Wiki-Health: A Big Data Platform for Health Sensor Data Management

Yang Li
Imperial College London, United Kingdom

Chao Wu
Imperial College London, United Kingdom

Li Guo*
University of Central Lancashire, United Kingdom

Chun-Hsiang Lee
Imperial College London, United Kingdom

Yike Guo
Imperial College London, United Kingdom

ABSTRACT

Quickly evolving modern technologies such as cloud computing, Internet of Things and intelligent data analysis have created great opportunities for better living. We visualize the role these technological innovations will play in the health care sector as spearheading a shift in focus from offering better health care services only to people with problems, to helping everyone achieve a healthier lifestyle. In this article, we first discuss the existing and potential barriers followed by an in-depth demonstration of a service platform named Wiki-Health that takes advantage of cloud computing and Internet of Things for personal well-being data management. It is a social platform which is designed and implemented for data-driven and context-specific discovery of citizen communities in the areas of health, fitness and well-being. At the end of the article, we analyse a case study to illustrate how the Wiki-Health platform can be used to serve a real world personal health training application.

INTRODUCTION

Recent developments in modern technologies such as cloud computing, wearable sensor devices and big data have significantly impacted people's daily lives, and offer great potential for an Internet-wide, people-centric ecosystem that will considerably extend human capabilities in acquiring, consuming and sharing personal information. In particular, these new capabilities will address a vital aspect of living – the practice and implementation of personal health care and well-being. Humans are actually becoming super organisms with support from this ecosystem. For instance, with the latest mobile devices we will be able to see and review our personal health status through backend analysis services using continuously collected data from those wearable body sensors; we could make complex decisions using our computing-aided "brains".

People have always been searching for the most accurate information to empower themselves for a healthier life. Social media has changed the nature of interactions among people and organizations. According to PwC's consumer survey of 1,060 U.S. adults, about one-third of consumers are using the

* *Please direct all your inquires to the corresponding author Dr. Li Guo at lguo@uclan.ac.uk*

social space as a natural forum for health discussions (Admins, 2011). We share our lives and thoughts with the social community or the public. We depend on our networks to help us make many decisions. We seek connection and access. However, the health and wellbeing industry has been slow to embrace social media due to often insurmountable issues of privacy, data protection, but now is beginning to see the benefits.

Meanwhile, the rapidly growing popularity of smartphones and tablets globally has created many opportunities for growth within the health care and wellbeing sector. These devices provide new ways to gather information, both manually and automatically, over wide areas. Many current smart phones come with a number of embedded sensors such as microphones, cameras, gyroscopes, accelerometers, compasses, proximity sensors, GPS and ambient light. The newer generation of professional wearable medical sensors can easily connect with the smart phones and transfer the sensing results directly. This has provided a more efficient and convenient way to collect personal health information like blood pressure, oxygen saturation, blood glucose level, pulse, Electrocardiogram (ECG), Electroencephalogram (EEG) and electrocardiography (EKG). A future in which we are all equipped with devices and sensors that passively collect and interpret our health and activity data is not too far off. The scale and richness of mobile sensor data being collected and analysed are rapidly expanding. This massive growth creates both data manageability and collaboration challenges.

Traditional sensor network systems increasingly face many issues and challenges regarding their communication and resources management including data storage, data query, data processing, privacy control, and data sharing. The emergence of cloud computing is seen as a remedy for these issues and challenges. Implementing cloud computing technologies appropriately can help healthcare providers improve the quality of medical services and the efficiency of operations, share information, improve collaboration, and manage expenditures.

In this chapter, we present our work – a service platform named Wiki-Health that takes advantage of cloud computing and Internet of things for personal well-being data management. Wiki-Health is a social platform for data-driven and context-specific discovery of citizen communities in the areas of health, fitness and well-being. Wiki-Health provides new ways of storing, tagging, retrieving, analysing, comparing and searching health sensor data. It makes health-related knowledge discovery available to individual users at a massive scale. Wiki-Health is based on the concept of the wiki and mass collaboration, according to which aggregated user-contributed content is collectively curated to produce unprecedented volumes of knowledge in a multi-perspective and socially engaging way. One of key research points for Wiki-Health is to deliver novel algorithms for data-driven community discovery in large health datasets.

BACKGROUND

The growing global popularity of smartphones and tablets has resulted in new ways to gather information, both manually and automatically by means of an array of embedded sensors. Professional wearable biosensors easily connect to smart phones and can track significant physiological parameters. Existing wearable fitness sensors such as the ("Fitbit,"), ("Zephyr,"), ("NIKE+ FUELBAND,") and ("Withings,") have already been on the market for a while, letting users automatically collect data on their walking steps, activity levels, food intake, blood pressure and heart rate. The result is a more efficient and convenient way of collecting information about person's health and well-being. However, there is currently no standardized format for these data and it is difficult for users to reclaim, manage or remix that data in their preferred ways. From the provider's point of view, such massive growth of these big health sensor data creates both data manageability and collaboration challenges.

People have always been searching for the best knowledge to empower themselves for a healthier life. We share our lives and thoughts with the social community or the public through social media and enjoy connection with and access to a wide variety of sources for personal fulfilment. Despite often

insurmountable issues of privacy and data protection facing the health and wellbeing industry, the benefits of embracing social media have begun to emerge. Today more and more patients are participating actively in all aspects of personal health information. At Sarasota Memorial Hospital (Fulmer) in Florida patients "tweet" their doctors when they have questions about their care. . At Chicago's Rush University Medical Centre physicians keep connected with patients through Facebook and Twitter so that they are notified of their recovery Rush University Medical Center (Liu, Han, Zhong, Han, & He, 2009). During a real-time brain surgery in March 2009, doctors at Detroit's Henry Ford Hospital answered questions via "tweets," broadcasting to more than 6000 followers Detroit's Henry Ford Hospital (Schmuck & Haskin, 2002). Other healthcare connector sites include PatientsLikeMe.com, which incorporates patient education with online peer-to-peer communication, using information sharing about conditions, symptoms, and treatments to link patients together. Doximity (Held, Wolfe, & Crowder, 1974) and Sermo (Berkelaar, Eikland, & Notebaert, 2004) are web and mobile based social networking platforms where physicians can share insights about medicine and specific cases. People are now becoming very interested in using data-sharing social platforms for healthcare to communicate with other people with common needs and to learn more about themselves.

Sensor networks (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002a, 2002b; Pantelopoulos & Bourbakis, 2010; Xu, 2002) provide infrastructure through which we obtain data about the physical environment, social systems and health data by means of sensing devices. They are being widely used in areas like healthcare and fitness. There are many existing sensor network systems such as Aurora (Abadi et al., 2003) and COUGAR (Yao & Gehrke, 2002) focused on storing and querying the sensor data. The Discovery Net system (AlSairafi et al., 2003) provides an example in which different users can develop their own data collection workflows specifying how sensor data can be processed before storing it in a centralized data warehouse. It also enabled them to develop analysis workflows for integrating the data with data collected from other data sources. Users of the system could thus share the same data and also derive new views and analysis results through sharing. The CitySense (Murty et al., 2008) project deployed a system allowing the general public to provide feedback on pollutants using mobile devices. This demonstrates how the system supports enriching the information from the users and allowing users to comment on the operation and trustworthiness of the sensors.
The aforementioned systems support a limited form of collaboration while operating on a fixed set of sensors. However, the drive toward the pervasive use of mobile and the rising adoption of sensing devices enabling everyone to collect data at any time or place is leading to a torrent of sensor data. Traditional sensor network systems face many challenges in managing this volume of data, and the emergence of cloud computing is seen as a remedy.

