Designing A Health Data Management System Based Hadoop-Agent

Fadoua KHENNOU[1], Youness Idrissi KHAMLICHI[2], and Nour El Houda CHAOUI[3]

[1,3]TTI Laboratory, Higher School of Technology
[2]REIS Laboratory, Faculty of Science and Technology
Sidi Mohamed Ben Abdellah University, Fes
fadoua.khennou@usmba.ac.ma, youness.khamlichi@usmba.ac.ma, houda.chaoui@usmba.ac.ma

***Abstract*—Nowadays the amount of data that is being generated every day is increasing in a high level for various sectors. In fact, this volume and diversity of data push us to think wisely for a better solution to store, process and analyze it in the right way. Taking into consideration the healthcare industry, there is a great benefit for using the concept of big data, due to the diversity of data that we are dealing with, the extant, and the velocity which lead us to think about providing the best care for the patients.**

In this paper, we aim to present a new architecture model for health data. The framework supports the storage and the management of unstructured medical data in a distributed environment based on multi-agent paradigm. The integration of the mobile agent model into hadoop ecosystem will give us the opportunity to enable instant communication process between multiple health repositories.

***Index Terms*—healthcare, big data, mobile agents, hadoop, unstructured data.**

I. Introduction

The concept of big data is experiencing a significant growth in many sectors and proving crucial benefits with the use of its technologies. Bringing big data into practice is the main focus for researchers nowadays [1], and among the leading industries that still require detailed studies, is the healthcare [2]. Indeed, large amount of medical data is generated and become available in the health organizations, thus need to be wisely managed in order to get proper understanding and execute further analysis.

Recently, we observe major results which aim to improve patient's care, telemedicine services and hospital outcome. As a matter of fact, in the last four years, the spotlight of this area research was related mainly on analytic studies [3, 4] and leveraging health data in order to get an insight and make the right decision, by identifying chronic and re-emerging infectious diseases, knowing patient's lifestyle and habits, which helps to drive an optimal care in the appropriate time. These analytic studies [5, 6] are performed on systems called electronic health records (Ehr), which are represented as large entities that store the data collected from many medical organizations. Their main purpose is to make information instantly accessible for the appropriate stakeholders. Along with that, health entities store their usual data in softwares illustrated as the electronic medical records (Emrs), which are very small repositories and defined as a digital version of paper.

Our goal throughout this paper, is to give an insight of our e-health workflow. It is initiated by the acquisition of the data from medical organizations, and finalized by proposing an adequate storage system. The remainder of this paper is organized as follows. Section II presents the related works. System description and the storage processing techniques will be discussed in Section III. Section IV describes our proposed approach of the mobile agent based hadoop framework and a case study scenario. Finally, Section V provides a summary and conclusion.

II. Related Work

There has been an increasing number of research studies, whose main purpose is to conceive an adequate system for monitoring health data. In fact, in the health sector, the data can be processed according to various levels: predictive analytics, which allow for an estimated criteria to predict the risk of getting diseases for some patients based on genomes [7], vital signs monitoring [8, 9], which represents an interesting thematic overlooked patients whose condition can change at any time, in addition to the analysis of health data present in social media networks [10].

All these mentioned thematics were developed recently by several researchers, in order to find appropriate solutions that can raise up the health sector in its high level [11]. Indeed, some studies shed light on the implementation of a multi-agent system, able to monitor patients remotely by designing agents that can retrieve the desired information and send them back to physicians [12], other approaches were proposed so as to assist professionals and physicians to accomplish a decision making support [13], improve the safety of patient's data, retrieve information from heterogeneous sources and analyze medical information in order to provide a quality care for patients.

The conception of such systems, with the use of multi-agent paradigm, offers new solutions of mobility, flexibility, effectiveness and portability. Since, the electronic health data are generally represented in a distributed and heterogeneous framework, and the need for coordinating the healthcare activities is prominent. The authors in [13], have presented

978-1-5090-0751-6/16/$31.00 ©2016 IEEE

a new approach of context-aware mobile agent for emergency applications. Their architecture involves agents placed for the ambulatory services and others for hospitals, in order to get instant patient's data and prepare a prior admission in the adequate hospital. The authors in [14], have also put into practice communicating agents for the real time monitoring of the vital signs of the patients. Thus, allow an extensive care with the use of telemedicine.

