Romano A, Bramanti A, Bramanti P, Villari M (2018) An oais-based hospital information system on the cloud: analysis of a nosql column-oriented approach. IEEE J Biomed Health Inform 22(3):1–7. https://doi.org/10.1109/JBHI.2017.2681126
13. CIMI (2015) http://www.opencimi.org/
14. Clarke M, De Folter J, Verma V, Gokalp H (2018) Interoperable end-to-end remote patient monitoring platform based on IEEE 11073 phd and zigbee health care profile. IEEE Trans Biomed Eng 65(5):1014–1025. https://doi.org/10.1109/TBME.2017.2732501
15. Coorevits P, Sundgren M, Klein GO, Bahr A, Claerhout B, Daniel C, Dugas M, Dupont D, Schmidt A, Singleton P, De Moor G, Kalra D (2013) Electronic health records: new opportunities for clinical research. J Int Med 274(6):547–560. https://doi.org/10.1111/joim.12119
16. Coventry L, Branley D (2018) Cybersecurity in healthcare: a narrative review of trends, threats and ways forward. Maturitas 113(March):48–52. https://doi.org/10.1016/j.maturitas.2018.04.008
17. da Silva PR, Ferreira (2012) Enabling agents to retrieve openEHR-based health data through implementing HL7 communication with departmental information systems. https://www.semanticscholar.org/paper/Enabling-agentsto-retrieve-openEHR-based-health-Silva-Ferreira/f99e0dc31654b317516232288cf446f5f602ec97
18. De Moor G, Sundgren M, Kalra D, Schmidt A, Dugas M, Claerhout B, Karakoyun T, Ohmann C, Lastic PY, Ammour N, Kush R, Dupont D, Cuggia M, Daniel C, Thienpont G, Coorevits P (2015) Using electronic health records for clinical research: the case of the ehr4cr project. J Biomed Inform 53:162–173. https://doi.org/10.1016/j.jbi.2014.10.006
19. Denaxas SC, George J, Herrett E, Shah AD, Kalra D, Hingorani AD, Kivimaki M, Timmis AD, Smeeth L, Hemingway H (2012) Data resource profile: cardiovascular disease research using linked bespoke studies and electronic health records (caliber). Int J Epidemiol 41(6):1625–1638. https://doi.org/10.1093/ije/dys188

20. Ercan MZ, Lane M (2014) Evaluation of NOSQL databases for EHR systems. In: 25th Australasian conference on information systems, p 10
21. Fanjiang G, Grossman JH, Compton WD, Reid PP (2005) Building a better delivery system: a new engineering/health care partnership. National Academies Press, Washington, D.C. https://doi.org/10.17226/11378
22. Gallagher N (2017) Concurrent suffix tree. https://mvnrepository.com/artifact/com.googlecode.concurrent-trees/concurrent-trees
23. Gardner SP (2005) Ontologies and semantic data integration. Drug Discov Today 10(14):1001–1007. https://doi.org/10.1016/S1359-6446(05)03504-X
24. Geissbuhler A, Kimura M, Kulikowski CA, Murray PJ, Ohno-Machado L, Park HA, Haux R (2011) Confluence of disciplines in health informatics: an international perspective. Methods Inf Med 50(6):545–555. https://doi.org/10.3414/ME11-06-0005
25. Geraci A, Katki F, McMonegal L, Meyer B, Lane J, Wilson P, Radatz J, Yee M, Porteous H, Springsteel F (1991) IEEE standard computer dictionary: compilation of IEEE standard computer glossaries. IEEE Press, New York
26. Gliklich RE, Dreyer NA, Leavy MB et al (2014) Registries for evaluating patient outcomes: a user's guide, vol 13. Government Printing Office, Washington, D.C
27. Gohil P, Panchal B (2014) Efficient ways to improve the performance of HDFS for small files. Comput Eng Intell Syst 5(1):45–49
28. Gupta B, Nath R, Gopal G (2016) An efficient approach for storing and accessing small files with big data technology. Int J Comput Appl 146(1):36–39. https://doi.org/10.5120/ijca2016910611
29. Henke N, Bughin J, Chui M, Manyika J, Saleh T, Wiseman B, Sethupathy G (2016) The age of analytics: Competing in a data-driven world, vol 4. McKinsey Global Institute
30. Hemingway H, Feder GS, Fitzpatrick NK, Denaxas S, Shah AD, Timmis AD (2017a) Conclusions and implications for clinical practice and further research. In: Using nationwide 'big data' from linked electronic health records to help improve outcomes in cardiovascular diseases: 33 studies using methods from epidemiology, informatics, economics and social science in the ClinicAl disease research using LInked Bespoke studies and Electronic health Records (CALIBER) programme, NIHR Journals Library
31. Hemingway H, Feder GS, Fitzpatrick NK, Denaxas S, Shah AD, Timmis AD (2017b) Using nationwide 'big data' from linked electronic health records to help improve outcomes in cardiovascular diseases: 33 studies using methods from epidemiology, informatics, economics and social science in the clinical disease research using linked besp. Program Grants Appl Res 5(4):1–330. https://doi.org/10.3310/pgfar05040
32. Hemingway H, Asselbergs FW, Danesh J, Dobson R, Maniadakis N, Maggioni A, Van Thiel GJ, Cronin M, Brobert G, Vardas P, Anker SD, Grobbee DE, Denaxas S (2018) Big data from electronic health records for early and late translational cardiovascular research: challenges and potential. Eur Heart J 39(16):1481–1495. https://doi.org/10.1093/eurheartj/ehx487
33. HL7 (2010) Health level 7 clinical document architecture (HL7 CDA). https://www.hl7.org/implement/standards/product_brief.cfm?product_id=7
34. HL7 (2011) Health level 7 version 2 (HL7v2) product suite. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=185
35. HL7 (2017) Health level 7 version 3 (HL7v3) product suite. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186
36. HL7 (2019) Fast healthcare interoperability resources (FHIR). https://www.hl7.org/fhir/overview.html
37. Hussain M, Hussain J, Sadiq M, Hassan AU, Lee S (2018) Recommendation statements identification in clinical practice guidelines using heuristic patterns. In: 2018 19th IEEE/ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing (SNPD), vol 95, no 10, pp 152–156. https://doi.org/10.1109/SNPD.2018.8441036
38. IBM (2015) IBM integration bus healthcare pack. https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpateam&supplier=897&letternum=ENUS215-165
39. Information H, Society MS (2013) Definition of interoperability. http://www.himss.org/library/interoperability-standards/what-is
40. Ishwarappa AJ (2015) A brief introduction on big data 5vs characteristics and hadoop technology. Procedia Comput Sci 48(C):319–324. https://doi.org/10.1016/j.procs.2015.04.188
41. Jesus P, Baquero C, Almeida P (2006) ID generation in mobile environments. http://hdl.handle.net/1822/36065

