Health Technol. (2013) 3:153–175
DOI 10.1007/s12553-013-0051-x

ORIGINAL PAPER

Cloud-based Smart CDSS for chronic diseases

M. Hussain · A. M. Khattak · W. A. Khan · I. Fatima · M. B. Amin · Z. Pervez · R. Batool · M. A. Saleem · M. Afzal · M. Faheem · M. H. Saddiqi · S. Y. Lee · K. Latif

Received: 3 October 2012 / Accepted: 19 February 2013 / Published online: 27 March 2013

Abstract The rise in living standards that has occurred with the advancement of new technologies has increased the demand for sophisticated standards-based health-care applications that provide services anytime, anywhere, and with low cost. To achieve this objective, we have designed and developed the Smart Clinical Decision Support System (Smart CDSS) that takes input from diverse modalities, such as sensors, user profile information, social media, clinical knowledge bases, and medical experts to generate standards-based personalized recommendations. Smartphone-based, accelerometer-based, environment-based activity-recognition algorithms are developed with this system that recognizes users' daily life activities. For example, social media data are captured for a diabetic patient from his/her social interactions on Twitter, e-mail, and Trajectory and then combined with clinical observations from real encounters in health-care facilities. The input is converted into standard interface following HL7 vMR standards and submitted to the Smart CDSS for it to generate recommendations. We tested the system for 100 patients from Saint Mary's Hospital: 20 with type-1 diabetes, 40 with type-2 diabetes mellitus, and 40 with suspicions for diabetes but no diagnosis during clinical observations. The system knowledge base was initialized with standard guidelines from online resources for diabetes, represented in HL7 Arden syntax. The system generates recommendations based on physicians' guidelines provided at the hospital during patient follow-ups. With support from the Azure cloud infrastructure, the system executed the set of guidelines represented in Arden syntax in a reasonable amount of time. Scheduling and executing the 3–5 guidelines called medical logic modules (MLMs) required less than a second.

Keywords Activity recognition · Social media · Cloud computing · Healthcare · CDSS · HL7

M. Hussain · A. M. Khattak · W. A. Khan · I. Fatima · M. B. Amin · Z. Pervez · R. Batool · M. A. Saleem · M. Afzal · M. Faheem · M. H. Saddiqi · S. Y. Lee (✉)
Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu,
Yongin-si, Gyeonggi-do 446-701, Korea
e-mail: sylee@oslab.khu.ac.kr

M. Hussain
e-mail: maqbool.hussain@oslab.khu.ac.kr

K. Latif
School of Electrical Engineering and Computer Science, National University of Sciences and Technology, H-12, Islamabad, Pakistan
e-mail: khalid.latif@seecs.edu.pk

1 Introduction

The rise in living standards that has occurred with the advancement of new technology has increased the demand for sophisticated health-care applications and services. This has led to the emergence of information and communication technology (ICT)–based clinical decision support systems (CDSS) and online health-care (e-health) applications, systems, and services. To decrease their associated costs, the US federal government is investing $27 billion in health information technology (HIT) under the American Recovery and Reinvestment Act of 2009 [1]. This huge investment was targeted to adopt electronic health records (EHR) at each level of care with meaningful use of HIT—the "meaningful use" criteria of EHR. It has been revealed that health-care costs could be reduced if HIT reduces expensive adverse events [2]. In recent studies, researchers have determined that CDSS support in EHR produces the best return on investment for providers, as most of the meaningful-use criteria target these functionalities [3]. Moreover, various experiments have shown that EHR can improve patient care, reduce errors, and reduce time, if properly equipped with clinical decision support services [4].

With the evolution of architectures and rapidly changing requirements, CDSS has also evolved from a stand-alone to a service-based architecture [5]. Although CDSS reduces costs of health care, improves patient care, and reduces errors and omissions, it remains a nightmare to widely implement the clinical decision support capabilities. According to a study of commercially available CDSS systems, inconsistent capabilities have been observed, even though each claims to fulfill the CCHIT criteria. Additionally, the clinical information system has significant limitations in allowing pluggable CDS services within complex workflows [4].

The most prominent challenges for adoption of CDSS capabilities are improvement of existing CDS interventions, creation of new CDS interventions, and dissemination of CDS knowledge and interventions [6]. At a more granular level, these challenges can be tackled by building a shareable knowledge base. Additionally, pluggable interfaces are required for seamless integration and unified interfaces through the Internet are needed for merging the various knowledge bases of different domains. Further needed refinements include prioritizing and avoiding duplications of alerts and recommendations and improving human computer interfaces (HCI) [6].

Managing some of these challenges, recent trends have progressed toward service-oriented architecture (SOA)–based CDSS with provision of standard interfaces and standard knowledge bases. HL7 is playing an active role in producing ANSI-accredited standards for CDSS [7]. The widely used standards include Arden syntax for sharable knowledge, GELLO, and vMR (virtual medical record) for standard interfaces to overcome heterogeneity of external clinical systems [8–10]. With joint collaboration of the OMG (object management group), HL7 is also standardizing interfaces for SOA-based service for decision support [8].

To overcome the shortcomings of existing approaches for CDSS, we propose Smart CDSS architecture, which provides a hybrid approach combining some existing systems. To enable Smart CDSS to have a sharable knowledge base, HL7 Arden syntax is used, and to cope with the heterogeneity of existing clinical systems, HL7 vMR is incorporated in conjunction with HL7 GELLO. Therefore, reducing cost and making the system accessible via the Internet while maintaining the privacy of patient data, Smart CDSS is deployed as a Microsoft WCF [11] service on a public cloud infrastructure Microsoft Azure [12, 13], with fabricated anonymization service [14].