Yet, after going in depth through all these mobile-agent applications for healthcare benefit, we encounter a huge lack in terms of storage techniques and optimal processing models for these massive data, which push us to think toward the combination of mobile agent paradigm as a communication process and big data technologies in order to insure an optimized storage and processing system as a platform.

III. Suggested electronic health record

In this part of our paper, we will present our system needs and describe its components in depth. In the last section, we will go through a comparison study of the most common open source projects in order to demonstrate our choice.

A. System description

The translation of our management system rules towards an effective implementation is quite a difficult task, since it requires the study of the different stakeholders that need to be deployed into the system, namely the collection of the data from hospitals and the storage in a centralized server, that will be accessible by the medical organizations. In the fig.1, we present our vision toward the Ehr management system needs and specifications.

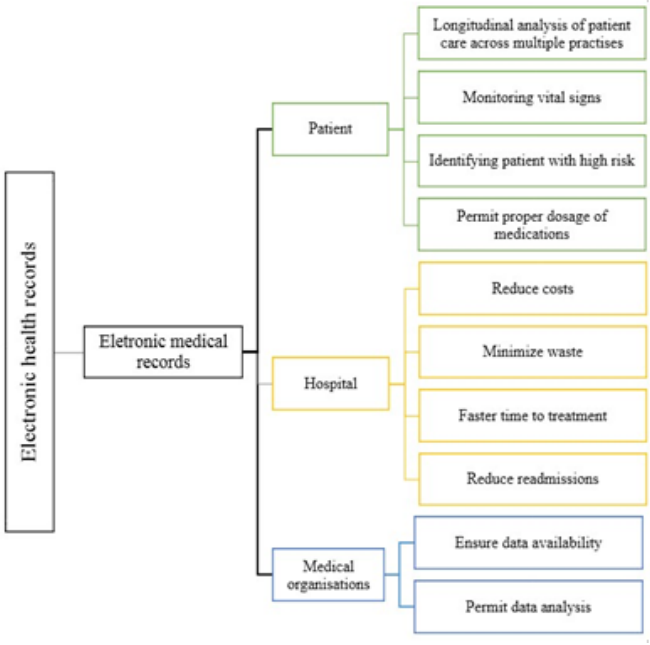

Figure 1: Ehr management system needs

This diagram shows roughly the functions that has to be implemented in our system. In fact, the objectives need to be achieved concern not only the patient's benefits, but also hospitals and medical organizations.

For that, physician nurses and medical staff should be able to work fluently with the system, by analyzing previous states of patient's history, identifying patients whose risk is high and allow adequate drug dosage in a momentary state. Once this is achieved, medical organizations will gain systematically in terms of cost and waste reduction, in addition to an optimized time for processing data.

B. Architecture overview

Our proposed architecture presents a workflow between different Emrs and a centralized Ehr system. In this part of our project, we focus on the conception of a system that has the ability to share health data between Emrs and update the Ehr repository. For that, we will present firstly our architecture overview and then shed light on the Electronic medical record systems. Here, we consider developing countries as a case study, as they generally suffer from a lack of organization in the health sector. Along with that, there is an excessive consumption of medical examinations, laboratory tests and diagnoses, which are often unnecessary and costly.

1) System workflow

The architecture in the fig.2, describes a general workflow and a communication process between multiple databases gathered from clinics.

