Resolving Data Interoperability in Ubiquitous Health Profile using semi-structured storage and processing

Fahad Ahmed Satti
Ubiquitous Computing Lab,
Department of Computer Engineering,
Kyung Hee University, South Korea
fahad.satti@oslab.khu.ac.kr

Wajahat Ali Khan
Ubiquitous Computing Lab,
Department of Computer Engineering,
Kyung Hee University, South Korea
wajahat.alikhan@oslab.khu.ac.kr

Ganghun Lee
Ubiquitous Computing Lab,
Department of Computer Engineering,
Kyung Hee University, South Korea
ghl@oslab.khu.ac.kr

Asad Masood Khattak
College of Technological Innovation,
Zayed University, Abu Dhabi, UAE
asad.khattak@zu.ac.ae

Sungyoung Lee (✉)
Ubiquitous Computing Lab,
Department of Computer Engineering,
Kyung Hee University, South Korea
sylee@oslab.khu.ac.kr

ABSTRACT

Advancements in the field of healthcare information management have led to the development of a plethora of software, medical devices and standards. As a consequence, the rapid growth in quantity and quality of medical data has compounded the problem of heterogeneity; thereby decreasing the effectiveness and increasing the cost of diagnostics, treatment and follow-up. However, this problem can be resolved by using a semi-structured data storage and processing engine, which can extract semantic value from a large volume of patient data, produced by a variety of data sources, at variable rates and conforming to different abstraction levels. Going beyond the traditional relational model and by re-purposing state-of-the-art tools and technologies, we present, the Ubiquitous Health Profile (UHPr), which enables a semantic solution to the data interoperability problem, in the domain of healthcare[1].

CCS CONCEPTS

• **Applied computing ~ Health care information systems** • Information systems ~ Mediators and data integration

KEYWORDS

Healthcare, Information Systems, Semantic Mediation, Data Interoperability

SAC '19, April 8–12, 2019, Limassol, Cyprus
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297354

ACM Reference format:

F. A. Satti, W. A. Khan, G. H. Lee, A. M. Khattak, and S. Y. Lee. 2019. In *Proceedings of ACM SAC Conference, Limassol, Cyprus, April 8-12, 2019 (SAC'19)*, 9 pages. DOI: 10.1145/3297280.3297354

1 INTRODUCTION

Recent advancements in information and communication technologies have led to the rapid expansion in development, deployment and usage of policies, software and devices towards better management of healthcare services[28]. Technologies, such as whole-exome and whole-genome sequencing[40], and precision medicine[31], along with smartphone based ECG, weight and activity monitors, and continuous glucose monitors[8, 16], besides others have made the traditional physician centric healthcare systems, financially unsustainable. This has also increased the number of available alternatives and caused an improvement in the quality of healthcare support systems and by extension the healthcare services, leading to an improved patient-centric diagnostic, treatment and follow-up process[30, 41].

However, this boom, has also led to a lack of interoperability between the participating software and devices[26], increased the disparity in the quality of healthcare data[25] and created communication and coordination gaps between the medical service providers and consumers[37]. Mitigating these problems, is of utmost importance for achieving ubiquitous healthcare.

The Ubiquitous Health Platform (UHP), provides a solution to the heterogeneity problem in healthcare, by using mediation based semantic technologies, in order to resolve the differences between medical data, knowledge, processes, and devices. An abstract representation of this platform is presented in Figure 1.

Through the UHP, we aim to develop a comprehensive platform for providing standardized Ubiquitous Health Profile (UHPr); a complete digital medical persona.