The Smart CDSS service is tested on a diabetes dataset of 100 patients from Saint's Mary Hospital, Korea.[1] Moreover, to handle the heterogeneity of other systems, we utilize the existing sensor-based activity recognition system [11] and social media interaction module to work with clinical decision support for diabetic patients. The sensor-based activity recognition system is deployed in the home health environment to monitor the activities of the diabetic patient, and the social media interaction module collects the social activities of the patient through Twitter, e-mails, and Trajectory mining. For a single-patient context, all the related information is converted to the standard format of vMR through its respective adapters, such as the sensor adapter and the social media adaptor. The clinical data from Saint Mary's is represented in HL7 CDA (clinical document architecture) and is transformed into HL7 vMR format through an adapter interoperability engine. The patient data in vMR format from the different sources is merged to a single HL7 vMR format that complies with the standard input interface of Smart CDSS, defined in the interface engine. Based on the common input, Smart CDSS generates recommendations that match the criteria of the diabetes rules in the knowledge base. The generated guidelines are converted into HL7 vMR standard output and returned to the fusion adapter. The fusion adapter converts the HL7 vMR output into consumer application format.

The knowledge base consists of sample diabetic rules derived from [3] and converted into proper Arden syntax format. Each diabetes rule is represented as an individual medical logic module (MLM) and converted into compiled C# class for execution. The knowledge broker selects the appropriate knowledge reasoner and launches to execute the input. The knowledge reasoner loads related MLMs based on events specified in the input and executes them one by one. The generated recommendations from all MLMs are accumulated into HL7 vMR output format and sent back to the consumer via adapters. The overall working of the Smart CDSS is explained in this paper through case studies of diabetes patients. The system has been evaluated on a clinical dataset of 100 patients from Saint Mary's Hospital: 20 with type-1 diabetes, 40 with type-2 diabetes, and 40 with suspicions for diabetes but no diagnosis of diabetes mellitus during clinical observations. The clinical dataset is used in combination with a patient social-media dataset, such as for social interactions on Twitter and Trajectory, and a dataset collected during activity recognitions. The social media datasets for this study are taken from online resources and associated with patient medical records.

The rest of the paper is organized as follow: Section 2 describes the related research work and existing tools in the areas of e-health and u-health. Section 3 discusses the proposed system with detailed descriptions of all the developed components. Section 4 presents the case studies running the proposed system for examples of diabetic patients and their information manipulation. Finally, Section 5 provides conclusions about the research findings and discusses future directions.

2 Related works

The clinical decision support system (CDSS) has a strong history, starting in 1960 with stand-alone environments.

[1] http://www.cmcseoul.or.kr/

With the advancement of architectural approaches and new requirements, CDSS has evolved from a stand-alone to a service-oriented architecture (SOA) environment. Moreover, for seamless integration of CDSS with existing health-care systems (EHRs, EMRs, PHRs, and CPOEs) to allow sharing of medical knowledge, various standards have emerged to achieve the desired goals. The most prominent knowledge representation language in the clinical domain is HL7 Arden syntax. Therefore, we will discuss the CDSS supporting Arden syntax as the main standard for the knowledge base.

Thyrexpert, Toxopert, and Heaxpert are commercially developed systems that use HL7 Arden syntax for knowledge representations. Thyrexpert assists in thyroid hormone test results and triggers reminders for quality assurance of thyroid diagnoses. Toxopert helps in interpretation of time sequences of toxoplasmosis serology test results. Heaxpert is a web-based system that helps in interpretation and plausibility-checking of hepatitis A and B and serology test results. It includes one main MLM that evokes other MLMs: three MLMs containing decision-making and one MLM containing the template texts for outputs. Its authors state it is integrated with Siemen and with Orbis of Agfa, and they are planning for integration with iPhone and iPad [9].

Moni-ICU detects and continuously monitors nosocomial (i.e., hospital-acquired) infections. Moni-ICU uses a distinct approach by invoking a number of MLMs and implementing different rules that are controlled from one central MLM. The Moni-ICU application works in the ICU connected to a microbiology lab and a patient management system. It monitors all patients on a daily basis in each of the normal intensive care units, which comprise around 100 beds in total [9].

Arden2ByteCode, a newly developed open-source compiler, runs on Java virtual machines (JVM) and translates Arden syntax directly to Java bytecode (JBC) executable. This complier is integrated with SOA-based environments called open services gateway initiative (OSGi) platforms. The compiler has the capability to support all operators of Arden syntax and compile production Java bytecode in minimal time. Due to this direct bytecode, the execution time of MLM is considerably reduced [15].

Arden/J is a Java-based MLM execution environment that provides integration with XML-based and EMR systems and produces recommendations by executing MLM compiled to Java code. Arden/J supports a runtime environment that allows integration with other systems by implementing mapper interfaces. The authors claim good performance of the compiler and have tested it with XML-based EMR [16].

All these systems have potential usages in integrated environments but have failed to communicate with third party heterogeneous health-care systems. The most prominent problem is lack of standardized interfaces that allow an external system to interact with these systems for decision support. Moreover, these systems have lacks in combining multiple knowledge bases in the same domain and controlling knowledge evolution.