Cloud computing has been widely discussed in the past few years as it shows great potential to shape the development, maintenance and distribution of both computing hardware and software resources. With this computing paradigm, the actual provision of resources is only a concern at run time for specific application requirements, and so is the case for software resources as they can also be used in an on demand and pay-per-use fashion. Cloud storage takes sharing hardware one step further: unlike local storage, cloud storage relieves end users of the task of upgrading their storage devices constantly. Cloud storage services enable inexpensive, secure, fast, reliable and highly scalable data storage solutions over the Internet. Many enterprises and personal users with limited budgets and IT resources are now outsourcing storage to cloud storage service providers, in an attempt to leverage the manifold benefits associated with cloud services. Leading cloud storage vendors, such as Amazon S3 (Amazon) and Google Cloud Storage (Google), provide clients with highly available, low cost and pay-as-you-go based cloud storage services with no upfront cost. A variety of companies have outsourced at least a portion of their storage infrastructure to Amazon AWS, including SmugMug (Szalay, Bunn, Gray, Foster, & Raicu,

2006), ElephantDrive ("ElephantDrive,"), Jungle Disk ("AWS Case Study: Jungle Disk,") and 37signals(Szalay et al., 2006). Amazon announced that as of June, 2012 it holds more than a trillion objects, and the service has so far been growing exponentially (Amazon, 2012). Even so, many enterprises and scientists are still unable to shift into the cloud environment due to privacy, data protection and vendor lock-in issues. An Amazon S3 storage service outage in 2008 left many businesses that rely on the service offline for several hours and resulted in the permanent loss of customer data (aDam LeVenthaL, 2008; Kim, Gurumurthi, & Sivasubramaniam, 2006) - an incident that led many to question the S3's "secret" architecture.

Implementing cloud computing technologies appropriately can aid healthcare providers in improving the quality of medical services and the efficiency of operations, sharing information, improving collaboration, and managing expenditures. Sensor-Cloud infrastructure (Yuriyama & Kushida, 2010) enables the sensor management capability of cloud computing by virtualizing a physical sensor. Commercial sensor network platforms such as Xively ("Xively,")(formerly known as Cosm) have taken off in recent years. They provide an online scalable sensor data management platform that allows developers to connect devices and applications through a web-based API. However, neither the architecture nor the implementation of this type of platform has yet been made public. There is still a need for a scalable data storage and high-performance computing infrastructure for efficiently storing, processing and sharing of health sensor data as well as collaboration to allow users to create, share, reuse and remix data analysis models. The knowledge of how to provide such big data sensor management service for system architecture, storage, querying and sharing mechanisms, data labelling and analysis frameworks as well as how to utilise all the resources, reduce storage consumption and costs, and improve the latency analytics against high velocity streaming data, remains untapped.

In order to examine the knowledge behind healthcare data management services and tackle the above challenges we propose the Wiki-Health platform, a cloud-based health data management system that provides new ways of storing, tagging, retrieving, analysing and searching health sensor data. In this chapter, we discuss the system design and implementation of Wiki-Health in detail. We also demonstrate a brain training prototype for ADHD (Barkley, 1997) as a use case that will help us show the proof of concept of our platform.

SYSTEM DESIGN

There is an exponential increase in the number of sensors and devices giving us data, from the personal level to environmental and global levels. These sensors and devices have their own characteristics such as sample rate, number of channels and power consumption value. These characteristics imply that such sensor data is multi-resolution. For example, most EEG devices have very high temporal resolution, typically at sampling rates between 250 and 2000 Hz in clinical and research settings, while GPS location readings are taken and stored every few minutes. When such data are aggregated and/or integrated for use in a new application the user is very quickly faced with the multi-resolution nature of that data. Moreover, users of the data will have their own conceptual multi-scale levels of abstraction in their models. To address all these issues, we designed Wiki-Health not only to solve the problems of managing and storing such high volume, velocity and variety of data, but to offer support tools for users to create application and analysis so they can make use of the data whilst lowering the complexity of dealing with its diversity.

Figure 1 illustrates the overall architecture of Wiki-Health. The Wiki-Health system is designed in three logical layers: application, query and analysis, and data storage.

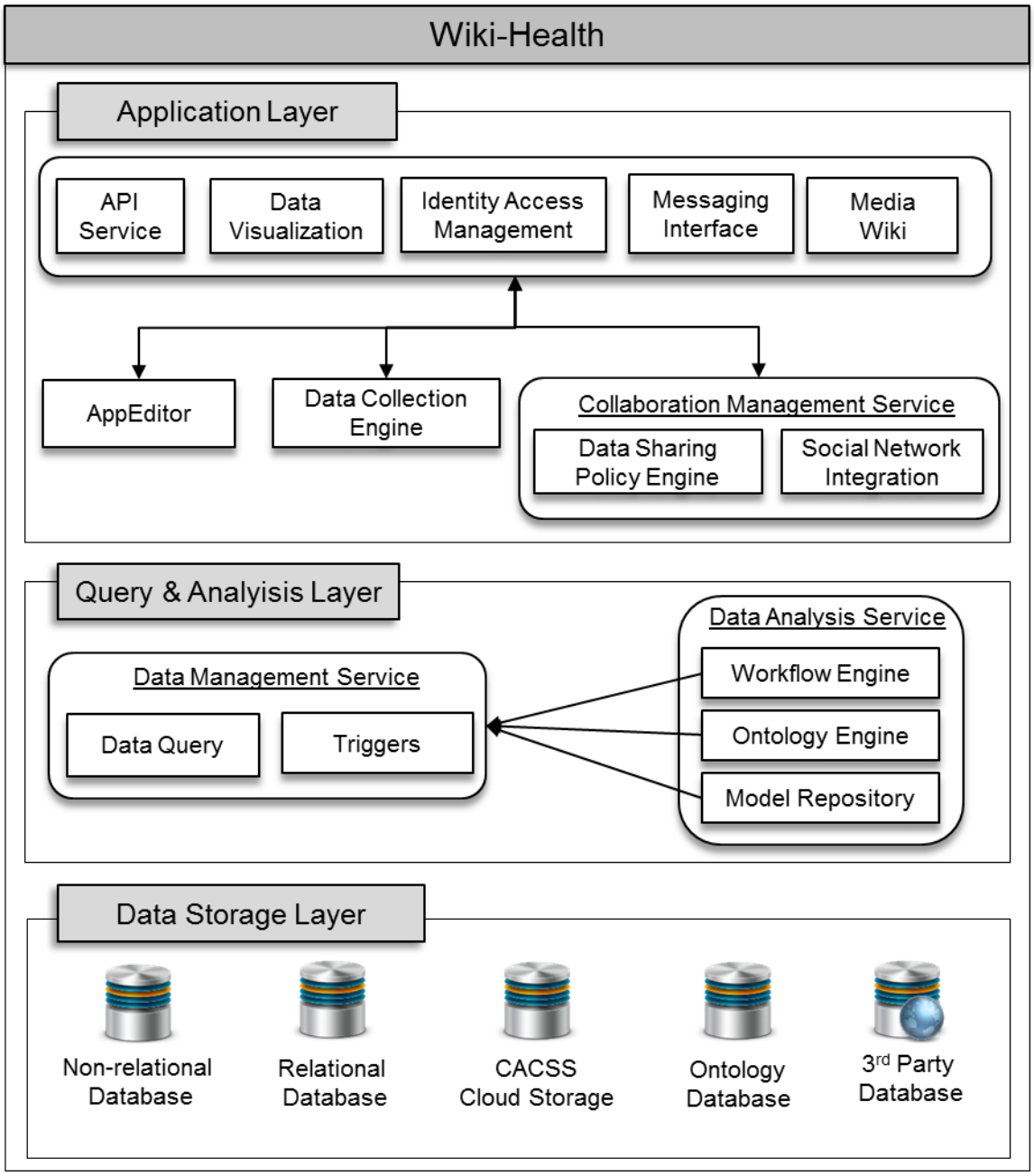

Figure 1. The architecture of Wiki-Health system

Application Layer

The application layer comprises all of the components required for managing data access, data collection, security, data sharing and other features that support online collaboration. The API service offers a web-based interface for access to all of the functionalities of Wiki-Health. The identity access management component is a separate service that provides authorization and access control of all requests. The messaging interface is used to connect Wiki-Health to external third party API and services. The data visualization module incorporates several charting libraries, such as HighChart ("HighCharts JS,") and Google Charts ("Google Charts,") to generate different graphical representations of raw sensor data as

well as processed data. Media Wiki provides a collaborative way of allowing users to add, edit and publish content relating to data analysis models, methods and workflows, based on the health sensor data stored in Wiki-Health. Sharing and collaboration are both key features that make the cloud storage useful and convenient.

The collaboration management service contains a data sharing policy engine to resolve data sharing policies defined for data streams, such that data can be shared with specific users at prerequisite times. The social network integration component creates a data-sharing social platform for users to share their data and communicate with other people who have common needs and to learn more about themselves.