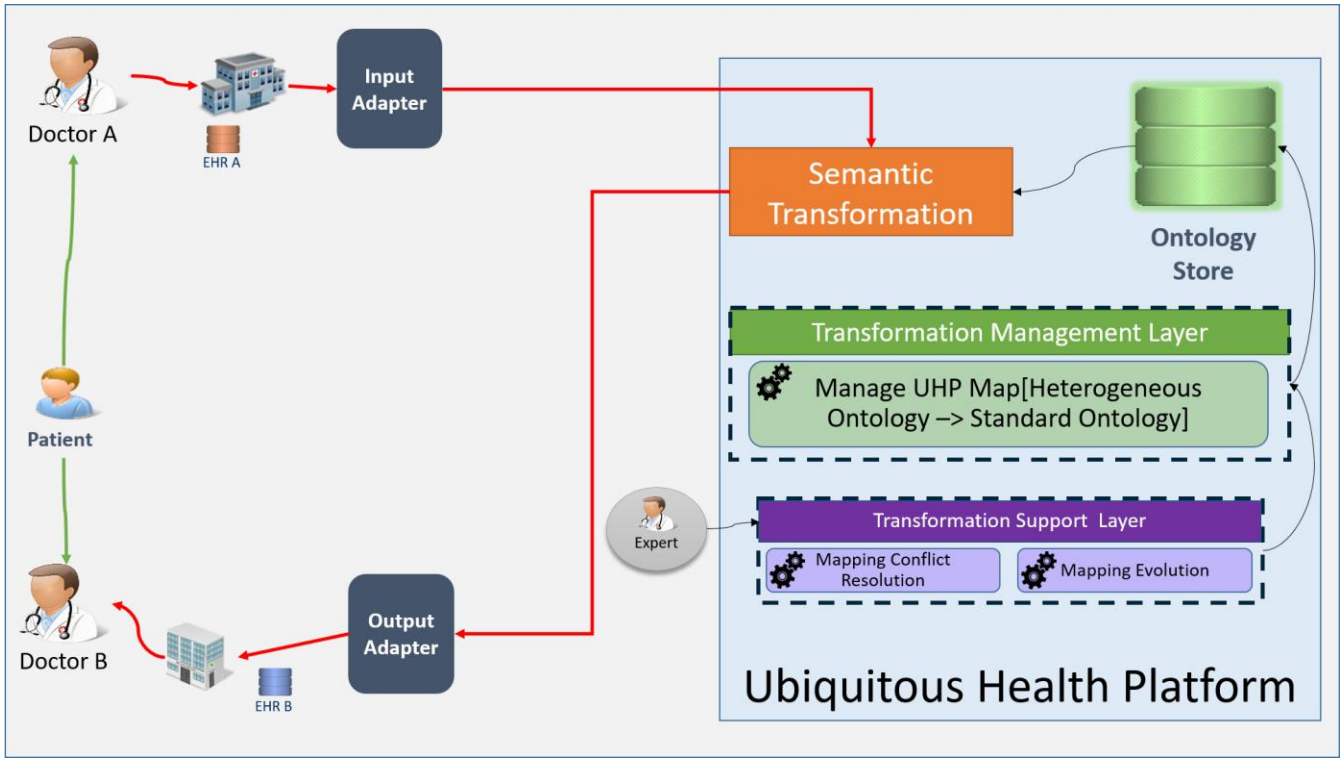

Figure 1: The Ubiquitous Health Platform

By using a graph data structure, we define the ontology maps, which are used to convert heterogeneous ontologies to a, terminology and message level, standardized one. UHP, also includes services for mapping conflict resolution, using expert intervention and a version control system for managing ontology map evolution. Details about the inner working of UHP and the motivation behind these services are out of scope of this paper.

The UHPr, represents a multi-dimensional storage structure, which is able to amalgamate medical data produced via, patient's personal input (e.g. surveys), direct intervention of the physician (e.g. exported reports from HMIS or CDSS), knowledge sources (such as the Clinical Practice Guidelines), medical IoT devices, and other sources. The main challenge behind creating, storing, and retrieving the UHPr, stem from the large volume of heterogeneous patient data, which is produced at varying rates (streaming or non-streaming), conforming to formal and non-formal, messaging standards (like HL7 v2 or FHIR), and terminological standards (such as SNOMED-CT or LOINC), and the difference in data abstraction. In order to resolve these problems, specialized storage and query engines are required[13, 24].

In this paper, we present a semi-structured, data curation methodology for the UHPr. Through a prototype implementation we also evaluated, and are presenting here, some initial results, which show a promising start towards archiving and retrieving heterogeneous medical fragments for each user, which can then be used to build a comprehensive medical history of the patient. Through this process, the medical expert will be able to view the various aspects of the patient's medical profile over a period of time. Additionally, this amalgamation of patient data can also mitigate communication gaps in healthcare service delivery.

2 MOTIVATION

2.1 Big "Healthcare" Data

Relational databases are known for their simplicity, especially as the main storage engine behind small to medium scale information systems. Consequently, in healthcare a plethora of information management systems have been designed around traditional relational database management systems (RDBMS). The driving force behind this adoption has been, easy integration with various programming languages, limited functional requirements, lack of need to share data, and the need for immediate consistency. However, with over 50% of the world population now connected with the internet[12], digital data has grown beyond the scope of the traditional RDBMS. In healthcare, this has resulted in a paradigm shift; whereby the medical experts, are starting to understand the need for scaling out, from the traditional physician and hospital/clinic centric approach to a more patient oriented one. Proprietary solutions (such as Essentia Health[2], Omni MD[3], and BlueEHR[4]), and open source HMIS (

[2] Essentia Health: http://www.essentiahealth.org
[3] https://www.omnimd.com/
[4] https://blueehr.com/our-services/electronic-health-records/

like openMRS[5] and openEMR[6]) are able to create a complete digital persona of a patient, by taking into account both direct data sources (e.g. the physician) and indirect data sources (e.g. insurance records). Yet, most solutions still use the traditional relational model of data, resulting in a lack of Big Data solutions, in healthcare[34].

Big "Healthcare" Data, represents, a non-formal characterization of Big Data in the healthcare domain. It is defined by the 5 Vs, as large "Volume" of patient data, produced at different "Velocity" (rate), adhering to a "Variety" of formal or non-formal standards, representing Veracity (different quality), and holding some implicit "Value"[23]. These attributes are very closely related to UHPr and are explained in the following subsections.

2.1.1 Volume. The UHPr consumes data from two types of data sources, primary and secondary. Primary data sources, require the direct interaction with the patient and include HMIS, Clinical Decision Support Systems (CDSS), and IoT devices. While, the secondary data sources, provide related information for the patient, but can would not require explicit input, such as general living habits, Medical Knowledge Management Systems, Biobanks, Genome data stores and others.

When integrated with Biobanks (like UK Biobank with 500,000 participants[43]), disease studies (such as the mendelian disorder risk study with 100 million participants[6]), clinical research systems (EHR4CR project with 45 partners in EU[32]) and medical devices (producing streaming data using body sensors), the storage requirements, scale beyond the scope of any traditional RDBMS, and require the use of specialized data curation solutions.

2.1.2 Velocity. Similarly, streaming data produced by medical devices, presents a challenge in terms of its timeliness, integration, and storage[29, 33]. As a case study, consider a smart watch based heartrate monitor, which produces many instances of very shallow data, while the EHR produced by an HMIS is infrequent, longitudinal, and holds more detailed data. The key requirement, for streaming data is low latency, while for non-streaming data, high reliability, is preferred.

2.1.3 Variety. Variety in UHPr can be defined in terms of, the associated data format and its purpose. Healthcare systems adhere to heterogeneous formats, which can be standard based[28] or custom[15]. Several semantic reconciliation tools and techniques, already exist which can resolve the interoperability problem in EHRs[27]. Collection of healthcare data, also suffers from a variety of purpose. Medical information systems can be categorized based on their target users, which can be patients (e.g. continuous glucose monitor), medical experts (e.g CDSS), organizations (e.g. hospital management information systems, insurance claim systems), or environment (e.g. public health systems)[10]. As a result, these systems, only produce data limited to their own abstraction level. As a result, the data collected by a medical research institute, is at a very different abstraction level, then what would be collected by a private clinic.