3 Smart CDSS framework

The Smart CDSS framework works on top of an SC^3 [17] environment working as a cloud service. The services of the Smart CDSS consist of health-care recommendations to the user based on his/her activities and interactions. Smart CDSS provides meaningful recommendations based on refined input coming from the activity and emotion recognition (AER) and the context-aware activity manipulation engine (CAME). The AER and CAME provide patient-personalized information like the profile, daily activities, and tracking of medications [11, 13, 18].

For the clinical data manipulation and storage, existing standards like HL7 are used in the proposed Smart CDSS system. Smart CDSS works on top of an SC3 infrastructure which is described in detail in [11, 17] (Fig. 1). This section describes the functionalities of the different components of the proposed Smart CDSS in detail, with their sequence of information flow. Smart CDSS collects user activity information from different sensors deployed in the home environment, collects user social interaction information from the social media with which the user frequently interacts and discusses their daily life activities and experiences, and collects knowledge from experts that is stored in the knowledge repository. The collected information is sequentially processed by different modules (discussed below) of Smart CDSS, which results in intelligent recommendations generation for the user. The proposed architecture of Smart CDSS is shown in Fig. 2, and the details of the internal components are discussed in the subsections below.

3.1 Activity recognition

Activity recognition is an important module in SC^3 used to recognize and monitor patient daily life activities. This module provides useful information to Smart CDSS for managing patient diseases such as diabetes. Wireless and built-in smartphone sensors are used to detect and collect human daily life activities. The collected real-time data is then transmitted to the activity recognition engine on Smart CDSS, deployed over the cloud server. The recognized activities are eventually used by Smart CDSS to provide different health-care and safety-monitoring services, social recommendations, an information sharing and exchange facility, emergency connection services, and patient monitoring and care services [11].

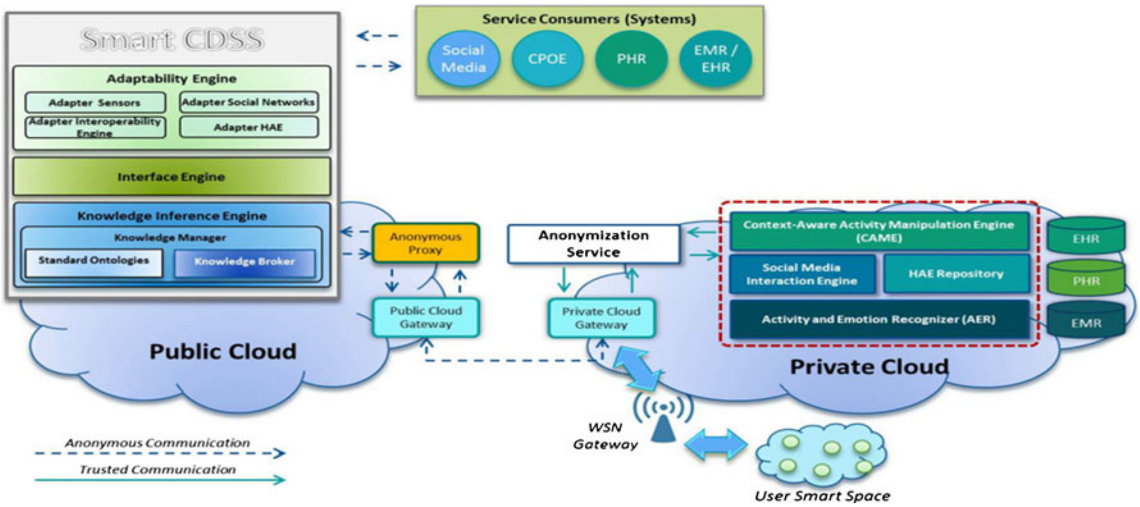

Fig. 1 The overall architecture of our existing SC^3 infrastructure

Activity recognition systems are mostly based on simple conditions and actions [19], reminders for scheduled daily life activities [20], and reminders based on location of the subject [21]. These systems lack the use of context, which is