The data collection engine is intended to liberate a user's personal health data collected through different providers, devices and platforms, by allowing a user to create a copy of all of his or her data and maintain it in the Wiki-Health platform through the third party API. AppEditor exposes an application development environment to offer developers a graphical user interface to construct their health sensor data applications.

Query and Analysis Layer

The query and analysis layer has two main purposes: data management and data analysis. The data management service handles all of the generic data access to different data sources in the data storage layer and triggers event actions under defined conditions. The data analysis service contains the workflow engine, ontology engine and model repository. The ontology engine component manages and queries the ontology via an API that supports the SPARQL language (Prud'Hommeaux & Seaborne, 2008). We adopt ontology in the system for two main reasons: first, ontology is used to describe the data source, and provide the semantic substrate to manipulate various health data sources, such as flexible query, data validation and etc. Second, ontology provides the basis for our data federation and fusion. For multiple data sources, we fuse them with common upper ontology. In general, ontology makes it possible for aggregating and/or integrating different resolutions and formats of sensor data. One of the goals of our system is to create an ecosystem where users can voluntarily contribute their data and models. The model repository serves this purpose; it stores all user and system defined functions, models and scripts for reuse of the data and knowledge. The workflow engine is a runtime environment to schedule and execute the workflows and processes constructed by the AppEditor.

Data storage layer

The data storage layer is a logical tier that hosts a non-relational database, a relational database, an ontology database, CACSS cloud storage (Li, Guo, & Guo, 2012, 2013) and data sources from external third party databases.

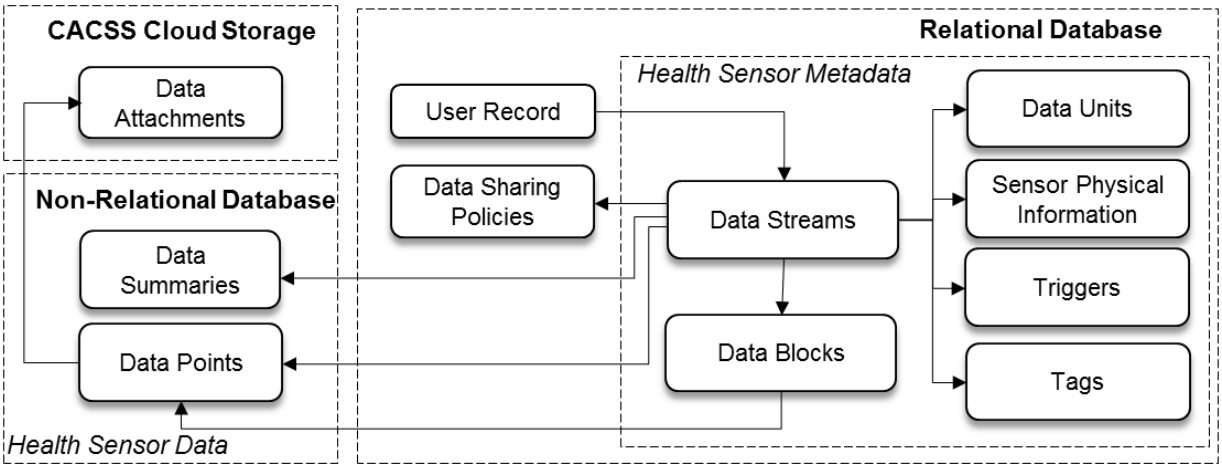

Figure 2. The hybrid data storage model

Figure 2 describes the Wiki-Health hybrid data storage model. In Wiki-Health, sensor metadata and sensor data are completely separate. The system uses a relational database to store all user information, in addition to sensor metadata such as sample rate, data format, sensor type and other Wiki-Health structured data. All sensor readings and other sensor data are stored in a non-relational (NoSQL) database and CACSS Cloud Storage System. The motivation behind this hybrid approach arises from sensor systems possessing common features such as the storage and processing of large amounts of data. Health sensor devices such as EEG and ECG employ a number of data channels and possess significantly higher temporal resolution than common environmental sensors like temperature and humidity. Such health sensor data, therefore, demands a scalable, fast and efficient system in order for it to be stored and processed. Traditional relational databases are designed for efficient transaction processing of small amounts of information in a large database. The data sets stored in these databases have no pre-defined notion of time unless timestamp attributes are explicitly added. Non-relational databases, such as HBase (HBase; Khetrapal & Ganesh, 2006), MongoDB (Chodorow, 2013; "MongoDB,") and Cassandra (Lakshman & Malik, 2010), store data in a key-value structure. They use looser consistency models than traditional relational databases, thus providing higher scalability and better performance (Cattell, 2011; Leavitt, 2010; Stonebraker, 2010; Varley, Aziz, Aziz, & Miranker, 2009). Cloud storage is another service that helps users reduce the costs, complexities and risks in managing large data growth. Wiki-Heath uses such a hybrid approach to achieve high performance in data access and operations.

In Wiki-Health's sensor data storage model, each data stream maps and holds all the information of the actual health sensor device. A single data stream can have many data units. Such data unit is useful for health devices that provide multiple channel readings at the same time. Data points contains sensor reading data, data attachment index, tag mappings, block mappings and other user defined data, they are stored as a collection of blocks addressed by an index using data stream ID together with a timestamp (as shown in Figure 3). Data blocks and tags are designed and implemented for developers and users to be able to add more dimensions to the data so that required data can be found more easily and accurately. Data summaries can be used for storing summary or intermediate analysis result to improve the response time on retrieving certain analysis and query results. A data stream can have multiple triggers. Triggers holds action and condition information, they are used to perform actions when a defined condition is reached. For example, such action can be an alert to inform a caregiver when a certain threshold or unusual reading is discovered.

$$\begin{bmatrix} \boldsymbol{streamid - timestampX} \\ \begin{bmatrix} v: unit\_id1 & 2.0 \\ v: unit\_id2 & 6.2 \\ v: unit\_id3 & 1.6 \end{bmatrix} \\ \begin{bmatrix} t: tag1 \\ t: tag2 \end{bmatrix} \\ \begin{bmatrix} b: blockid1 \\ b: blockid2 \end{bmatrix} \\ \begin{bmatrix} ud: userdefined1 & data \\ ud: userdefined2 & data \end{bmatrix} \\ \begin{bmatrix} \text{a}: attachment1 & fileloc1 \\ a: attachment2 & fileloc2 \end{bmatrix} \end{bmatrix}$$

Figure 3. Data points storage format

IMPLEMENTATION AND DETAILS

In this section, we will discuss more details about each component and its implementation. We also demonstrate how Wiki-Health adapts existing storage technologies to provide efficient and scalable services.

After considerable research and experimentation, we chose HBase as the foundational non-relational database storage for the sensor data. It is designed to provide fast real time read/write data access. Some research has already been done to evaluate the performance of HBase (Carstoiu, Cernian, & Olteanu, 2010; Khetrapal & Ganesh, 2006). Its column-orientation design confers exceptional flexibility in the storing of data. We use the CACSS cloud storage system (Li et al., 2012, 2013) to store additional unstructured data as attachments to the sensor data. We chose MYSQL as the relational database for storing sensor metadata and other related information. A data sharing policy engine is implemented using Drool tools ("DROOL,"). Drools is a business rule management system (BRMS) with a forward chaining inference based rules engine, tailored for the Java language. Users are able to write their own policy rules for the engine to interpret. The data collection engine currently implements over a few public fitness and social platform APIs such as ("Endomondo,"), ("Fitbit,"), ("Withings,"), ("Foursquare,") and ("Twitter,"), by taking those platforms' APIs as plug-ins in the Wiki-Health system. It periodically retrieves the data stores on those systems according to the user's configuration and store the data in Wiki-Health for future use. The user can specify what kind of data to collect and how often they are collected. The media wiki component uses the open source PHP script from ("MediaWiki,"). Wiki-Health interacts with the wiki component through MediaWiki's API to insert and update contents.

Ontology Management

The implementation of Wiki-Health ontology management is depicted in Figure 4. ("Virtuoso,") is used to store and manage ontologies. Virtuoso is a "universal server" enabling a single multithreaded server process that implements multiple protocols such as RDF data, XML and Linked data Management. The ("dotNetRdf library,") is used to communicate with the Virtuoso server. The Wiki-Health ontology engine invokes the functionalities of the dotNetRDF library and communicates with the Virtuoso server. A successful ontology submission will output a unique ontology identity. This identity will be required in order for the user to link the data stream with the ontology schema. Submitted ontology schemas are stored in the Virtuoso ontology repository. Data querying involves the user submitting a query either in SQL format or SPARQL and the system returning a result set. The user has the option to obtain the query output as either XML or RDF.