2.1.4 Veracity. Another related challenge in UHPr, is that of low quality. The domain of healthcare, lacks any golden ontology, which can be used to standardize all EHRs. On the contrary, EHRs remain valid, while they remain associated with their schema conforming information system, but are not useful, outside of it. This is due to the use of very different terminological and messaging standards and the existence of non-formal custom standards. For quantitative data, this problem can be diluted by the high volumes of medical data, but same is not true for qualitative data[7]. LinkedEHR can be used to create a common platform for integrating, primary and secondary data, leading to better support for diagnosis and treatment[11, 20].

2.1.4 Value. Extraction of meaningful, implicit value from UHPr is another challenge. Due to the distributed nature of the storage engine, and in presence of eventual consistency, the UHPr should be able to mimic the accuracy of traditional healthcare systems, while also providing new insights, resulting from the integration of medical data[21].

2.2 Data Interoperability

Heterogeneity in healthcare, is a major challenge, which prevents integration, exchange and effective utilization of medical data, across system boundaries. The key to solving this problem lies in identification of relationships between the participating schemas, which can be achieved by using schema matching and schema mapping approaches[28]. However, due to the variety in format for healthcare systems, this task requires the use of semantic technologies, which could be categorized into standard based and mediation based approaches. The standard based approach, uses mutually agreeable standards, while the mediation based approach creates context based translations, from the source to target schemas, and vice versa[36]. Linked Data uses the standard based approach, for creating such semantic links and resolving the heterogeneity problem[5], while Semantic Information Layer (SIL)[42] uses the mediation based approach for achieving data interoperability in Enterprise Information Systems (EIS).

Implementation of these technologies towards achieving data interoperability in healthcare, can not only benefit the physician and the patient by reducing overhead and redundant costs and saving time, but can also prevent operational waste, and support policy makers in improving accountability and privacy[4].

In UHPr, data interoperability is achieved via a mediation based approach, which creates the UHP map from Figure 1. The resultant transformations are verified by measuring the data quality in terms of its timeliness, completeness, uniqueness, validity, consistency, and accuracy[1]. These attributes are defined (w.r.t UHPr), as follows:

- *Timeliness* ─ An event, is recorded in real-time, after it has occurred (Inverse is the time difference between an event occurring in the real world and to it being recorded).

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

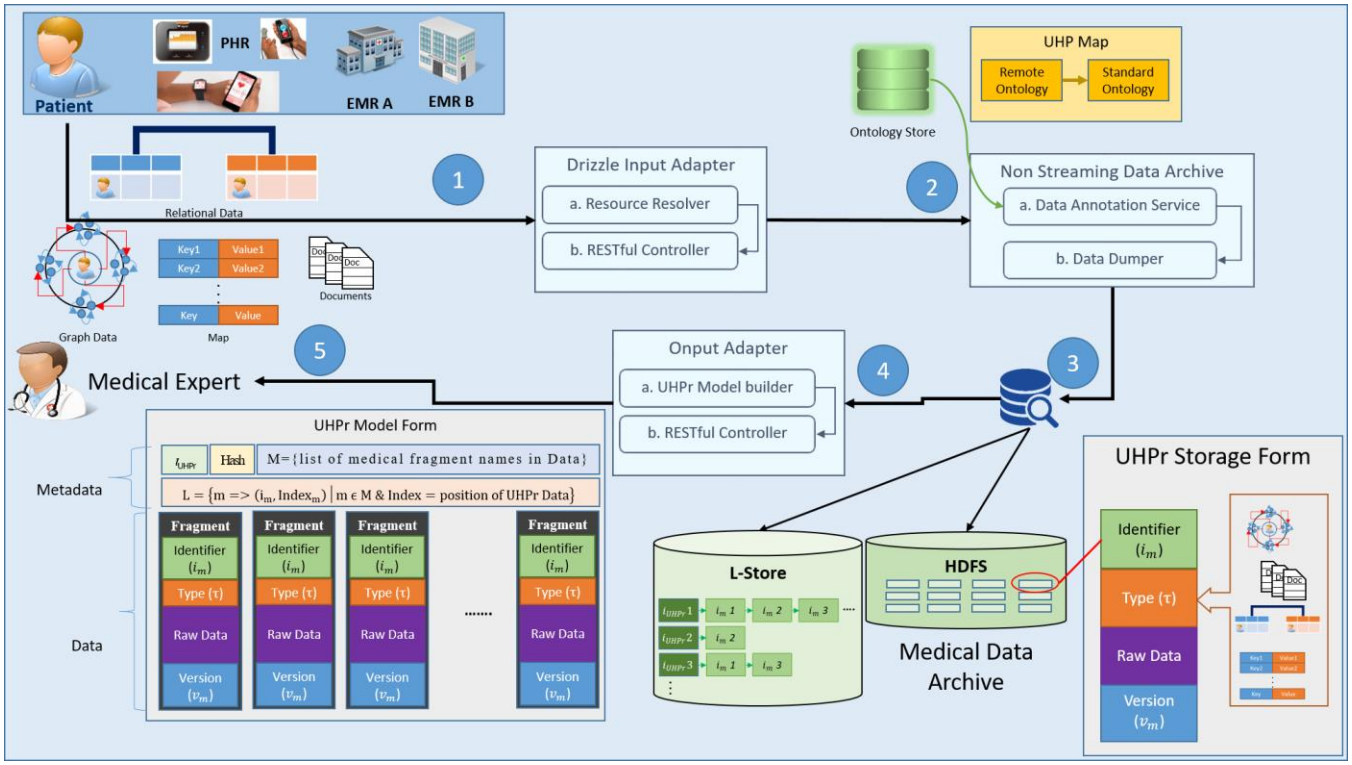

Figure 2: The UHPr Storage and Model forms

- *Completeness* — All the data, pertaining to an event is recorded. (Inverse is the difference between what is recorded and what could/should have been recorded).
- *Uniqueness* — Recording each individual event, once
- *Validity* — The UHPr schema/container and the UHPr data, conforms to a standard.
- *Consistency* — Recorded UHPr data is similar to what is expected/possible.
- *Accuracy* — UHPr Data is recorded, accurately and mirrors the characteristics of the real world event.