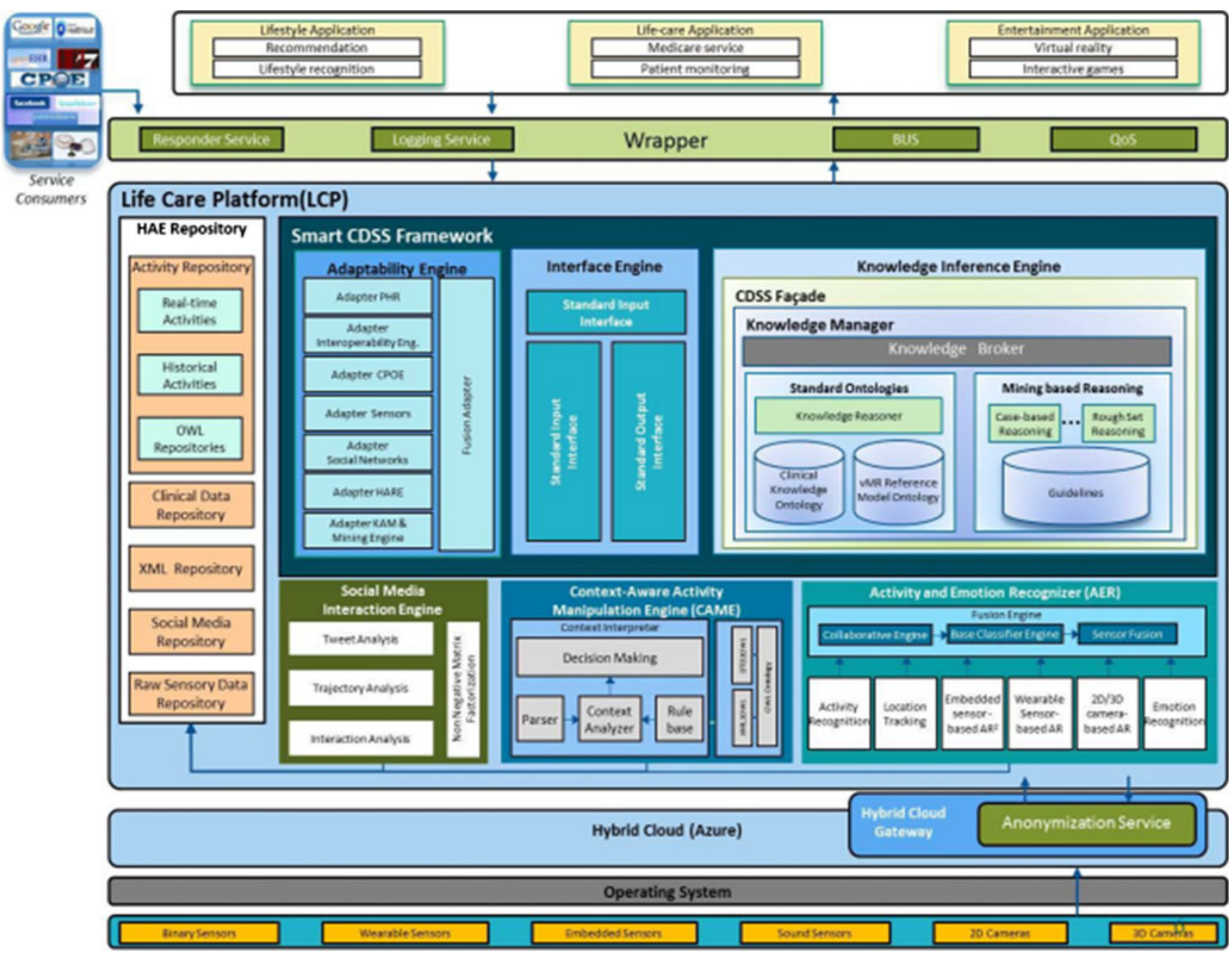

Fig. 2 The framework architecture for the proposed Smart CDSS

incorporated in HYCARE [22] for building an activity scheduling system. The ontology-based reminder system was introduced in [23], which incorporates rules for manipulating the recognized activities of the elderly and then provides them with intelligent reminders and recommendations for performing their activity. However, most of the existing systems are based on one-input modality and in some cases use imperfect context information. Our focus is to use multiple diverse sensors for low-level activity (context) recognition and then use ontology to infer the actual context of the activity. From the low-level activity recognition and the derived high-level context, our existing human activity recognition engine [11] uses multiple and diverse sensors for accurate activity recognition and situation analysis, as shown in Fig. 3.

Our proposed human activity recognition system [11] incorporates video, accelerometer, location, physiological data, patient profile information, and context information using ontology to improve the robustness of high-level activity recognition. To recognize the activities using the data collected from different deployed sensors, we have developed different activity recognition algorithms with better results and coverage than existing algorithms. In addition, we incorporate identification of user emotion in the system's overall working. Users' emotions are recognized to help in identifying their moods in particular instances while performing certain activities. It is also necessary to know user mood before making recommendations.

After the activities are recognized, they are forwarded to a fusion engine [18] that fuses the activities recognized using different input modalities. At a certain time, two different modalities may produce different activity labels for an action performed by a subject. The fusion engine is used to handle such uncertain situations and reach a common consensus. It aggregates the low-level activities to derive high-level context.

With the help of a context-aware activity recognition engine, user profile information and context information are linked in chain with customized rules to achieve higher-level activities [11]. This component infers high-level activities from low-level activities based on user profile information and context information stored in the knowledge repository (discussed later) [13]. For instance, low-level activities in a series, e.g., bending, sitting, jumping, and walking, with the help of ontology will result in higher-level activity, e.g., exercising. The high-level activities are forwarded to the adaptability engine that converts the activity information into vMR format and uses it for clinical recommendations.

3.2 Social media interaction

Social media also plays a significant role in the domain of health care. Social media empowers people to know more about themselves, including their health. For example, research [24] investigated that four out of five users are using the Internet to find personalized health-care information