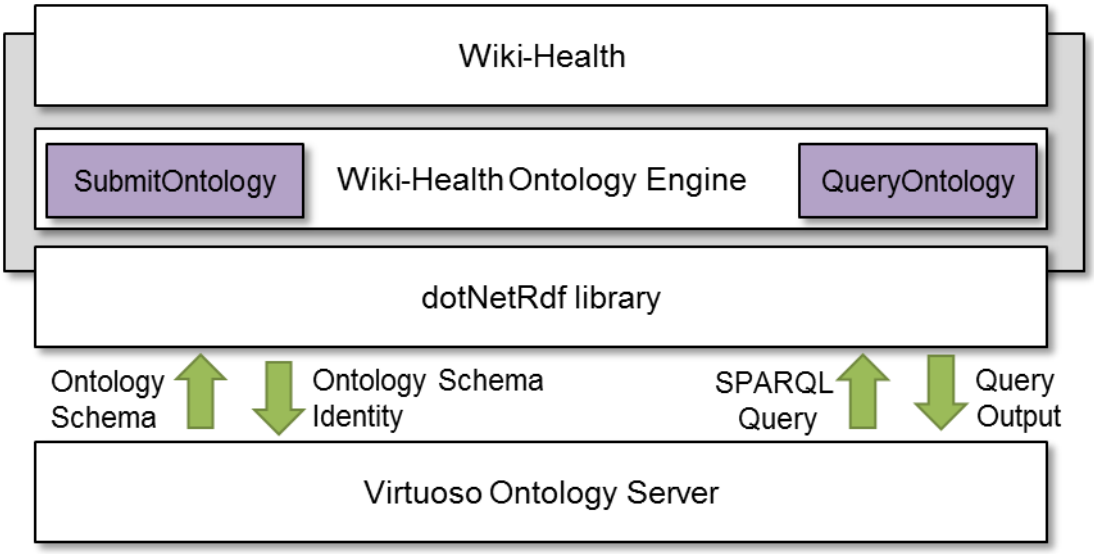

Figure 4. The implementation for ontology support

Model Repository

The model repository provides an environment to store analysis model and do model training. The repository is currently built on a distributed R platform on top of IC-Cloud (Y.-K. Guo & Guo, 2011). R is an open source software programming language and a software environment for statistical computing and graphics. R provides a wide variety of statistical models and graphical techniques, and is highly extensible. It has been used among statisticians and data miners for developing statistical software and data analysis (Gentleman, 2008; "R Project," ; Warnes et al., 2009). The model repository consists with the following components:

- Model source storage: The submitted models are stored as R source file in a file storage system.
- Web portal for R: A web front-end portal (as an example shown in Figure 6) for R is provided based on R-Node, to connect to the R runtime environment, and interact with it through the R console. Users can upload training data, view their scripts for model, and check the visualization of the runtime status and result.

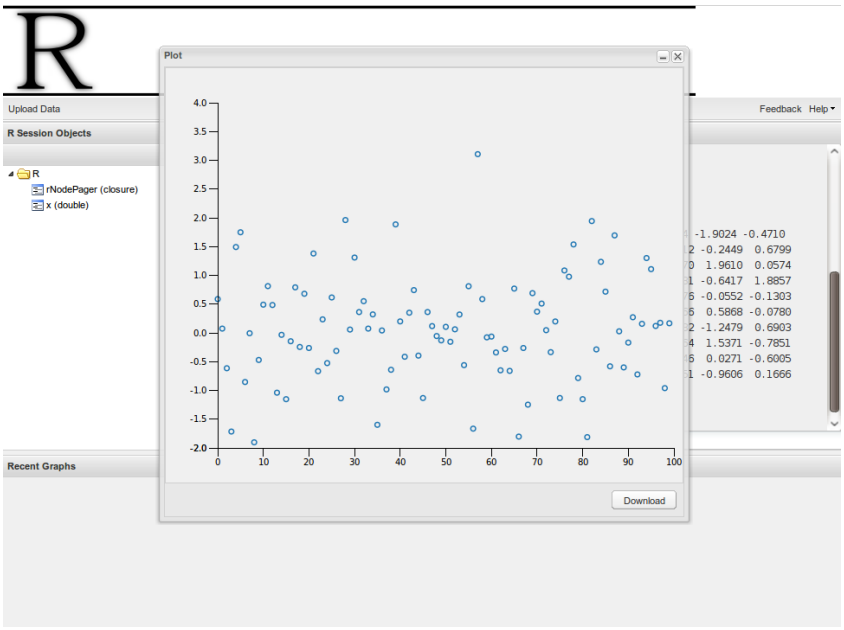

Figure 5. Web portal for R (data visualization)

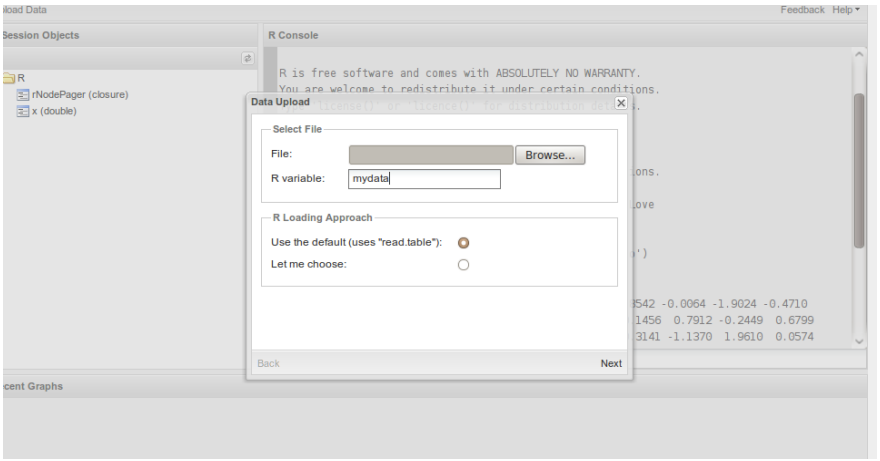

Figure 6. Web portal for R (data upload)

- API package for R (Wiki-R): Defines the interface between the R application and the environment, so that functions can be called back for actions like model training, parameter reflection, etc. When users submit a training algorithm for a model to the system via these cases, the training algorithm submitted will be verified with following conditions:
    - It's a valid R program
    - It implements the model interface (as shown in Figure 7) for call-back functions including:
        - Prediction
        - DataInput
        - DataTrain
        - ParameterList
        - ParameterRange
- Job System: A job system is provided for scheduling the modelling works in the R runtime environment and supporting the operations. The job system is built with Portable Batch System

(PBS). It connects to a set of data sources or a query. Once all the validations are passed, a model is successfully submitted, and a job is created for model training. A job ID is returned to the user.