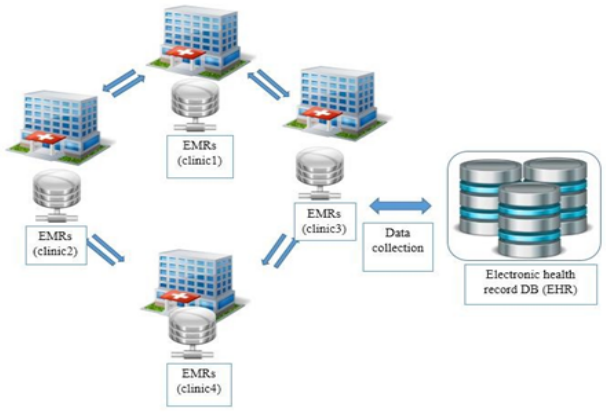

Figure 2: Data flow between (Ehr-Emrs)

Each health establishment will have its own database noted as Emr, this database stores data limited to a specific clinic and is usually related to a single practice. The aim of this first process, is to track the data of all patients who are registered in a specific health organization, and ensure their effective care. For more advanced analytic services, the electronic health record system describes a centralized database that will store the data gathered from several medical organizations (Emrs). That is to say, it presents a personalized and advanced view of all the patients and their medical history.

2) Emrs characteristics

The electronic medical records are represented among the crucial entities of our system. In our approach, the purpose

Table I: Emrs open source projects

Projects	Category	Complexity	Certification	Multi platform	Practices	Database	Language	Modules
Openmrs	Emr	High	✓	✓	Medium	MySQL	Java	Form entry, Clinical summary, Rest API, Compare list, Birt reporting Jasper Reports
Gnuhealth	Emr Health information system	Medium	✓	X	Large	Postgre-SQL	Python	Epidemiology, Genetics, Laboratory, Hospital, Pediatric, Surgery, Radiology, Hermato, Oncology
OpenEmr	Emr	Low	✓	✓	Medium	MySQL	PHP	Complete Ambulatory Certified EHR, Patient, Demographics, Scheduling, Prescriptions,Billing, Clinical Decision Rules
FreeMED	Emr	Low	X	✓	Small	MySQL	PHP	Patient, Document, Billing, Reporting, appointment
Drchrono	Mobile Emr	Low	✓	✓	Small	MySQL	Python	medical icd-9, search engine electronic medical insurance, iPhone ICD-9 search engine

is not only limited to the selection of the ideal Emr, but to present the mostly used projects.

Furthermore, since we need an interoperable system, the selection of a unique emr system is not advisable. Indeed, our interest is to conceive a platform which can maintain a fluid data flow communication between multiple systems regardless of their difference and characteristics.

In Table I, we have listed the commonly used Emr systems. Each open source project, has been developed in different modules that support either small, medium or large practices. It depends whether they will be implemented only for the administration's needs or even for diagnoses, prescriptions (Gnuhealth), ambulatory services (OpenEmr) and more.

Taking as an example the Openmrs project, which is widely used as an Emr system, it can be personalized in order to meet different medical organization's needs. Yet, it is not completely developed as a final product, which may require further knowledge and efforts to set it up.

While some projects are characterized by a complex ease of use, others can be simply managed and implemented for the organization. As for the certification criteria, some systems have been certified for their flexibility, ease of use and adaptability of the modules that they offer.

Regarding our approach, we will base our study on the open source projects that have been developed for medium to large practices, as this will help us go through the implementation of a communication process between different modules of different systems.

C. Big data technologies

In the Health industry, the structure of data is the number one factor that is usually questioned. In fact, it represents a real challenge for IT professionals, as there is a rising need for conceiving adequate storage and processing technologies in order to manage the variety of data. Nevertheless nowadays, many open source projects have been put into practice in order to tackle such challenges [15].

We present in Table II, a comparative study of the main open source projects for the big data systems and describe their specifications and use cases.

Table II: Big data open source projects

Projects	Performance	Use case	Modules
Hadoop	Scalability Fault-tolerance processing on disk	Large volume of data Distributed storage and processing	Hive Mahout Giraph Hbase Impala
Storm	Scalability Fault-tolerance Stream processing	Real time analytics Online machine learning distibuted RPC Etl	Kinesis AMQP Kestrel RabbitMQ Kafka JMS
Spark	Memory processing dStream processing	Fast engine prcossing scale data Mlibx Graphx	Shark sql Spark streaming GraphX MLBase HDFS
Sector	Efficient data access wide area networks High security	Data across multiples datacenters Upload/download from remote locations	Sphere UDT MalStone

Storm is an efficient stream processing system. Yet, in our first case study, which concerns processing over complex and voluminous data, we need the execution of a batch platform for the Ehr repository. Spark can do both streaming and micro-batch treatment, although it uses the hadoop distributed file system in order to operate. The particularity of Sector is that it supports processing over multiple data centers across wide area networks. Still, it uses C++ as a programming language

and the Sector/Sphere community is not widely developed yet as it is the case with the other open source projects.