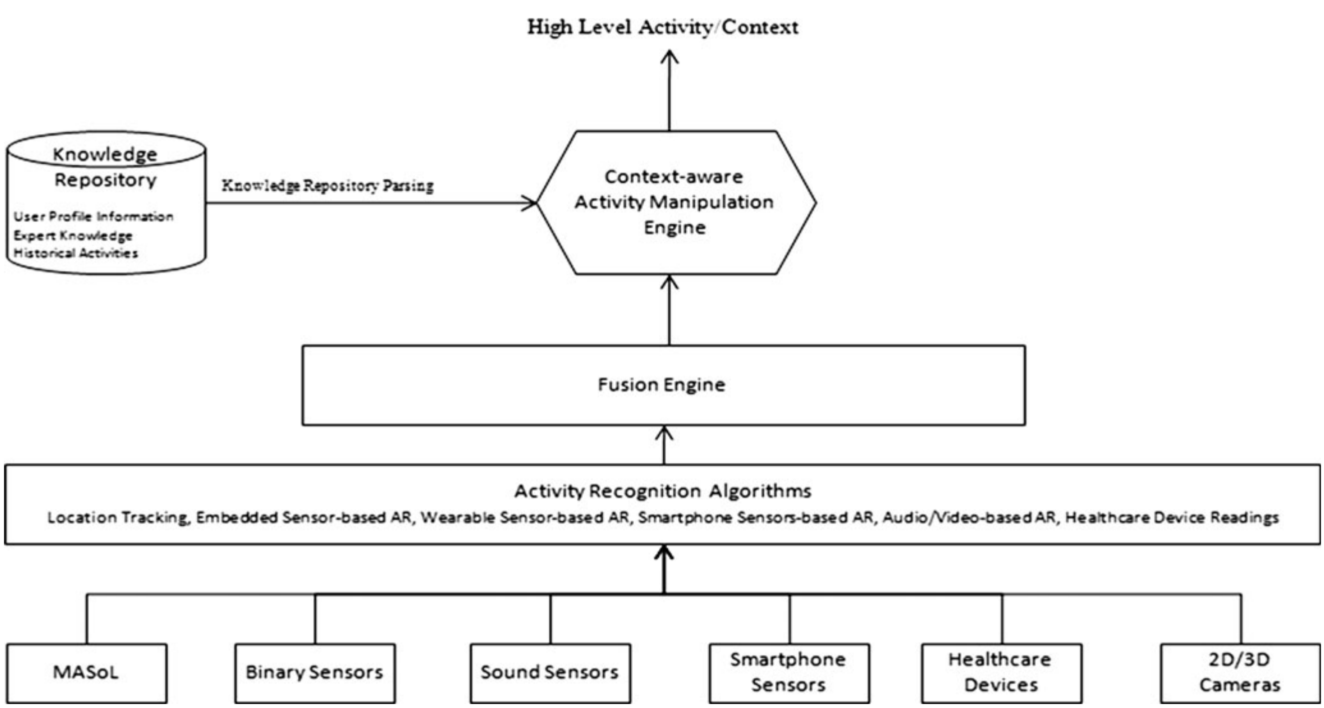

Fig. 3 Low-level activity detection, recognition, fusion, and high-level context extraction

related to a particular disease and its treatments. By knowing more about health, people are more prepared to manage the minefield of modern medical treatment. The problem lies in how patients can effectively use social media to manage their health-related issues as well as how personalization can be achieved by exposing personal health-related issues. For Smart CDSS to provide robust and personalized health-care recommendations, information about the user and user preferences are required because an individual's personality characteristics, environment, life pattern, and emotion reflection are collectively different and have different effects. For Smart CDSS, we process three kinds of social media information—Twitter, Trajectory, and e-mail—in conjunction with the activities data, and these provide a valuable information base for generating personalized health-care recommendations for users. The proposed architecture for social media information gathering and manipulation is shown in Fig. 4, and details about each module are presented in the subsequent sections.

3.2.1 Tweet analysis

Tweet analysis extracts user interests, health conditions, and sentiments from user tweets and provides this information to Smart CDSS. Smart CDSS use this information to provide guidance in any alarming condition, while allowing patients and physicians to intervene and modify treatment plans when required. The tweet analysis module is illustrated in Fig. 5. In the proposed tweet analysis component, the data manager is responsible for fetching data from Twitter and processing the fetched data. The data fetcher sends a request to Twitter for the user stream, and after getting the user's posts from Twitter, the data are passed to a data preprocessor module. Twitter allows users to post up to 140 characters in each tweet, so due to space limitations, people use abbreviations. The slang and abbreviations in tweets affects the information extracted from tweets. To avoid related information loss, the data preprocessor first removes slang and abbreviated words using a slang lexicon. We have a lexicon of 1,300 slang words commonly used in social media posts and chats. The data preprocessor also checks posts containing URLs and separates them from the text to process text and URLs separately. The data manager component forwards the pre-processed data to the knowledge generator. The main purpose of this component is to process users' tweets to extract users' interests, sentiments, and health conditions from Twitter data and store the resulting data in personalized profiles. The personalized profile maintains the history of the user data so it can be used in the future. The knowledge generator uses Alchemy API to extract the user's interests. Alchemy API is a cloud-based text-mining platform that uses deep