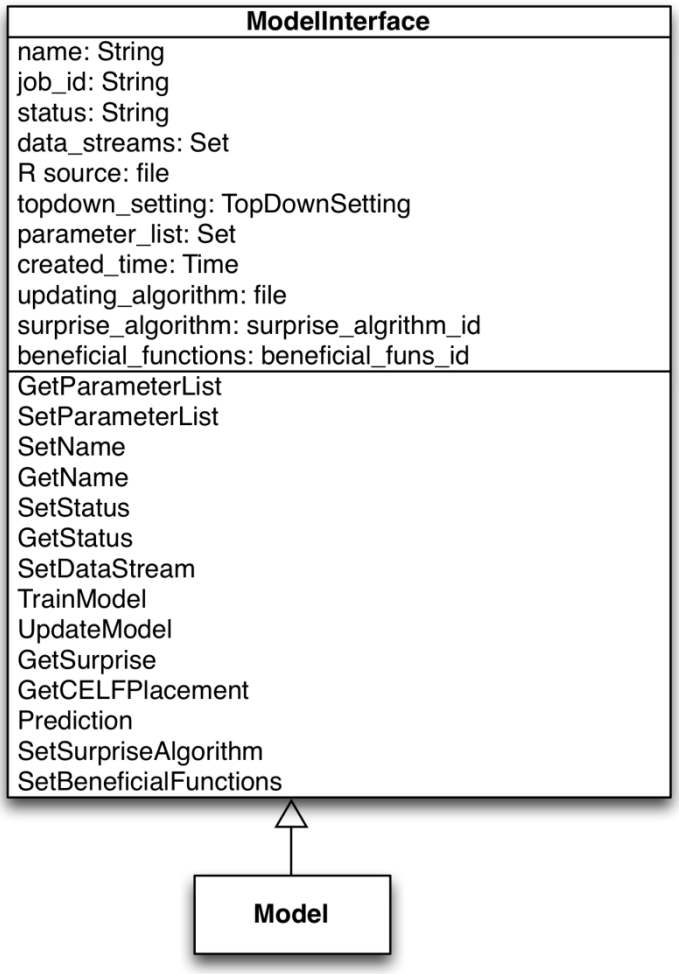

Figure 7. Model Interface for the Job System

AppEditor and Workflows

AppEditor is an online development environment, which provides a graphical user interface to integrate data from different sensors, apply analytic models, construct the workflow, and publish the final service as a sensor application for future data analysis and queries. The workflow engine component is implemented using JBOSS jBPM (Cumberlidge, 2007; Koenig, 2004). It executes the workflows constructed by the AppEditor. As shown in Figure 8, AppEditor comprises the modules described below.

Data/Models Explorer: The developers can visually discover and browse the sensor data as well as analytic models in our platform before constructing their application. It also retrieves the lists of models from the model repository.

Workflow Editor: This module provides an efficient tool to assist developers to build data applications. It is a workflow-like environment where developers can couple available elements shown in Table 1. The elements and links between them, including data and models which are listed in the tree structure by categories, can be added into a design diagram by drag-and-drop. The settings of each element, like the parameters for a model and data connection string of one data source, are edited directly in the property

window. After the developers create an application in the editor, a corresponding execution script in xml format will be generated. The script includes the detail information of each element and the relations between them. It can be delivered to the workflow engine via APIs and stored to the model repository.

Task Scheduler: Once the developers have done the design of their sensor application products, the developers can control the execution process of the application through task scheduler. The developers can run, stop, pause or check the status of their applications running via the APIs provided by the workflow engine.

Result Viewer: The Results Viewer allows users to view the results of executing their workflow using various visualization tools.

Service Publisher: Developers can finally publish the application as a web service to allow other users to reuse the analysis methods, workflows and results (as shown in Figure 9). Some social functions like rating and comments are also provided. Such list of application would be further developed towards sensor application marketplaces.

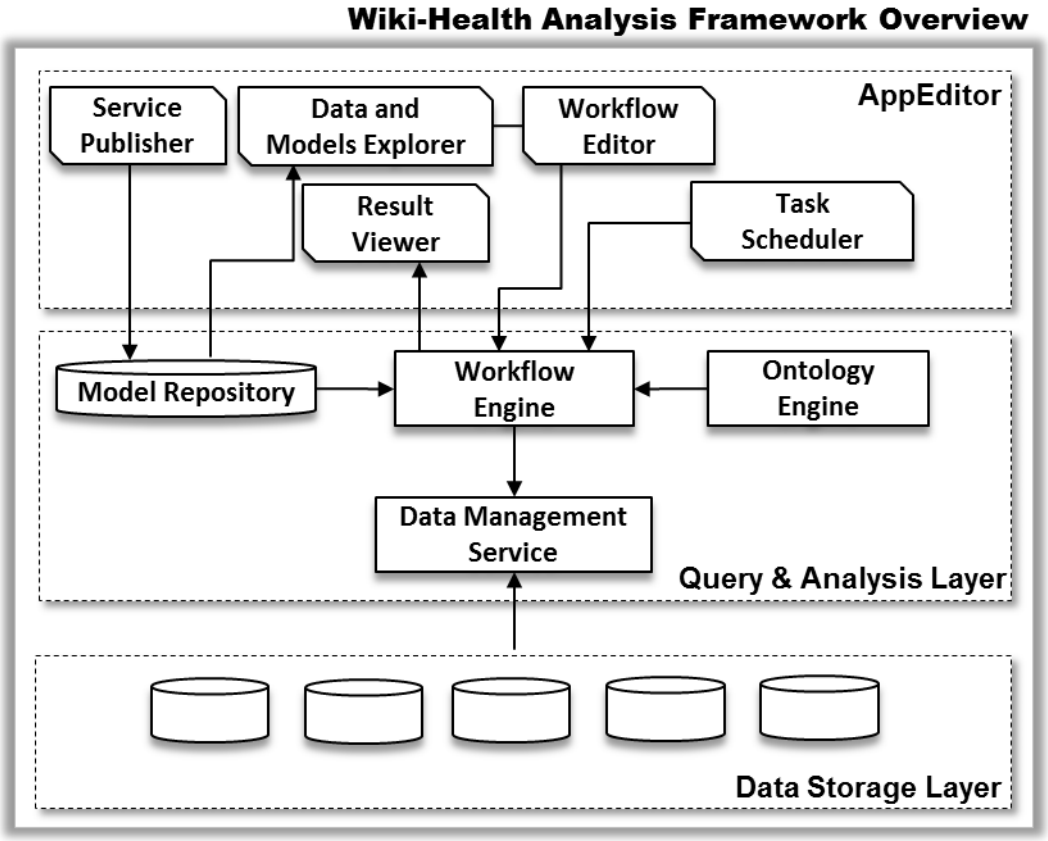

Figure 8. Wiki-Health Analysis Framework Overview

Element (shape)	Description
Data Source (circle)	Select single or multiple sensor data sources available in the platform
Filter (diamond)	Query the data source (e.g. query with location, time, values, etc.), undertake data fusion operations (e.g.

	`aggregateValue`, `union`, etc.)
Model (rectangle)	Apply various prediction, mining and analysis models on data. The system has a model repository and runtime environment where users can contribute their own models.
Connector (triangle)	Control the flow of the application, such as loops and conditions.
Visualizer (square)	Visualize the output of the application. The visualization libraries are also user-contributed.