We will focus on the hadoop project, since it represents a combined and complete ecosystem which already includes all the modules, along with the possibility to add other components once needed.Through hadoop, we can perform complex processing [16] of high volume of data, in the centralized Ehr repository.

Using batch processing with hadoop will give us the opportunity to store all health data acquired from multiples medical bodies and process them in batches. Along with that, we can perform an initial synchronization across heterogeneous Emrs.

IV. Implementation methodology

Now that we have selected multiples Emrs for health organizations, and the hadoop ecosystem for the centralized Ehr repository. We can study and analyze our main issue, which concerns the data flow communication process between the different stakeholders.

The electronic medical records implemented for hospitals, clinics, and other health systems, should interact continuously with each other regardless of their difference, so as to insure the interoperability in the platform. And while hadoop provides us with the batch processing method, in order to process a huge amount of health data at once, we need methods and mechanisms that will help us provide a communication data flow between all the parties. For that, the mobile agent paradigm has been proved through many studies to be the adequate solution [17].

A. Mobile agent paradigm

A mobile agent is defined as a software that has the ability to move through the network in a specific environment. It has the potential of bringing both code and data to be executed in different machines. The main goal that pushed us to use this paradigm is related primarily to its specific properties, which can be beneficial to our health platform. We present below the useful assets:

Cooperation: Able to interact with other agents. This is the case of several agents whose function is to index patient data in a specific clinic. Indeed, the recovered data must be assembled and finally stored by a different agent into the Ehr repository.

Autonomy: Able to react without the intervention of a third party. In our case, this represents an important criteria due to the need of implementing instant patient's history recovery when needed. In fact, they can either perceive their environment and react right away or take the initiative in order to proceed in the right time.

Adaptation: While managing health data, we are faced to operate in a distributed and heterogeneous environment with the use of different type of Emrs, that is to say, we need agents able to accommodate in different situations.

Mobility: As defined previously, this characteristic allows the agent to travel through the network in order to execute independently a precise function.

Our architecture model includes multiple agents, each one is responsible for a specific operation. As a first use case in the fig.3, we present an activity diagram of a communication process between the main agents:

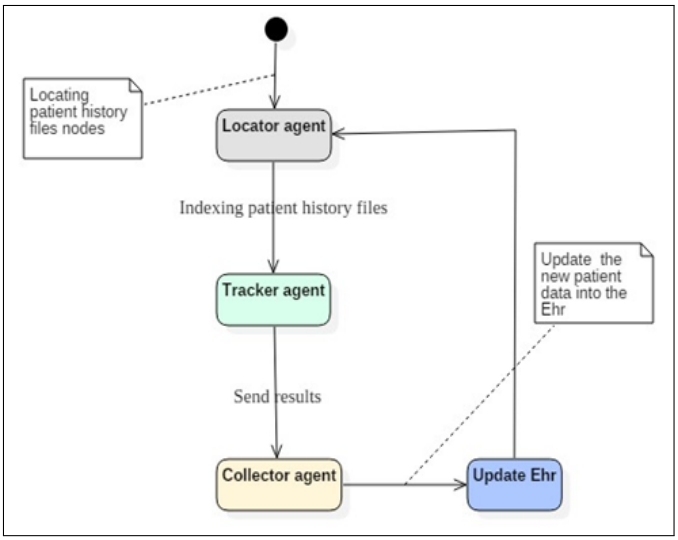

Figure 3: Data flow using proposed mobile agents activity diagram

The distributed structure of our system requires cooperative, adaptive and flexible agents. They are defined as follows:

Locator agent: In order to ensure the best support care for the patients, we will consider an agent whose function is to search and index data once the patient get in the hospital. In fact, this function will not only be performed within the same hospital but also within different Emrs. The aim of this agent, is to help physicians to have a clear idea of his previous and present state.