42. Kartoun U, General M, Harvard H (2016) A methodology to generate virtual patient repositories. CoRR abs/1608.00570. http://arxiv.org/abs/1608.00570, 1608.00570
43. Kempe S, Booth D (2015) SmartData Webinar: yosemite project for healthcare information interoperability. https://www.dataversity.net/smartdata-webinar-the-yosemite-project-for-healthcare-information-interoperability/
44. Khan D (2015) Efficient semantic reconciliation for data interoperability among heterogeneous healthcare systems. Ph.D. thesis, Department of Computer Engineering, Kyung Hee University, South Korea
45. Kiah ML, Haiqi A, Zaidan BB, Zaidan AA (2014) Open source emr software: profiling, insights and hands-on analysis. Comput Methods Programs Biomed 117(2):360–382. https://doi.org/10.1016/j.cmpb.2014.07.002
46. Krishnan NB, Sai SSS, Mohanthy SB (2016) Real time internet application with distributed flow environment for medical IoT. In: Proceedings of the 2015 international conference on green computing and Internet of Things, ICGCIoT, 2015 pp 832–837. https://doi.org/10.1109/ICGCIoT.2015.7380578
47. Lahtiranta J (2017) Mediator—enabler for successful digital health care. Finnish J eHealth eWelfare. https://doi.org/10.23996/fjhw.60923
48. Leach P, Mealling M, Salz R (2005) Mealling Refactored Networks, LLC R. Salz DataPower Technology, Inc, 4122. https://www.ietf.org/rfc/rfc4122.txt
49. Li J (2017) A service-oriented approach to interoperable and secure personal health record systems. In: Proceedings—11th IEEE international symposium on service-oriented system engineering. SOSE 2017, pp 38–46. https://doi.org/10.1109/SOSE.2017.20
50. Liu H, Singh P (2004) Conceptnet–a practical commonsense reasoning tool-kit. BT Technol J. https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d
51. LOINC (2018) Learn LOINC. https://loinc.org/learn/
52. Maldonado JA, Moner D, Boscá D, Fernández-Breis JT, Angulo C, Robles M (2009) Linkehr-ed: a multi-reference model archetype editor based on formal semantics. Int J Med Inform 78(8):559–570. https://doi.org/10.1016/j.ijmedinf.2009.03.006
53. Maldonado JA, Costa CM, Moner D, Menárguez-Tortosa M, Boscá D, Miñarro Giménez JA, Fernández-Breis JT, Robles M (2012) Using the researchehr platform to facilitate the practical application of the EHR standards. J Biomed Inform 45(4):746–762. https://doi.org/10.1016/j.jbi.2011.11.004
54. Marcos M, Maldonado JA, Martínez-Salvador B, Boscá D, Robles M (2013) Interoperability of clinical decision-support systems and electronic health records using archetypes: a case study in clinical trial eligibility. J Biomed Inform 46(4):676–689. https://doi.org/10.1016/j.jbi.2013.05.004
55. Martínez Costa C, Menárguez-Tortosa M, Fernández-Breis JT (2011) Clinical data interoperability based on archetype transformation. J Biomed Inform 44(5):869–880. https://doi.org/10.1016/j.jbi.2011.05.006
56. Meridou D, Patrikakis C, Kapsalis A, Venieris I, Kasnesis P, Kaklamani DT (2015) An event-driven health service bus. In: MOBIHEALTH 2015—5th EAI international conference on wireless mobile communication and healthcare—transforming healthcare through innovations in mobile and wireless technologies. https://doi.org/10.4108/eai.14-10-2015.2261684
57. Mesko B (2017) The role of artificial intelligence in precision medicine. Expert Rev Precis Med Drug Dev 2(5):239–241. https://doi.org/10.1080/23808993.2017.1380516
58. MuleSoft (2020) Mule ESB. https://www.mulesoft.com/platform/soa/mule-esb-open-source-esb
59. National Institute of Health (2020) Unified modeling language system (UMLS). https://www.nlm.nih.gov/research/umls/sourcereleasedocs/index.html#
60. Nguyen HH, Mirza F, Naeem MA, Nguyen M (2017) A review on IoT healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. In: Proceedings of the 2017 IEEE 21st international conference on computer supported cooperative work in design, CSCWD 2017, pp 257–262. https://doi.org/10.1109/CSCWD.2017.8066704
61. NI of Health(NIH) (2020) National Cancer Institute (NCI) Thesaurus. https://ncithesaurus.nci.nih.gov/ncitbrowser/
62. OAEI (2020) Ontology alignment evaluation initiative (OAEI). http://oaei.ontologymatching.org/
63. Pagano P, Candela L, Castelli D (2013) Data interoperability. Data Sci J 12(0):GRDI19–GRDI25. https://doi.org/10.2481/dsj.GRDI-004
64. Pentaris F, Ioannidis Y, Manifold I (2001) Interoperability via mapping objects. Proceedings of the third DELOS network of excellence workshop on interoperability and mediation in heterogeneous digital libraries, pp 1–5