Table 1. Available elements in the AppEditor

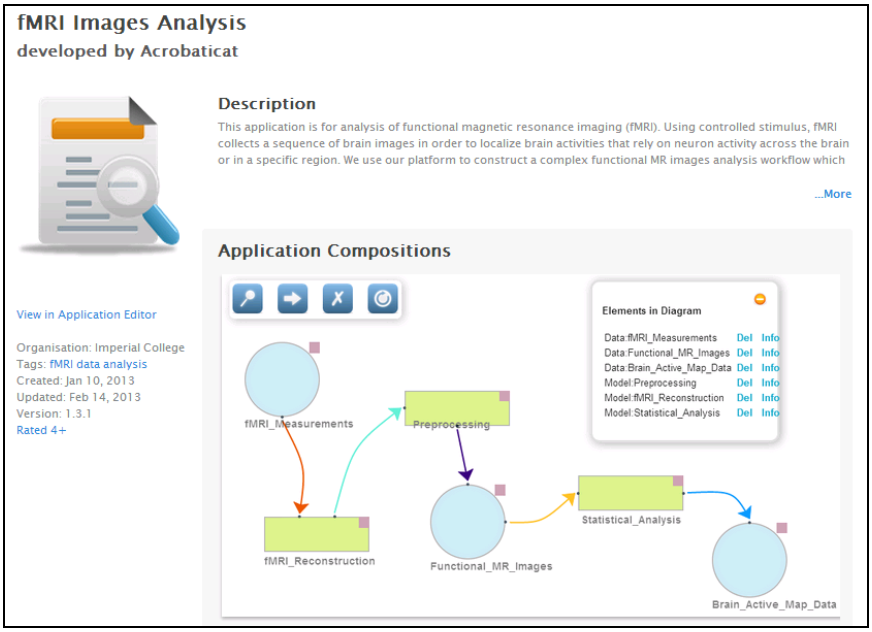

Figure 9. An example of fMRI image analysis application composed by AppEditor

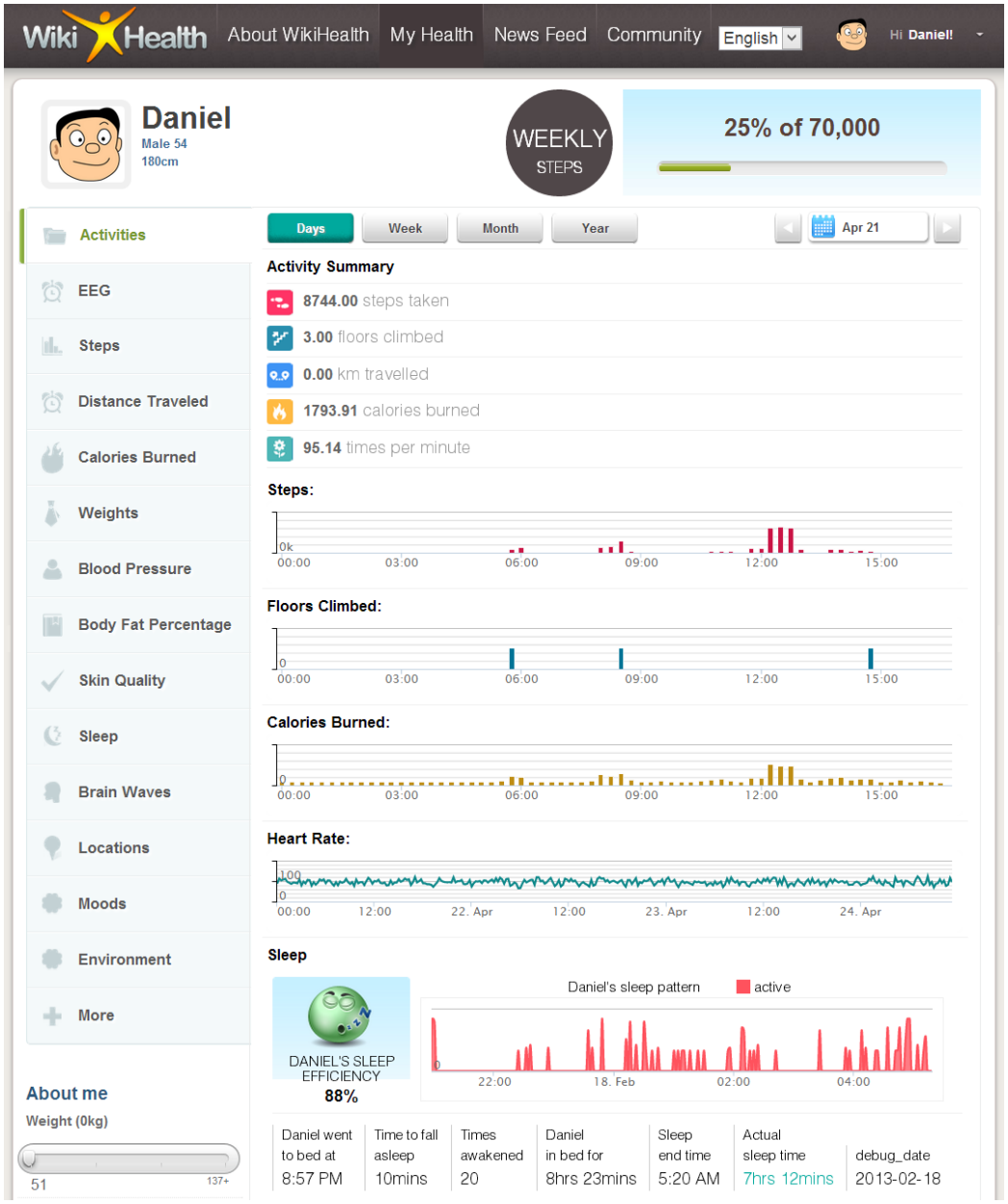

Figure 10. GUI of Wiki-Health web-based health data browser

Figure 10 shows the web-based GUI of the Wiki-Health health data browser. All the data displayed are retrieved through the web interface API of Wiki-Health. All the system components above described are deployed across different virtual machines on top of IC-Cloud (L. Guo, Guo, & Tian, 2010; Y.-K. Guo & Guo, 2011). IC-Cloud is a generic IaaS cloud computing infrastructure. It allows us to design and quickly compose a cloud computing environment in a flexible manner.

CASE STUDY

The recording of brain activity via EEG provides a way of measuring the state of the brain and of specific brain functions, offering the potential for safe neurofeedback treatments for ADHD (Arns, de Ridder, Strehl, Breteler, & Coenen, 2009; Barkley, 1997; Fuchs, Birbaumer, Lutzenberger, Gruzelier, & Kaiser, 2003; Lubar, Swartwood, Swartwood, & O'Donnell, 1995). Real-time measurement of concentration

levels has been used in the development of video games designed to improve focus levels of patients through neurofeedback training over a period of time. In this case study, a cloud-based neurofeedback prototype system is built on top of Wiki-Health platform and by aiding the integration of models and data together using AppEditor (shown in Figure 11). One of the analysis models is used by this prototype is a k-nearest neighbour (kNN) approach to classify level of alertness (concentration) using a pre-created profile for the individual as training data.

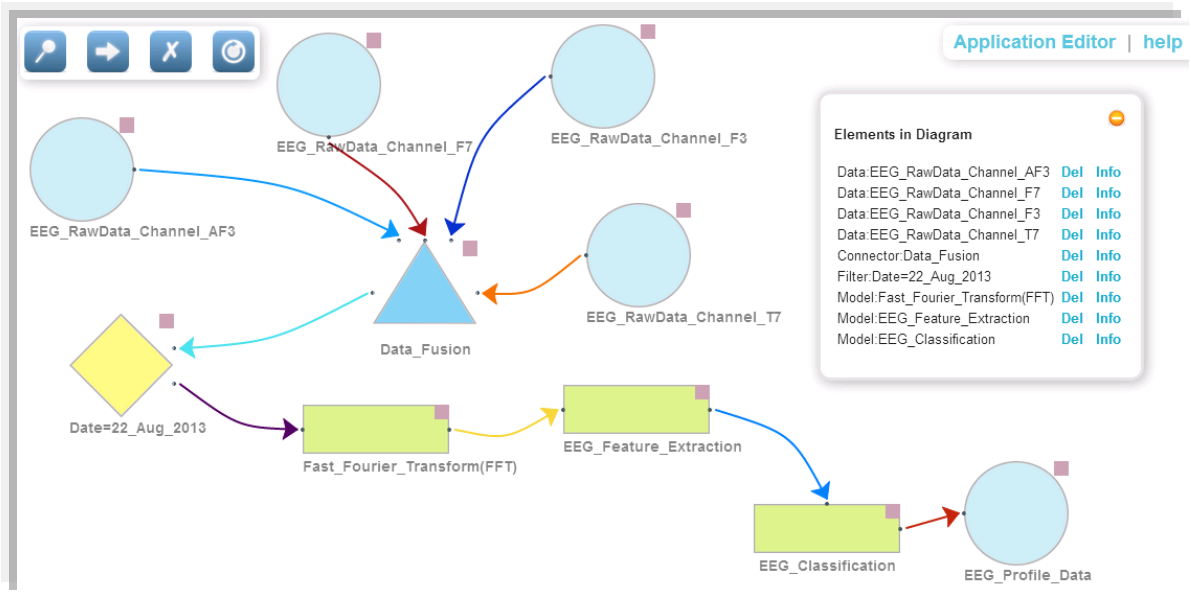

Figure 11. Example of EEG analysis application composed by AppEditor

The prototype system also includes a table tennis brain training game for evaluating the result. The real-time alertness level of the subject is calculated in order to adjust the difficulty of the game. If the subject is alert, the game difficulty is reduced; if the subject is relaxed, the game difficultly is increased. In our experiments, six subjects are tested with the table tennis braining training game. Figure 12 shows the result of the alertness level and the difficulty level during the game play. We found the game successfully guides the subject into a stable state. At the beginning, the difficulty is very low, so subject feels relaxed and the difficulty keeps rising; at 400 seconds, the difficulty is high enough so the subject's state becomes alertness, but the brain state still unstable, the alertness level is fluctuating in a wide range. After about 1000 seconds, the subject becomes fully stable; alertness and difficulty level are stable in a small range. The purpose of the brain training games is to improve the player's ability to concentrate and put the brain into a stable and relaxed state at the same time. Typically, the game difficulty should increase if patient is not alert and decrease if patient is alert, through this feedback, the difficulty will be adjusted to a degree which allows the user to keep focus on playing game, but not tense. Therefore, this kind of brain training games can potentially be used for ADHD treatment.