Updater agent: Typically our framework includes different Emrs, along with the hadoop ecosystem as a storage system. For that, this agent's behavior concerns the execution of scheduled update queries, in order to insure that our centralized repository will be up to date regarding every practice, diagnoses, hospital admissions and any new medical process. This will allow us to gather a massive amount of data for all patients through time, and will help us to get a clear history of all patients.

Tracker agent: The tracker agent will have the unique behavior of verifying whether the data, indexed by the locator agent, is already updated in the Ehr repository or not. And due to the fact that hospitals contain myriad departments and medical services, we need to retrieve all patient's data gathered from each department.

We consider a scenario, where a patient has been admitted to a hospital and need to get an emergent heart surgery. That's to say, physicians need to get immediately his latest medical laboratory tests and all his diagnoses.

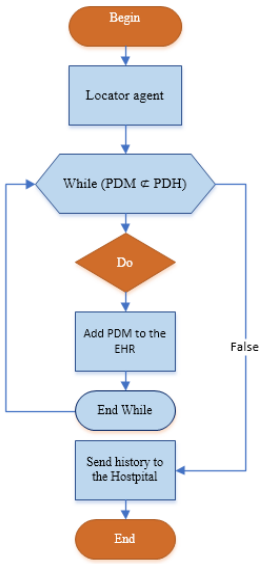

Figure 4: Tracker agent flowchart

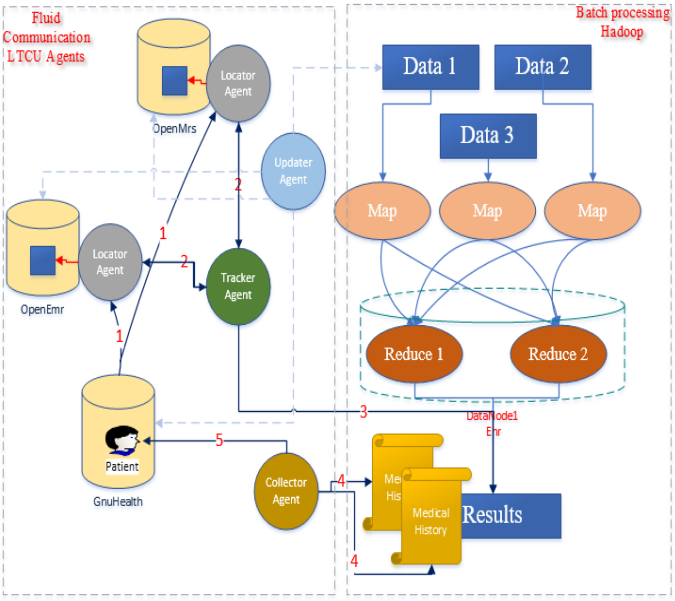

Figure 5: Parallel Mapreduce-agent paradigm

In the flowchart above, we present a data flow that will be processed according to the function defined by the while condition : $(PDM \nsubseteq PDH)$

The PDM indicates the patient's data indexed in the emrs by the locator agent, and the PDH indicates the patient's data already present in the Ehr.

On the one hand, we verify if the Ehr repository already includes the data found by the locator agent. Normally, the updater agent should have previously refreshed the Ehr system with the latest updates. Yet, due to the important amount of data that we are dealing with, we may encounter a slight delay for the update function to be executed because of a node failure or a batch processing delay which is usually calculated in terms of minutes.

On the other hand, if the updated data related to any new diagnoses, practices or laboratory tests have been successfully added. The collector agent will communicate the patient history, which will be systematically generated from the hadoop ecosystem and sent back to the requested hospital.