65. Priya R, Sivasankaran S, Ravisasthiri P, Sivachandiran S (2018) A survey on security attacks in electronic healthcare systems. In: Proceedings of the 2017 IEEE international conference on communication and signal processing, ICCSP 2017 2018-Janua, pp 691–694. https://doi.org/10.1109/ICCSP.2017.8286448
66. Renner SA, Scarano JG, Rosenthal AS (1996) Data interoperability: standardization or mediation. In: 1st IEEE metadata conference, pp 1–8
67. Ryan A, Eklund P (2010) The health service bus: an architecture and case study in achieving interoperability in healthcare. Stud Health Technol Inform 160(PART 1):922–926. https://doi.org/10.3233/978-1-60750-588-4-922
68. Samal L, Dykes PC, Greenberg JO, Hasan O, Venkatesh AK, Volk LA, Bates DW (2016) Care coordination gaps due to lack of interoperability in the United States : a qualitative study and literature review. BMC Health Serv Res. https://doi.org/10.1186/s12913-016-1373-y
69. Sanchez-Gomez MC, Dundon K, Deng X (2019) Evaluating data quality of newborn hearing screening. J Early Hear Detect Interv 4(3):26–32. https://doi.org/10.26077/fz0y-v617
70. Satti FA, Khan WA, Lee G, Khattak AM, Lee S (2019) Resolving data interoperability in ubiquitous health profile using semi-structured storage and processing. In: Proceedings of the 34th ACM/SIGAPP symposium on applied computing (SAC'19). ACM, pp 762–770. https://doi.org/10.1145/3297280.3297354
71. Satti FA, Ali Khan W, Ali T, Hussain J, Yu HW, Kim S, Lee S (2020) Semantic bridge for resolving healthcare data interoperability. In: 2020 International conference on information networking (ICOIN), pp 86–91. https://doi.org/10.1109/ICOIN48656.2020.9016461
72. Schulz WL, Nelson BG, Felker DK, Durant TJ, Torres R (2016) Evaluation of relational and NOSQL database architectures to manage genomic annotations. J Biomed Inform 64:288–295. https://doi.org/10.1016/j.jbi.2016.10.015
73. Schwarze K, Buchanan J, Taylor JC, Wordsworth S (2017) Are whole-exome and whole-genome sequencing approaches cost-effective? A systematic review of the literature. GENET MED. https://doi.org/10.1038/gim.2017.247
74. Shaw T, Hines M, Kielly-Carroll C (2018) Impact of digital health on the safety and quality of health care. Australian Commission on Safety and Quality in Health Care. https://apo.org.au/node/245811?utm_source=APOfeed&utm_medium=RSS&utm_campaign=rss-all
75. SNOMED (2020) SNOMED clinical terminologies. http://www.snomed.org/snomed-ct/five-step-briefing
76. Song F, Zacharewicz G, Chen D (2013) An ontology-driven framework towards building enterprise semantic information layer. Adv Eng Inform 27(1):38–50. https://doi.org/10.1016/j.aei.2012.11.003
77. Speer R, Chin J, Havasi C (2017) Conceptnet 5.5: an open multilingual graph of general knowledge. In: Proceedings of the thirty-first AAAI conference on artificial intelligence, AAAI Press, AAAI'17, pp 4444–4451
78. Sudlow C, Gallacher J, Allen N, Beral V, Burton P, Danesh J, Downey P, Elliott P, Green J, Landray M, Liu B, Matthews P, Ong G, Pell J, Silman A, Young A, Sprosen T, Peakman T, Collins R (2015) Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. PLoS Med 12(3):1–10. https://doi.org/10.1371/journal.pmed.1001779
79. Sánchez-De-Madariaga R, Muñoz A, Lozano-Rubí R, Serrano-Balazote P, Castro AL, Moreno O, Pascual M (2017) Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NOSQL approaches. BMC Med Inform Decis Mak 17(1):1–14. https://doi.org/10.1186/s12911-017-0515-4
80. Tashkandi A, Wiese I, Wiese L (2018) Efficient in-database patient similarity analysis for personalized medical decision support systems. Big Data Res 13:52–64. https://doi.org/10.1016/j.bdr.2018.05.001
81. The Apache Software Foundation (2019) Apache Hadoop. https://hadoop.apache.org/docs/stable/
82. The Apache Software Foundation (2019) Hive. https://cwiki.apache.org/confluence/display/Hive
83. The CentOS Project (2020) CentOS. https://wiki.centos.org/
84. (VeraTech for Health) (2019) LinkEHR. https://linkehr.veratech.es/research.html
85. Vujasinovic M, Ivezic N, Kulvatunyou B, Barkmeyer E, Missikoff M, Taglino F, Marjanovic Z, Miletic I (2010) Semantic mediation for standard-based b2b interoperability. IEEE Internet Comput 14(1):52–63. https://doi.org/10.1109/MIC.2010.17
86. W3C (2001) Web services description language (WSDL). https://www.w3.org/TR/wsdl.html
87. W3C (2007) Simple object access protocol (SOAP). https://www.w3.org/TR/soap12/
88. W3C (2013) SPARQL. https://www.w3.org/TR/sparql11-overview/