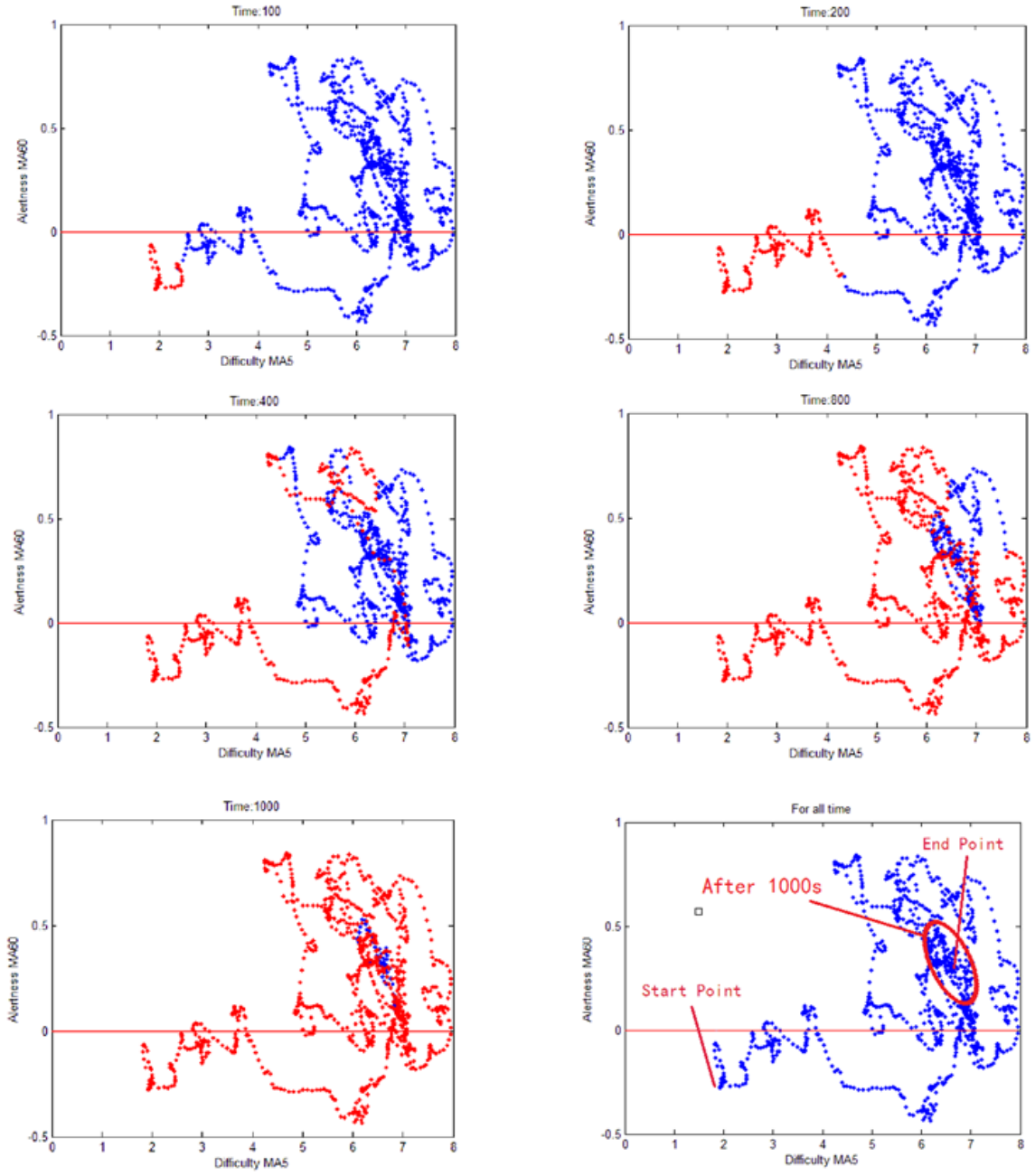

Figure 12. Experiment results of the averaged alertness values and the difficulty values at different times (100s, 200s, 400s, 800s, 1000s, 1000+s); alertness: positive means alert, negative means relaxed Difficulty: the higher the harder.

Managing the large amount of growing biomedical data is always a challenge for many researchers. Especially if there is a requirement for real time access to all of the stored data. In this case study, all of the collected EEG signal data is stored and processed in Wiki-Health. We first define all of the data analysis models: the data fusion model of the multi-channel EEG data; the FFT model, which transforms EEG data into the frequency domain; the EEG feature extraction model, which utilizes high correlation parameters and EEG channels; and the data classification model, which estimates the alertness level by comparing the present EEG features with the previously recorded EEG profile. We then use AppEditor to

design and compose the application workflow, in order to connect all of the data sources and analysis models. The final published application can be reused and remixed by other users, thereby providing a new mode of research collaboration. Using a cloud-based system such as Wiki-Health is ideal for ease of storage of a growing amount of online signal data, as well as providing seemingly unlimited space in which to store extensive legacy signal data for the analysis of long term training performances.

CONCLUSION

This chapter has presented the design and implementation of Wiki-Health, a cloud-based big data platform for health sensor data management. We introduced a new collaborative approach for health sensor data management that not only allows users to collect, label, tag, annotate, update and share sensor data, but also allow users to create, reuse and remix data analysis results and models. To validate our proposed platform we have also built a brain training prototype which can potentially be used for ADHD treatment. The approach we propose offers enterprises and the research community considerable advantage in combining existing technologies such as cloud computing, cloud storage and other tools in developing and extend our work towards a successful collaborative health sensor data management system.

The purpose of labelling, tagging or annotating sensor data is often to identify the correct event, stimulus or cause associated with a corresponding sequence of sensor data. Such labels are useful for data analysis algorithms. However, labelling sensor data can be very time consuming, often requiring input from users. A fundamental data-mining problem is to examine data for "similar" items. Researchers are always interested in finding "similar" data with factors such as correlation and causality across different sets of individuals, groups or entire populations. For our future work, we plan to design and implement a smart data labelling and linking framework to allow the system to automatically label corresponding sequences of the data and link "similar" data across all the data stored in the system based on the data analysis results obtained.

REFERENCES

Abadi, D. J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., . . . Zdonik, S. (2003). Aurora: a new model and architecture for data stream management. *The VLDB Journal—The International Journal on Very Large Data Bases, 12*(2), 120-139.

aDam LeVenthaL, B. (2008). Flash storage memory. *Communications of the ACM, 51*(7).

Admins, R. (2011). reddit's May 2010. *State of the Servers" report," Reddit. com, 11*.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002a). A survey on sensor networks. *Communications magazine, IEEE, 40*(8), 102-114.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002b). Wireless sensor networks: a survey. *Computer networks, 38*(4), 393-422.

AlSairafi, S., Emmanouil, F.-S., Ghanem, M., Giannadakis, N., Guo, Y., Kalaitzopoulos, D., . . . Wendel, P. (2003). The design of discovery net: Towards open grid services for knowledge discovery. *International Journal of High Performance Computing Applications, 17*(3), 297-315.

Amazon. Amazon Simple Storage Service (S3). from http://aws.amazon.com/s3/

Amazon. (2012). Amazon S3 - The First Trillion Objects.

Arns, M., de Ridder, S., Strehl, U., Breteler, M., & Coenen, A. (2009). Efficacy of neurofeedback treatment in ADHD: the effects on inattention, impulsivity and hyperactivity: a meta-analysis. *Clinical EEG and neuroscience, 40*(3), 180-189.

AWS Case Study: Jungle Disk.

Barkley, R. A. (1997). Behavioral inhibition, sustained attention, and executive functions: constructing a unifying theory of ADHD. *Psychological bulletin, 121*(1), 65.

Berkelaar, M., Eikland, K., & Notebaert, P. (2004). lpsolve: Open source (mixed-integer) linear programming system. *Eindhoven U. of Technology*.

Carstoiu, D., Cernian, A., & Olteanu, A. (2010, 11-13 May 2010). *Hadoop Hbase-0.20.2 performance evaluation.* Paper presented at the New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on.

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record, 39*(4), 12-27.

Chodorow, K. (2013). *MongoDB: the definitive guide*: O'Reilly.