B. Integrated MA with Mapreduce paradigm

The authors in [18], have proposed the integration of mobile agent into the map reduce paradigm MRAM and proved it to be an interesting approach for improving big data analytics. Yet, their framework is based on implementing new features into hadoop in order to enhance its performance, thus supporting specific factors of tasks scheduling, disk management and resource allocation. In our context, we shed light on designing a framework that implements a parallel execution of the mapreduce paradigm and mobile agent system, because our main purpose is to conceive an interoperable system that can support a fluid data flow between different relational databases and hadoop ecosystem.

As shown in fig.5, we can divide our architecture into two parts. The first one concerns the prompt communication between medical organizations through their emrs. In fact, each emr is different from another and may be covering medium to large practices. Along with that, they support different types of relational databases. The specific properties of these agents is that they can execute code independently of any type of database or system. The three main agents will communicate in order to localize, index, track, verify and add new updates of the patient's data.

The locator agent will firstly locate the desired data of the patient in Emrs, present in different medical bodies. Once the data is localized in one or several Emrs, we verify if the Ehr repository already includes the data found by the locator agent. If not, the tracker agent will add those new updates to the centralized repository (Ehr).

In the second part of this process, the updater agent executes programmed revisions to the Ehr system. The Mapreduce functions will process every newly added health data into the repository and generate history reports, which can be exploited at any time in order to explore all historical data through time.

C. Mapreduce use case

Among the three agents presented above, we consider in the fig.6, the case of the updater agent and the execution of its related mapreduce functions.

Hadoop uses MapReduce technique to create tasks that can be executed independently of each other. In MapReduce, data is always read or written in the format <key, value>.

In our example, the map function reads a record that comes in the form of a pair <Patient, data>. In fact, We consider many patients who have their data stored in many emrs, in each data block we apply the map function in order to retrieve a list of objects indexed by a patient identification Key. By

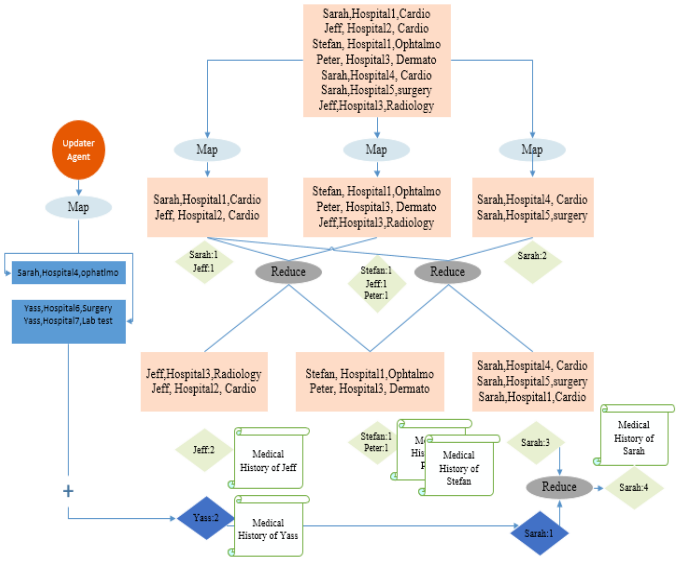

Figure 6: Parallel Mapreduce-updater agent execution

then, The system will rearrange the data in order to group objects that have the same key. For each group defined by the patient identification key is called the reduce function(), which regroup every value corresponding to the criteria. Finally, the system provides the output with the defined key and the corresponding values.

For each output provided, we can generate patient's medical history data, which can help physicians explore the past and present state of the patients. Along with that, patient's data will be stored in a centralized repository which is beneficial in terms of reducing costs and minimizing the consumption of unnecessary diagnoses or laboratory tests.

The particularity about the updater agent, is his ability to launch an automatic map call to be executed in the input data. While calculating an average time between two updates. Indeed, the scheduled updates should be implemented several times in order to reduce the workload for the mappers, which will allow the reduce function, to be launched simply by adding the results to the previous outputs, as presented in the scheme.

V. Conclusion

In this paper, we presented our e-health framework for health data management. We described firstly our platform, which combines multiple Emrs implemented for every health organization, and an Ehr repository as a centralized system. Secondly, we focused on the proposal of the adequate mobile agents, in order to ensure data sharing between various Emrs.