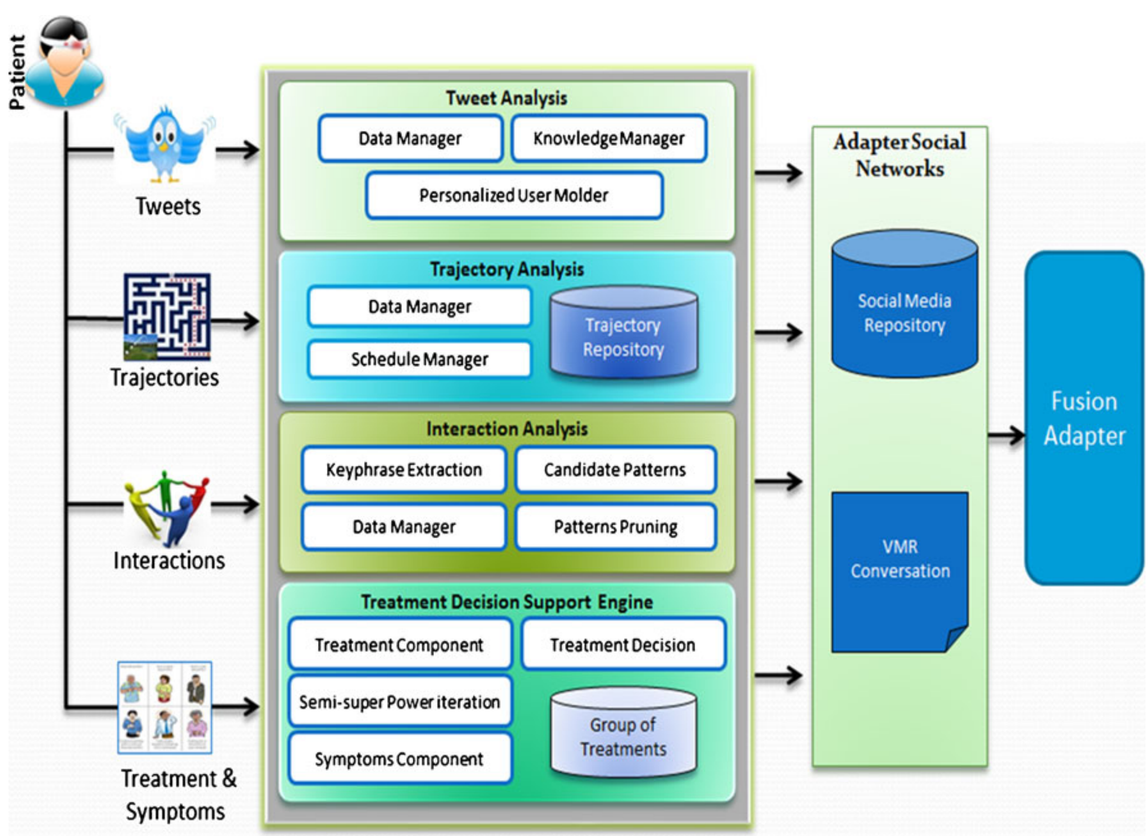

Fig. 4 Social media interaction for Smart CDSS

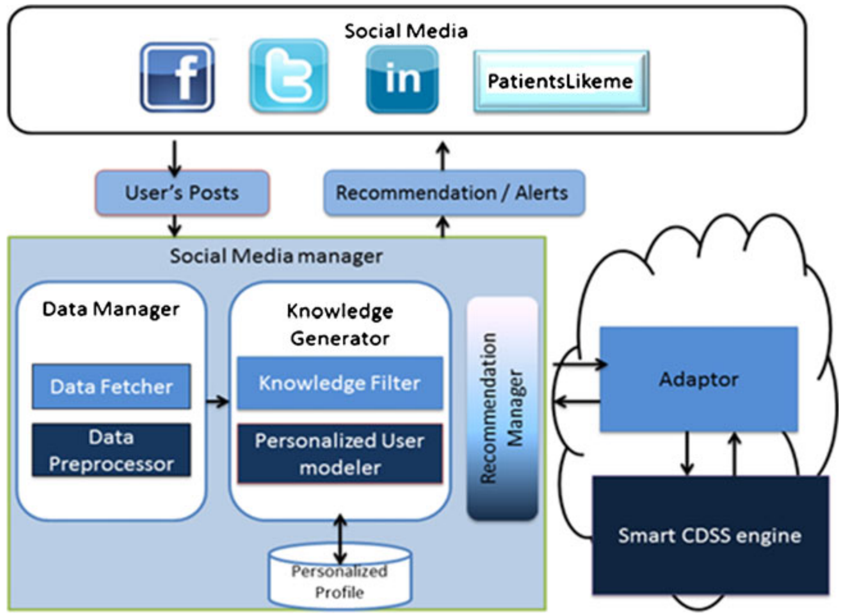

Fig. 5 The architecture for tweet analysis

linguistics parsing, statistical natural language processing, and machine learning to extract knowledge. After processing by the knowledge generator, data are converted into meaningful, usable information. Better recommendations require maintenance of the history of an individual's interests. The personalized user modeler maintains the user's data in a personalized profile and passes this information to Smart CDSS and to experts/clinicians when required.

3.2.2 Trajectory analysis

We also track the outdoor activities of the user to help in personalized recommendations generation. These activities are tracked using GPS-enabled location-aware mobile devices, such as smart phones. Our focus is to record the locations where significant activities are performed. GPS coordinates of each tracked position are sent to the imperative location finder, where a decision regarding the importance of the activity is made. If the activity is considered significant, its location coordinates are converted into geographic tags using Google API. A semantic tag of the location for contextual awareness is also acquired by patients. Similarly, whole-day routine activities of the user are collected and stored in the followed patterns in trajectory data, as shown in Fig. 6.

A prescribed schedule and suggestions for carrying out each activity according to the health and ailment condition of the patient are added by the practitioner. The prescribed schedule is stored in the prescribed patterns, and suggestions regarding the individual's activity are stored in the recommendations. The difference between the prescribed schedule and the recommendations is the prescription of daily routine activities. For example, the prescription specifies the amount of time to spend and the frequency of user visits to a particular place, whereas the recommendations suggest individual activities, such as the kind of food the user should get and the type of exercise suitable for the patient.