2.3 Healthcare Data Storage Solutions

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 able to achieve Atomicity, Consistency, Isolation, and Durability; otherwise known as ACID properties. This kind of storage is beneficial for small to medium scaled medical systems, with little to no interoperability. However, with the emergence of unstructured and semi-structured data, along with high speed networks and more complex data models, medical systems are now focusing on using NoSQL technologies[14, 38, 39]. NoSQL data bases represent the set of non-relational distributed databases, which focus on providing scalability, availability and partition tolerance in context of the CAP theorem[17]. As a result, NoSQL databases are only able to provide Basic Availability, can exist is Soft state, but will always Eventually become consistent[2], also known as the BASE properties.

Healthcare data, based on their schema can be categorized into four parts, Relational data, Column Oriented data, Graph data, Documents and Key-Value maps.

2.3.1 Relational Medical Data. Medical systems, using a RDBMS as data store, provide this data. Here the data instances are uniquely identifiable with primary keys, and also contain foreign keys to identify their relationships, with other instances.

2.3.2 Column-Oriented data. Specialized data stores which organize their data into columns rather than rows, to optimize column-oriented operations are kept in this category. Implementation of medical systems using column-oriented data stores have shown high scalability and improved performance in comparison with traditional RDBMS, and provide a viable alternative[9].

2.3.3 Graphs. Medical systems, which provide data in the form of nodes and their associated edges, fall in this category. Due to these relationships, which are not bound by any schema, data presented in graph form, has more expressability, semantics and scalability than the relational model. When optimized, graph can allow fast insertions and traversal exploration[3].

2.3.3 Documents. NoSQL databases, that use a schema-less approach to store data in documents and knowledge sources (such as clinical practice guidelines) can provide very fast insertions and scalability. The read operations are relatively slower, especially if documents are unstructured, requiring deep searches[39].

2.3.3 Key-Value maps. The simplest form of medical data, which holds completely unstructured data, providing the fastest insertion speed, and slowest querying speed. While it is not typically used as a data store, it can be used to store streaming data, from medical IoT devices.

3 Methodology

The UHPr, represents a logical amalgamation of medical data, which represents a complete digital persona of the patient. This model and implementation details of the prototype UHPr data structure is presented in the following subsections.

3.1 UHPr version 1

The UHPr, has two forms, storage and model. Where the former represents a semi-structured form, kept in a Big Data storage platform, while the later represents an integrated, volatile form, which is consumed by medical platforms and experts to extract value.

As shown in Figure 2, Patient data is collected from a variety of source, using the drizzle input adapter (which can handle non-streaming data only. Streaming data will be supported in later implementations). The data is mapped to a standard ontology form using the UHP Map from the ontology store. This is then send to the data curation service, identified as step 3, in Figure 2. Here the UHPr is converted to its storage form, whereby data from one of the supported types (relational, graph, document, column, or key-value) is wrapped in a data structure, that also contains a unique identifier for the medical fragment, its type (same as above) and version information (used for managing mapping evolution). Additionally, user identifier (typically the MR number), is extracted from the incoming data, along with patient firstname, lastname, and date of birth. This information is then used to update indexes in the L-Store, which provides a naïve, logical indexing service for the medical fragments. The L-Store, generates a 128 bit, uuid, the i_{UHPr}, which is mapped against the user's firstname, lastname, and date of birth. The i_{UHPr} is then used to build an adjacency list with related medical fragment identifiers (i_m). A possible, semi-relational Entity Relationship Diagram (ERD) for the L-Store is shown in Figure 3.

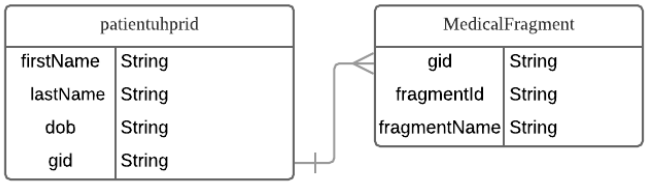

Figure 3: L-Store ERD

When querying for a patient's data, the UHPr model builder, uses the L-Store to identify the required iUHPr, followed by the related medical fragment. For general queries, not related to a specific patient, this process is skipped and MapReduce algorithms are used to extracting information. Conversion to the UHPr Model form, requires the build-up of metadata, which contains, the $i_{UHPr,}$ Hash of the UHPr model, list of medical fragments names, for the user, and the Identifier Map (*L*). The identifier map, provides an index for the UHPr model. It contains key-value pairs of all medical fragments contained in the UHPr model. Fragment names, form the key, while their unique identifiers from UHPr storage form and position/index in the fragment list, forms the value. This can be used to quickly iterate over the medical fragments, and filtering only required value, based on fragment names. Finally, the UHPr model is delivered to the medical expert, containing comprehensive medical history for the patient. The two key elements, in UHPr, are the combination of different data formats into one composite data structure and the use of semi-structured data store to maintain logical indexing.