89. W3C - OWL Working Group (2012) OWL. https://www.w3.org/2001/sw/wiki/OWL
90. W3C - RDFCore Working Group (2014) RDF. https://www.w3.org/RDF/
91. WHO (2019) ICD-11. https://icd.who.int/icd11refguide/en/index.html
92. Zhang H, Guo Y, Li Q, George TJ, Shenkman E, Modave F, Bian J (2018) An ontology-guided semantic data integration framework to support integrative data analysis of cancer survival. BMC Med Inform Decis Mak. https://doi.org/10.1186/s12911-018-0636-4
93. Zikos D, Delellis N (2018) CDSS-RM: a clinical decision support system reference model. BMC Med Res Methodol 18(1):1–14. https://doi.org/10.1186/s12874-018-0587-6

Affiliations

Fahad Ahmed Satti[1] · Taqdir Ali[2] · Jamil Hussain[1] · Wajahat Ali Khan[3] · Asad Masood Khattak[4] · Sungyoung Lee[1]

Fahad Ahmed Satti
Fahad.satti@oslab.khu.ac.kr

Taqdir Ali
taqdirstar@gmail.com

Jamil Hussain
jamil@oslab.khu.ac.kr

Wajahat Ali Khan
w.khan@derby.ac.uk

Asad Masood Khattak
asad.khattak@zu.ac.ae

1 Ubiquitous Computing Lab, Department of Computer Engineering, Kyung Hee University, Global Campus, Yongin, South Korea

2 Division of ICT, College of Science and Engineering, Hamad Bin Khalifa University (HBKU), Education City, Doha, Qatar

3 College of Engineering and Technology, University of Derby, Markeaton Street, Derby DE223AW, UK

4 College of Technological Innovation, Zayed University, Abu Dhabi, UAE