Comparison of the prescribed schedule with the user-followed schedule is performed in activity analysis, and inconsistencies are converted into vMR format by the vMR conversion module. An example output format of the vMR conversion module for a diabetes patient is given to the adaptability engine. Providing suggestions of some useful information prior to an activity's happening is much more valuable than stating it afterward [25, 26]. The imminent activity advisor seeks to predict forthcoming activities of the patient based on analysis of past data. Upon every change of position, historical statistics are evaluated to discover the activities of the user in the related time period. The activity with the maximum probability of occurrence is selected as an immediate forthcoming activity.

3.2.3 Interaction analysis

In the interaction analysis module, we mine users' frequent and periodic interaction patterns that change over time. The purpose is to gain knowledge about the preferences, needs, and habits of the user. Users can act in two different roles: senders and receivers. These two roles are not interchangeable. We propose a two-phase strategy to identify the hidden structures shared across different dimensions in dynamic networks, such as type of interaction, time of interaction, interaction interval, and interaction response based on priorities. We extract structural features from each dimension of the e-mail network via periodic and frequent interaction mining, and then integrate them to ascertain robust patterns

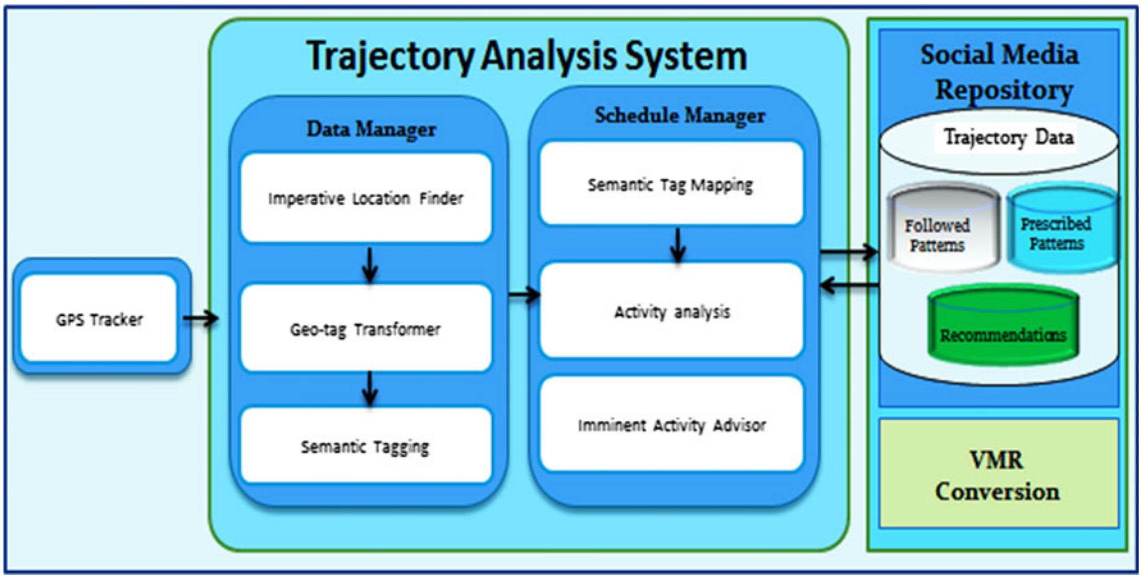

Fig. 6 Trajectory analysis system

about users, as shown in Fig. 7. Furthermore, with the right formal definition of what constitutes periodic behavior, the aggregate periodicities of an entire set of mined interaction patterns can assist Smart CDSS in better recommendations generation. Therefore, learning users' common behaviors becomes an important step towards allowing Smart CDSS to provide personalized recommendations more accurately and effectively. The social media adapter plays a vital role in integrating the output of interaction analysis with Smart CDSS. It is a cloud-based application that works as a connector and performs automatic mapping of obtained information into the virtual medical record (vMR). The detailed flow of each sub-module is as follows.

Keyphrase extraction The keypharses are extracted from the contents of the e-mail by using Algorithm 1. Extraction of the semantic keyphrases is an essential requirement of accurate data modeling based on the user interactions. First, all parameters of the extraction algorithm KEA++ [27] are set with respect to keyphrases' lengths in the taxonomy and lengths of the documents. Second, KEA++ is trained on the set of e-mails using the taxonomy. Finally, KEA++ is applied on