3.2 UHPr version 1 Prototype

In order to validate the the UHPr model, an initial prototype has been developed, which integrates patient data coming from OpenEMR patient records and a custom implementation of, expert driven medical diagnostic system, the KRSilo. Using Hadoop[7] as the data storage and processing engine, the UHPr storage form and L-Store is kept in HDFS. From here, Hive is used to keep L-Store in memory, and to create temporary schemas for retrieving medical fragments. Initial seeding for the L-Store is done using pseudo-random sampling, based on 40 private patient records (which we are not allowed to make public), from local hospitals. The sample size was 80,000 patient records and 10 medical fragments with different versions for each of the two participating systems. Since Hive does not support primary keys or foreign keys, the ERD from Figure 3, is converted into a hive table, using the following queries:

```
> create external table patientuhprid(firstName string,
lastName string, dob string, gid string) row format
delimited fields terminated by ',' lines terminated by '\n'
location '/Lstore/patientuhprid';

> create external table medicalfragment(gid string,
fragmentId string, fragmentName string) row format
delimited fields terminated by ',' lines terminated by '\n'
location '/Lstore/medicalfragment';
```

In HDFS, each individual medical fragment is stored in a directory, identified by the related user's, uuid based "gid" field. This ensures, that medical fragments of each user are kept under the directory 'UHP/$gid'. The UHPr storage form is converted into UHPr model, by creating a temporary relation in Hive, which only exists till the connection remains active.

[7] Our custom deployment is composed of, 1 master and 2 slave nodes, with 1.8TB hdfs size, 20MB block size, Block Replication of 3, and 64GB ram on the master, while 32GB on the slaves.

```
hive> create temporary external table uhpr(identifier string, fragmentname string, data string, version
 string) row format delimited fields terminated by '|' lines terminated by '\n' location '/UHPv2/UHP/e0
ac4ccc-8c7e-49a5-af9d-0550ba9ae542.csv';
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. MetaException(messa
ge:Got exception: org.apache.hadoop.fs.FileAlreadyExistsException Path is not a directory: /UHPv2/UHP/e
0ac4ccc-8c7e-49a5-af9d-0550ba9ae542.csv
```

Figure 4. HIVE error, on populating tables from files (not a folder given in path)

Using the "gid" from L-Store, the following query provides schema-on-read for the medical fragments.

```
> create temporary external table uhpr(identifier string, fragmentname string, data string, version string) row format delimited fields terminated by '|' lines terminated by '\n' location '/UHP/609cc551-8197-494a-813d-ee318dd933d7' ;
```

Finally, the data from the Hive table is extracted and pushed onto the UHPr model builder, which converts it into JSON form and sends it to the UHPr consumer.

3.3 Results from UHPr version 1 implementation

Our initial results, resulted in the formation of very large number of small sized files. For 80,000 patients with 20 fragments, a total of 1,500,000 medical fragments were created. This change caused the hdfs insertion and bulk retrieval process to become very slow. A screenshot of the list all directory command, with approx. 7-9 folders shown per minute, is shown in Figure 5.

```
23:19 /UHP/00105c8f-4075-40ad-8478-65154dd0009c
23:20 /UHP/00125494-ced4-4184-94b2-649b8802c0d5
23:20 /UHP/0013def5-f478-437b-b946-55a43042a8c1
23:20 /UHP/0014458d-a893-4b4b-90cd-2b003d514d61
23:20 /UHP/00149b9a-c54d-40ee-90c5-24dd8a9705ff
23:20 /UHP/0016117a-1e7d-4ab1-bb00-7a4b17af32a1
23:20 /UHP/0018c4e8-3491-401d-a24a-629ddd1dc2ff
23:20 /UHP/0019012e-8eff-4976-80db-70ac20ddcd12
23:20 /UHP/001a2086-b5d7-4a64-95c6-f4a4b9501225
23:20 /UHP/001aba60-6baa-4145-9a95-82eb765f9e2e
23:21 /UHP/001c4e8f-5e47-4a3b-b8b3-a6ce7d83a673
```

Figure 5. Slow indexing in HDFS

Additionally, it is also evident from literature that the MapReduce operation employed by Hive is also very slow, when dealing with large number of very small files[18, 19]. In order to resolve this problem, the storage strategy for UHPr was changed to concatenate the medical fragments of every individual user into an independent file, leading to the creation of UHPr version 2.

3.4 UHPr version 2

The UHPr, version 2, stores the medical fragments of each patient into individual files. In this iteration, we got 500 files, corresponding to 500 sampled patients (In version 1, this would generate 10,000 files), which were inserted in HDFS. The insertion process, now finishes in less than 2 minutes at a rate of 360 files/min. The process is even faster, since HDFS does not create extra directories. However, when trying to create temporary schemas on HIVE for building the UHPr model, we encountered another challenge. HIVE can create a table and use the location property to load data from a directory, creating records for each file, however it cannot load the contents from a subset of files in the directory(open JIRA issue HIVE-951[22]) only and throws an error, as shown in Figure 4.

In order to resolve this problem, we loaded the complete UHPr version 2 storage data set into a temporary HIVE table using the following query:

```
> create temporary external table uhpr(identifier string, fragmentname string, data string, version string) row format delimited fields terminated by '|' lines terminated by '\n' location '/UHPv2/UHP';
```

As shown in Figure 6, an aggregate query, was executed, to count the number of records in the newly created uhpr table. A total of 10,000 rows were counted using MR job, which took 67.649 seconds.

The next step of the UHPr model building process is to select all medical fragments for one patient only. Since the "uhpr" hive table does not contain the gid, which is the patient identifier, we have to create a JOIN query which can combine these 10,000 fragments with the "fragmentId" and "gid" fields of the L-Store's MedicalFragment table, shown in the Figure 3. This query is as follows:

```
> select f.identifier, f.fragmentname, f.data, f.version from medicalfragment l join uhpr f on (l.fragmentid = f.identifier) where l.gid = '2f69eb4e-c35c-4763-954a-a04eeba501fe';
```

This query took 91.181 seconds with the serialized "data" column and returned the correct 20 fragments. However, due to limitation of space, the result of the same query without the "data" column is shown in Figure 7, which took 86.384 seconds.