Our proposed approach is based on solving, on one side, the problems of storage and processing over an important amount of health data, by using hadoop ecosystem. On the other side, it helps us to maintain a smooth sharing of medical data between different organizations, throughout the intelligence of our proposed agents.

In our future work, we intend to implement our designed system in a practical case study, which requires a powerful server for the centralized Ehr system, along with distributed parallel machines as a replicated hard disk storage. As for analytic processing, we aim to leverage the hadoop real time processing tools and agent's intelligence, in order to analyse the state of patients across multiple practices.

References

[1] C. K. Emani, N. Cullot, and C. Nicolle, "Understandable big data: A survey," *Computer Science Review*, vol. 17, pp. 70 – 81, 2015.

[2] T. Huang *et al.*, "Promises and challenges of big data computing in health sciences," *Big Data Research*, vol. 2, no. 1, pp. 2 – 11, 2015, special Issue on Computation, Business, and Health Science.

[3] V. Vijayakumar *et al.*, "Big data, cloud and computing challenges a survey of big data analytics in healthcare and government," *Procedia Computer Science*, vol. 50, pp. 408 – 413, 2015.

[4] F. A. Batarseh and E. A. Latif, "Assessing the quality of service using big data analytics: With application to healthcare," *Big Data Research*, 2015.

[5] V. Chandola, S. R. Sukumar, and J. C. Schryver, "Knowledge discovery from massive healthcare claims data," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: ACM, 2013, pp. 1312–1320.

[6] H. A. Reijers *et al.*, *Business Process Management Workshops: BPM 2009 International Workshops, Ulm, Germany, September 7, 2009. Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ch. Workflow for Healthcare: A Methodology for Realizing Flexible Medical Treatment Processes, pp. 593–604.

[7] V. Vijayakumar *et al.*, "Big data, cloud and computing challenges predictive methodology for diabetic data analysis in big data," *Procedia Computer Science*, vol. 50, pp. 203 – 208, 2015.

[8] T. W. Kim *et al.*, "A big data framework for u-healthcare systems utilizing vital signs," in *Computer, Consumer and Control (IS3C), 2014 International Symposium on*, June 2014, pp. 494–497.

[9] A. Page *et al.*, "Visualization of health monitoring data acquired from distributed sensors for multiple patients," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.

[10] M. M. Cruz-Cunha *et al.*, "Health twitter big bata management with hadoop framework," *Procedia Computer Science*, vol. 64, pp. 425 – 431, 2015.

[11] J. Andreu-Perez *et al.*, "Big data for health," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 4, pp. 1193–1208, July 2015.

[12] V. Chan, P. Ray, and N. Parameswaran, "Mobile e-health monitoring: an agent-based approach," *IET Communications*, vol. 2, no. 2, pp. 223–230, February 2008.

[13] F. Burstein, P. D. Haghighi, and A. Zaslavsky, *Decision Support: An Examination of the DSS Discipline*. New York, NY: Springer New York, 2011, ch. Context-Aware Mobile Medical Emergency Management Decision Support System for Safe Transportation, pp. 163–181.

[14] W.-S. Hsu and J.-I. Pan, "Secure mobile agent for telemedicine based on p2p networks," *Journal of Medical Systems*, vol. 37, no. 3, pp. 1–6, 2013.

[15] H. Hu *et al.*, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652–687, 2014.

[16] N. Nagaveni *et al.*, "Graph algorithms, high performance implementations and its applications (icghia 2014) performance analysis of agent based framework," *Procedia Computer Science*, vol. 47, pp. 37 – 44, 2015.

[17] R. S. Gray *et al.*, "Mobile-agent versus client/server performance: Scalability in an information-retrieval task," in *Proceedings of the 5th International Conference on Mobile Agents*, ser. MA '01. London, UK, UK: Springer-Verlag, 2002, pp. 229–243.

[18] Y. M. Essa, G. Attiya, and A. El-Sayed, "Mobile agent based new framework for improving big data analysis," in *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on*, Dec 2013, pp. 381–386.