Cumberlidge, M. (2007). *Business Process Management with JBoss JBPM: A Practical Guide for Business Analysts; Develop Business Process Models for Implementation in a Business Process Management Sytem*: Packt Publishing.

dotNetRdf library. from http://www.dotnetrdf.org

DROOL. from http://www.jboss.org/drools/

ElephantDrive. from http://aws.amazon.com/solutions/case-studies/elephantdrive/

Endomondo. from http://www.endomondo.com/

Fitbit. from http://www.fitbit.com/

Foursquare. from https://foursquare.com

Fuchs, T., Birbaumer, N., Lutzenberger, W., Gruzelier, J. H., & Kaiser, J. (2003). Neurofeedback treatment for attention-deficit/hyperactivity disorder in children: a comparison with methylphenidate. *Applied psychophysiology and biofeedback, 28*(1), 1-12.

Fulmer, J. Siege HTTP regression testing and benchmarking utility. *URL http://www. joedog. org/JoeDog/Siege*.

Gentleman, R. (2008). *R programming for bioinformatics*: CRC Press.

Google. Google Cloud Storage Service. from http://code.google.com/apis/storage/

Google Charts. from https://developers.google.com/chart/

Guo, L., Guo, Y., & Tian, X. (2010). *IC cloud: a design space for composable cloud computing.* Paper presented at the Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on.

Guo, Y.-K., & Guo, L. (2011). IC cloud: Enabling compositional cloud. *International Journal of Automation and Computing, 8*(3), 269-279. doi: 10.1007/s11633-011-0582-4

HBase, A. from http://hbase.apache.org/

Held, M., Wolfe, P., & Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical programming, 6*(1), 62-88.

HighCharts JS. from http://www.highcharts.com/

Khetrapal, A., & Ganesh, V. (2006). HBase and Hypertable for large scale distributed storage systems. *Dept. of Computer Science, Purdue University*.

Kim, Y., Gurumurthi, S., & Sivasubramaniam, A. (2006). *Understanding the performance-temperature interactions in disk I/O of server workloads.* Paper presented at the High-Performance Computer Architecture, 2006. The Twelfth International Symposium on.

Koenig, J. (2004). JBoss jBPM white paper. *JBoss Labs*.

Lakshman, A., & Malik, P. (2010). Cassandra: a decentralized structured storage system. *SIGOPS Oper. Syst. Rev., 44*(2), 35-40. doi: 10.1145/1773912.1773922

Leavitt, N. (2010). Will NoSQL databases live up to their promise? *Computer, 43*(2), 12-14.

Li, Y., Guo, L., & Guo, Y. (2012). *CACSS: Towards a Generic Cloud Storage Service*. Paper presented at the CLOSER. http://dblp.uni-trier.de/db/conf/closer/closer2012.html#LiGG12

Li, Y., Guo, L., & Guo, Y. (2013). An Efficient and Performance-Aware Big Data Storage System *Cloud Computing and Services Science* (pp. 102-116): Springer.

Liu, X., Han, J., Zhong, Y., Han, C., & He, X. (2009). *Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS.* Paper presented at the Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on.

Lubar, J. F., Swartwood, M. O., Swartwood, J. N., & O'Donnell, P. H. (1995). Evaluation of the effectiveness of EEG neurofeedback training for ADHD in a clinical setting as measured by changes in TOVA scores, behavioral ratings, and WISC-R performance. *Biofeedback and Self-regulation, 20*(1), 83-99.

MediaWiki. from http://www.mediawiki.org

MongoDB. from http://www.mongodb.org

Murty, R. N., Mainland, G., Rose, I., Chowdhury, A. R., Gosain, A., Bers, J., & Welsh, M. (2008). *Citysense: An urban-scale wireless sensor network and testbed.* Paper presented at the Technologies for Homeland Security, 2008 IEEE Conference on.

NIKE+ FUELBAND. from http://www.nike.com/

Pantelopoulos, A., & Bourbakis, N. G. (2010). A survey on wearable sensor-based systems for health monitoring and prognosis. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40*(1), 1-12.

Prud'Hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. *W3C recommendation, 15*.

R Project. from http://www.r-project.org/

Schmuck, F. B., & Haskin, R. L. (2002). *GPFS: A Shared-Disk File System for Large Computing Clusters.* Paper presented at the FAST.

Stonebraker, M. (2010). SQL databases v. NoSQL databases. *Communications of the ACM, 53*(4), 10-11.

Szalay, A., Bunn, A., Gray, J., Foster, I., & Raicu, I. (2006). *The importance of data locality in distributed computing applications.* Paper presented at the NSF Workflow Workshop.

Twitter. from https://twitter.com

Varley, I. T., Aziz, A., Aziz, C.-s. A., & Miranker, D. (2009). No relation: The mixed blessings of non-relational databases.

Virtuoso. from http://virtuoso.openlinksw.com

Warnes, G. R., Bolker, B., Bonebakker, L., Gentleman, R., Huber, W., Liaw, A., . . . Moeller, S. (2009). gplots: Various R programming tools for plotting data. *R package version, 2*(4).

Withings. from www.withings.com

Xively. from https://xively.com/

Xu, N. (2002). A survey of sensor network applications. *IEEE Communications Magazine, 40*(8), 102-114.

Yao, Y., & Gehrke, J. (2002). The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record, 31*(3), 9-18.

Yuriyama, M., & Kushida, T. (2010). *Sensor-cloud infrastructure-physical sensor management with virtualized sensors on cloud computing.* Paper presented at the Network-Based Information Systems (NBiS), 2010 13th International Conference on.

Zephyr. from http://www.zephyranywhere.com/

Lee, C. H., Birch, D., Wu, C., Silva, D., Tsinalis, O., Li, Y., ... & Guo, Y. (2013, October). *Building a generic platform for big sensor data application*. In Big Data, 2013 IEEE International Conference on (pp. 94-102). IEEE.

KEY TERMS & DEFINITIONS

Wiki-Health System: A technology takes advantage of cloud computing and Internet of Things for social and personal well-being data management.

Data Collection Engine: This engine collects health data from different service providers, devices and platforms through 3rd party APIs on the behalf of the user.

Collaboration management service: This service integrates social network features and manages data sharing policies for health sensor data.

AppEditor: A graphical user interface to allow users and developers to construct workflows and health sensor data applications.

Workflow engine: This engine executes workflows and processes constructed by the AppEditor.

Model Repository: A storage space to store all user and system defined functions, models and scripts for reuse of the data and knowledge.

Ontology Engine: This engine provides the basis for the health sensor data and schema validation, federation and fusion.

Triggers: An approach to store action and condition information. It allow certain actions to be performed when defined conditions are reached.

BIOGRAPHICAL SKETCH

Mr.Yang Li is a researcher and PhD. student at the department of computing, Imperial College London, UK. He received M.Sc. in computing from Imperial College London and B.Sc in mathematics from University College London, UK. His research interests are cloud computing, cloud storage, cloud computing applications for healthcare and big data platforms.

Dr. Chao Wu is a research associate in Department of Computing, Imperial College London. He got his Doctor degree from Zhejiang University. His research interest is in the areas of social networking, cloud computing, folksonomy system, data visualization, etc.

Dr. Li Guo is Lecturer at the School of Computing, Engineering and Computer Science at the University of Central Lancashire. He has PhD from the University of Edinburgh (2007) and has worked as research associate at Imperial College London. His research interests are in cloud computing, distributed sensor informatics, big data analysis and intelligent multi-agent systems, and has more than 40 peer-reviewed publications in these areas. He has contributed to many EPSRC and EU research projects and was the chief architect of the Imperial College Cloud (IC Cloud) system currently in use in a wide variety of Digital Economy projects, and was also chief architect for EU FP6 project-GridEcon providing computational facilities for data analysis services from a variety of sources and devices.

Prof. Yike Guo has been working in the area of data intensive analytical computing since 1995. During last 15 years, he has been leading the data mining group to carry out many research projects, including UK e-science projects such as: Discovery Net on Grid based data analysis for scientific discovery; MESSAGE on wireless mobile sensor network for environment monitoring; BAIR on system biology for diabetes study; iHealth on modern informatics infrastructure for healthcare decision making; UBIOPRED on large informatics platform for translational medicine

research; Digital City Exchange on sensor information-based urban dynamics modelling. He was the Principal Investigator of the Discovery Science Platform grant from UK EPSRC where he is leading the team to build the IC Cloud system for large scale collaborative scientific research. He is now the Principal Investigator of the eTRIKS project, a 23M Euro project in building a cloud-based translational informatics platform for global medical research