These results provide a proof of concept, and partially validate the effectiveness of our ubiquitous health profile, storage model, using a semi-structured storage and processing engine.

```
hive> select count(identifier) from uhpr;
Query ID = hduser1_20180925021504_de21ea21-29b9-431a-8d4c-e8f87111a81c
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
2018-09-25 02:15:06,623 INFO  [9b65ecf8-ccc7-451b-97f0-97bafdae81b2 main] client.RMProxy: Connecting to ResourceManager at master/10
.13.13.102:8050
2018-09-25 02:15:07,635 INFO  [9b65ecf8-ccc7-451b-97f0-97bafdae81b2 main] client.RMProxy: Connecting to ResourceManager at master/10
.13.13.102:8050
Starting Job = job_1537514790612_0002, Tracking URL = http://localhost:8088/proxy/application_1537514790612_0002/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1537514790612_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-09-25 02:15:22,143 Stage-1 map = 0%,  reduce = 0%
2018-09-25 02:15:42,459 Stage-1 map = 20%,  reduce = 0%, Cumulative CPU 11.41 sec
2018-09-25 02:15:48,765 Stage-1 map = 36%,  reduce = 0%, Cumulative CPU 17.1 sec
2018-09-25 02:15:54,021 Stage-1 map = 52%,  reduce = 0%, Cumulative CPU 22.76 sec
2018-09-25 02:16:00,314 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 28.06 sec
2018-09-25 02:16:09,904 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 30.34 sec
MapReduce Total cumulative CPU time: 30 seconds 340 msec
Ended Job = job_1537514790612_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 30.34 sec   HDFS Read: 5263921 HDFS Write: 105 SUCCESS
Total MapReduce CPU Time Spent: 30 seconds 340 msec
OK
_c0
10000
Time taken: 67.649 seconds, Fetched: 1 row(s)
```

Figure 6. Count of all rows in the uhpr table

```
hive> select f.identifier, f.fragmentname, f.version from medicalfragment l join uhpr f on (l.fragmentid = f.identifier) where l.gid = '2f69eb4e-c35c-
4763-954a-a04eeba501fe';
Query ID = hduser1_20180925024225_560798dc-c448-42c4-b634-d43a5878b6f8
Total jobs = 1
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
2018-09-25 02:42:46,157 INFO  [9b65ecf8-ccc7-451b-97f0-97bafdae81b2 main] client.RMProxy: Connecting to ResourceManager at master/10.13.13.102:8050
2018-09-25 02:42:46,183 INFO  [9b65ecf8-ccc7-451b-97f0-97bafdae81b2 main] client.RMProxy: Connecting to ResourceManager at master/10.13.13.102:8050
Starting Job = job_1537514790612_0006, Tracking URL = http://localhost:8088/proxy/application_1537514790612_0006/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1537514790612_0006
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2018-09-25 02:42:59,620 Stage-3 map = 0%,  reduce = 0%
2018-09-25 02:43:19,625 Stage-3 map = 18%,  reduce = 0%, Cumulative CPU 11.72 sec
2018-09-25 02:43:25,949 Stage-3 map = 34%,  reduce = 0%, Cumulative CPU 17.54 sec
2018-09-25 02:43:32,220 Stage-3 map = 49%,  reduce = 0%, Cumulative CPU 23.26 sec
2018-09-25 02:43:38,486 Stage-3 map = 65%,  reduce = 0%, Cumulative CPU 29.04 sec
2018-09-25 02:43:43,731 Stage-3 map = 81%,  reduce = 0%, Cumulative CPU 34.81 sec
2018-09-25 02:43:49,990 Stage-3 map = 98%,  reduce = 0%, Cumulative CPU 40.62 sec
2018-09-25 02:43:51,040 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 41.53 sec
MapReduce Total cumulative CPU time: 41 seconds 530 msec
Ended Job = job_1537514790612_0006
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 41.53 sec   HDFS Read: 5260183 HDFS Write: 2217 SUCCESS
Total MapReduce CPU Time Spent: 41 seconds 530 msec
OK
f.identifier    f.fragmentname  f.version
889ce4bf-f80e-4605-bc15-4960658337ac    krsiloemr_tblpatient    85c2eb52-b9fc-48e6-9fb5-d117eba3548e
f829568f-3229-49cf-a2ed-c50047abdcec    krsiloemr_tblpatient    27399b31-53fd-4d04-a4a7-305e3f3fae96
a7f95536-9243-4424-b16d-eea47f3250b8    krsiloemr_tblpatient    e1df3c69-6db0-4da3-974a-961649a86041
af4d5f1f-24f1-491b-8cb5-7f207b7a690f    krsiloemr_tblpatient    5c08f2d4-214f-4439-b31d-797892e348d0
da733484-bb9b-4ae1-a526-92f72d0cd866    krsiloemr_tblpatient    11c87cb9-26f0-4150-8209-36d96c52f380
2ae8286f-3724-47dc-b494-b643b277bac6    krsiloemr_tblpatient    c941c1c4-9263-428f-a22c-cbff9105a459
74883d76-ccda-4f37-b69e-b26159b22d73    krsiloemr_tblpatient    a7c7e26a-2a56-4dd1-8681-66df45e666d5
796abb10-9cda-474f-a126-15ccbe9e98da    krsiloemr_tblpatient    57374dda-1a78-4369-89d6-b5c97d957b74
2d0ebf53-8da1-4c60-8a22-708e9ac20644    krsiloemr_tblpatient    e866e6c5-9ef6-4e21-947e-e73a02940fea
380cd7eb-c7fc-4dd7-8851-f842641a937b    krsiloemr_tblpatient    61282077-f76b-4b2f-96c3-ce170b448023
8ff8e1f3-1073-4617-9b25-69bcd19f19b9    openemr_patientdata     b1a8f8ca-ba7a-41b4-b269-914d268ff7ff
54c1fa4f-5ff2-4bb3-a8e8-ce60e2f992f4    openemr_patientdata     232a3218-3275-4942-aa58-a5869047ca42
c8bc22b6-5a1e-487f-acc1-d0820b37333b    openemr_patientdata     322c5eff-da36-47be-9d22-f17ae47b0e80
eb2d3c94-69ce-410f-bccc-bb89ed498329    openemr_patientdata     00977bdf-06c3-43a2-9beb-a35588b5ce61
8765e417-958d-4526-a798-d2e49b52177e    openemr_patientdata     9b3c9126-39dd-4e6a-b703-892ac62dea4a
e544b496-d82b-47f1-b144-be1d3ffeb27b    openemr_patientdata     b613e541-47b1-431d-abdf-29bf25478c8f
4f9e0db7-241d-4ba0-ae5a-94858d08f309    openemr_patientdata     fa465c09-58b2-480a-9c6e-f90726875c8c
f8492663-afd3-44c8-9b94-d50005352c00    openemr_patientdata     db82d931-58b9-48ba-a7b2-63ef6cc46c34
0492158a-642c-4acb-9223-4659ed10c984    openemr_patientdata     4e49cded-baf7-40f1-8970-47bd16c984d9
1d63428f-86fa-416b-b5a6-48e08267b865    openemr_patientdata     4d71728f-9f63-4a84-8743-4ba96315fe50
Time taken: 86.384 seconds, Fetched: 20 row(s)
```

Figure 7. Result of retrieving all medical fragments, except for the data part, for a patient

4 CONCLUSIONS

Data interoperability in healthcare is a major challenge, which can be resolved using the Ubiquitous Health Platform. UHP provides mediation based semantic reconciliation for heterogeneous healthcare data. A key requirement for this platform to work is the use of semi-structured storage solution, which can handle the scalability requirements associated with the large volume of medical data. It should also provide mapping for resolving the variety problem, between different data formats, and formal and non-formal, schema standards. While we were faced with many challenges in the form of a large number of small sized files being generated in version 1 of the UHPr, and HIVE only populating data from folders, in UHPr version 2. We were eventually able to create a prototype implementation of the UHPr which is able to provide the a very basic version of the previously mentioned services. Additionally, a number of modules have to be implemented, like services for handling streaming data, better L-Store indexing mechanisms, and feedback for improving the veracity of medical data. Moreover, while we have extrapolated our test samples from real data, access to healthcare data, and real world testing is very necessary to identify any problems, before the UHP becomes production ready. The future direction involves finding solutions to these problems and integrating UHP with other medical platforms.

ACKNOWLEDGMENTS

This research was supported by an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean government (MSIT) (No. 2017-0-00655). This work 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) and NRF- 2016K1A3A7A03951968.

REFERENCES

[1] Askham, N. et al. 2013. THE SIX PRIMARY DIMENSIONS FOR DATA QUALITY ASSESSMENT Defining Data Quality Dimensions. *DAMA UK Working Group*. (2013).

[2] Bailis, P. and Ghodsi, A. 2013. Eventual consistency today. *Communications of the ACM*. 56, 5 (2013), 55. DOI:https://doi.org/10.1145/2447976.2447992.

[3] Balaur, I. et al. 2017. EpiGeNet: A Graph Database of Interdependencies Between Genetic and Epigenetic Events in Colorectal Cancer. *Journal of Computational Biology*. 24, 10 (2017), 969–980. DOI:https://doi.org/10.1089/cmb.2016.0095.

[4] Berryman, R. et al. 2013. Data Interoperability and Information Security in Healthcare. (2013).

[5] Bizer, C. and Heath, T. 2009. Linked data - the Story so Far. *International Journal on Semantic Web and Information Systems*. 5, 3 (2009), 1–22.

[6] Blair, D.R. et al. 2013. A nondegenerate code of deleterious variants in mendelian loci contributes to complex disease risk. *Cell*. 155, 1 (2013), 70–80. DOI:https://doi.org/10.1016/j.cell.2013.08.030.

[7] Boyd, D. and Crawford, K. 2011. Six Provocations for Big Data. *A decade in Internet time: symposium on the dynamics of the Internet and society* (2011), 1–17.

[8] Cappon, G. et al. 2017. Wearable Continuous Glucose Monitoring Sensors: A Revolution in Diabetes Treatment. *Electronics*. 6, 3 (2017), 65. DOI:https://doi.org/10.3390/electronics6030065.

[9] Celesti, A. et al. 2018. An OAIS-based hospital information system on the cloud: Analysis of a NoSQL column-oriented approach. *IEEE Journal of Biomedical and Health Informatics*. 22, 3 (2018), 1–7. DOI:https://doi.org/10.1109/JBHI.2017.2681126.

[10] Dale Compton, W. et al. 2005. *Building a Better Delivery System: A New Engineering/Health Care Partnership*.

[11] Denaxas, S.C. et al. 2012. Data resource profile: Cardiovascular disease research using linked bespoke studies and electronic health records (CALIBER). *International Journal of Epidemiology*. 41, 6 (2012), 1625–1638. DOI:https://doi.org/10.1093/ije/dys188.

[12] Digital in 2017: Global Overview: 2017. *https://wearesocial.com/sg/blog/2017/01/digital-in-2017-global-overview*.

[13] Divyakant Agrawal, U.S.B. et al. 2012. *Challenges and Opportunities with Big Data: A white paper prepared for the Computing Community Consortium committee of the Computing Research Association*.

[14] Ercan, M.Z. and Lane, M. 2014. Evaluation of NoSQL databases for EHR systems. *25th Australasian Conference on Information Systems*. (2014), 10.

[15] Geissbuhler, A. et al. 2011. Confluence of disciplines in health informatics: An international perspective. *Methods of Information in Medicine*. 50, 6 (2011), 545–555. DOI:https://doi.org/10.3414/ME11-06-0005.

[16] Gia, T.N. et al. 2017. IoT-based continuous glucose monitoring system: A feasibility study. *Procedia Computer Science*. 109, (2017), 327–334. DOI:https://doi.org/10.1016/j.procs.2017.05.359.

[17] Gilbert, S. and Lynch, N. 2012. Perspectives on the CAP Theorem. *Computer*. 45, 2 (2012), 30–36. DOI:https://doi.org/10.1109/MC.2011.389.

[18] Gohil, P. and Panchal, B. 2014. Efficient Ways to Improve the Performance of HDFS for Small Files. *Computer Engineering and Intelligent Systems*. 5, 1 (2014), 45–49.

[19] Gupta, B. et al. 2016. An Efficient Approach for Storing and Accessing Small Files with Big Data Technology. *International Journal of Computer Applications*. 146, 1 (2016), 36–39. DOI:https://doi.org/10.5120/ijca2016910611.

[20] Hemingway, H. et al. 2017. Conclusions and implications for clinical practice and further research. *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*. 221–236.

[21] Hemingway, H. et al. 2017. 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. *Programme Grants for Applied Research*. 5, 4 (2017), 1–330. DOI:https://doi.org/10.3310/pgfar05040.

[22] HIVE-951>Selectively include EXTERNAL TABLE source files via REGEX: *https://issues.apache.org/jira/browse/HIVE-951*. Accessed: 2018-09-24.

[23] Ishwarappa and Anuradha, J. 2015. A brief introduction on big data 5Vs characteristics and hadoop technology. *Procedia Computer Science*. 48, C (2015), 319–324. DOI:https://doi.org/10.1016/j.procs.2015.04.188.

[24] Kadadi, A. et al. 2014. Challenges of data integration and interoperability in big data. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*. (2014), 38–40. DOI:https://doi.org/10.1109/BigData.2014.7004486.

[25] Kahn, M.G. et al. 2015. Transparent Reporting of Data Quality in Distributed Data Networks. *eGEMs (Generating Evidence & Methods to improve patient outcomes)*. 3, 1 (2015), 7. DOI:https://doi.org/10.13063/2327-9214.1052.

[26] Khan, D.W.A. 2013. *Efficient Semantic Reconciliation for Data Interoperability among Heterogeneous Health-care Systems*.

[27] Khan, W.A. 2015. *E FFICIENT S EMANTIC R ECONCILIATION FOR D ATA I NTEROPERABILITY AMONG H ETEROGENEOUS H EALTH - CARE S YSTEMS H ETEROGENEOUS H EALTH - CARE S YSTEMS*. Kyung Hee University.

[28] Kiah, M.L.M. et al. 2014. Open source EMR software: Profiling, insights and hands-on analysis. *Computer Methods and Programs in Biomedicine*. 117, 2 (2014), 360–382. DOI:https://doi.org/10.1016/j.cmpb.2014.07.002.

[29] Krishnan, N.B. et al. 2016. Real Time Internet Application with distributed flow environment for medical IoT. *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015*. (2016), 832–837. DOI:https://doi.org/10.1109/ICGCIoT.2015.7380578.

[30] Lahtiranta, J. 2017. Mediator – enabler for successful digital health care. *Finnish Journal of eHealth and eWelfare*. 9, 4 SE-Scientific articles (Nov. 2017). DOI:https://doi.org/10.23996/fjhw.60923.

[31] Mesko, B. 2017. The role of artificial intelligence in precision medicine. *Expert Review of Precision Medicine and Drug Development*. 2, 5 (2017), 239–241. DOI:https://doi.org/10.1080/23808993.2017.1380516.

[32] De Moor, G. et al. 2015. Using electronic health records for clinical research: The case of the EHR4CR project. *Journal of Biomedical Informatics*. 53, (2015), 162–173. DOI:https://doi.org/10.1016/j.jbi.2014.10.006.

[33] Nguyen, H.H. et al. 2017. A review on IoT healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. *Proceedings of the 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design, CSCWD 2017*. (2017), 257–262. DOI:https://doi.org/10.1109/CSCWD.2017.8066704.

[34] Nicolaus, H. et al. 2016. The age of analytics: Competing in a data-driven world. *McKinsey Global Institute*. December (2016), 136. DOI:https://doi.org/10.1111/bjet.12230.

[35] Pagano, P. et al. 2013. Data Interoperability. *Data Science Journal*. 12, 0 (2013), GRDI19-GRDI25. DOI:https://doi.org/10.2481/dsj.GRDI-004.

[36] Renner, S.A. et al. 1996. Data interoperability: Standardization or Mediation. *1st IEEE metadata conference*. (1996), 1–8.

[37] Samal, L. et al. 2016. Care coordination gaps due to lack of interoperability in the United States : a qualitative study and literature review. *BMC Health Services Research*. (2016), 1–8. DOI:https://doi.org/10.1186/s12913-016-1373-y.

[38] Sánchez-De-Madariaga, R. et al. 2017. Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: Relational vs. NoSQL approaches. *BMC Medical Informatics and Decision Making*. 17, 1 (2017), 1–14. DOI:https://doi.org/10.1186/s12911-017-0515-4.

[39] Schulz, W.L. et al. 2016. Evaluation of relational and NoSQL database architectures to manage genomic annotations. *Journal of Biomedical Informatics*. 64, (2016), 288–295. DOI:https://doi.org/10.1016/j.jbi.2016.10.015.

[40] Schwarze, K. et al. 2018. Are whole-exome and whole-genome sequencing approaches cost-effective? A systematic review of the literature. *GENETICS in MEDICINE*. 00, August 2017 (2018). DOI:https://doi.org/10.1038/gim.2017.247.

[41] Shaw, T. et al. 2017. *Impact of Digital Health on the Safety and Quality of Health Care*.

[42] Song, F. et al. 2013. An ontology-driven framework towards building enterprise semantic information layer. *Advanced Engineering Informatics*. 27, 1 (2013), 38–50. DOI:https://doi.org/10.1016/j.aei.2012.11.003.

[43] Sudlow, C. et al. 2015. UK Biobank: An Open Access Resource for Identifying the Causes of a Wide Range of Complex Diseases of Middle and Old Age. *PLoS Medicine*. 12, 3 (2015), 1–10. DOI:https://doi.org/10.1371/journal.pmed.1001779.