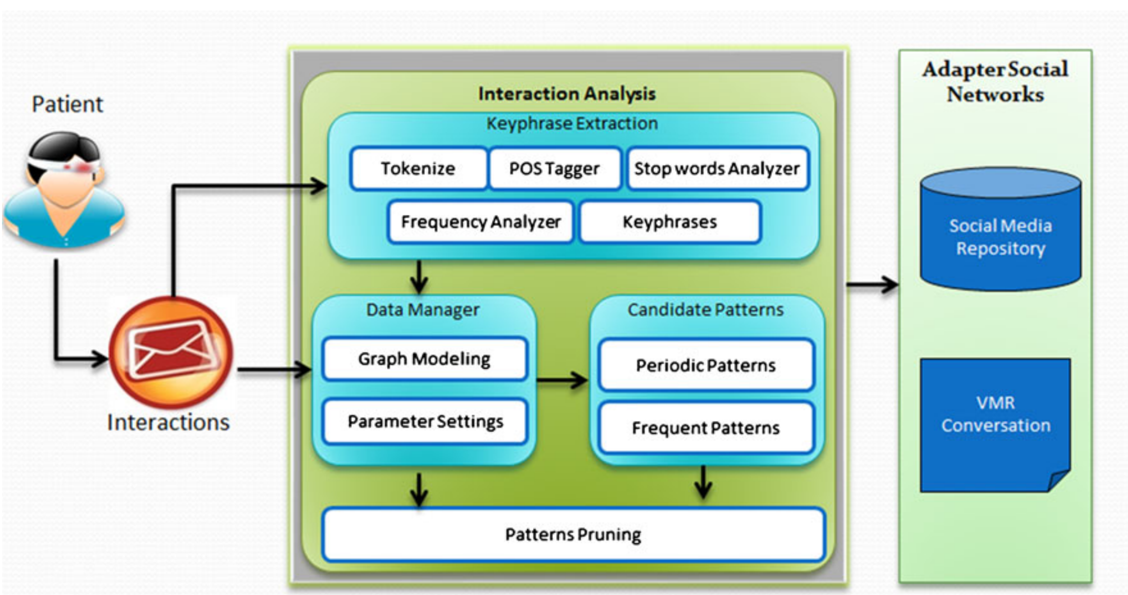

Fig. 7 The proposed architecture for interaction analysis for Smart CDSS

actual e-mails (data). Initially, the e-mail contents are tokenized by using the POS tagger and the stop-words analyzer. The POS tagger assigns parts of speech to each word, such as noun, verb, or adjective. Stop words can cause problems when searching for phrases using exact matching. So stop words, such as *the*, *is*, *at*, *which*, and *on* are removed from the e-mail contents. The frequency of each word in the e-mail is counted, and then KEA++ returns the relevant keyphrases. Each keyphrase returned by KEA++ is processed and assigned its level label in the taxonomy. Identifying level labels is required before applying the refinement rules because they represent the hierarchical order of the keyphrases. If the keyphrases are labeled as training-level, they are retained in the results set, according to steps 5 to 12 of Algorithm 1. Lower-level keyphrases are stemmed to their training-level keyphrases and kept in the results set if they are associated with the general category at a lower-level in the taxonomy. Otherwise, lower-level keyphrases are discarded. Upper-level keyphrases, which belong to the same level as training-level keyphrases, are discarded after identifying and preserving their equivalent keyphrases from the taxonomy. If the initial results do not contain any training-level keyphrases, then lower-level keyphrases in the results are preserved and added in the final refined result. This process is executed in steps 13 to 21 of the algorithm. Finally, redundant keyphrases are removed from the final refined set of keyphrases.

```
Input: Training:
    (a) Set the parameters of KEA++ by keeping in view the keyphrase length in the taxonomy and email type
    (b)email along with their keyphrase and taxonomy
  Dataset for Extraction:
    (a) email with unknown keyphrases
Output:Set of refined keyphrases

1: TrainLevel ← KEA++ TrainLevel //(Rule I)
2: resultSet [] ← returnedkeyphrases by KEA++[]
3: resultSet [] ← level labels (Resultset [])
4: forresultSet[] <> empty do
5:         if(resultSet(training level)) then
6:         if(keyphrase level = lower level keyphrases) then
7:                 processSet[] = preserving lower level keyphrases //Rule II
8:         else
9:                 processSet[] ← identifying and preserving training level equivalent //Rule V
10:                processSet[] ← remove redundant keyphrases //Rule VI
11:                refineSet[] ← processSet[]
12:        end if
13:        else
14:        if(keyphrase level = training level) then
15:                refineSet[] ← processSet[]
16:            else
17:                if (keyphrase level = upper level) then
18:                processSet[] ← identifying and preserving training level
                           equivalent keyphrases //Rule V
19:                else
20:                processSet[] ← stemming lower level general keyphrases //Rule II
21:            end if
22:                processSet[] ← remove redundant keyphrases //Rule VI
23:                refineSet[] ← processSet[]
24:        end if
25:        end if
26:end for
27: returnrefineSet[] //refine result set of semantic keyphrases.
```

Algorithm 1. Keyphrases Extraction from E-mail Contents

Data manager The data manager module helps in data modeling and parameter setting before applying the mining algorithm. It extracts a population of interest from messy e-mail interaction data by removing noise. The extracted information is modeled in graphs based on user-defined interaction intervals and extracted keyphrases. In each graph, nodes are the individuals with keyphrases as node labels, and the directed edge represents the interaction between them. The graph is modeled as a dynamic graph in which the times of interaction are presented as edge labels. Our graph model can easily be separated based on user-defined parameters, such as days, months, and years for identification of patterns of interest. Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all sets of actions that have higher confidence levels than the minimum confidence are considered as basic frequency periodic patterns.

Candidate patterns The candidate patterns module identifies a set of frequent and periodic patterns from the e-mail interaction graphs. The frequent patterns (FP) emphasize the significance of patterns, and the periodic patterns consider their regularity. Frequent patterns are mined by using an FP tree-based approach, while periodic patterns are mined using PSEMiner with integration optimization. First, we identify the frequent patterns, and then these patterns are checked for periodicity. If a pattern is both frequent and periodic, then it is regarded as a pattern of interest. The objective is to identify the sets of actions that frequently and periodically occur together. Once basic frequent periodic patterns have been discovered, an aspect to consider is whether there is any action that is present in some, but not all, of a particular pattern's occurrences, and so it should be included in the pattern. Taking into account all the periods, while it may not be discovered in the previous task, it is frequent enough overall in those periods where the basic frequent periodic patterns occur to be included.

Patterns pruning The patterns pruning module applies a mining process to identify frequent and periodic patterns under the given parameter settings. Patterns pruning reflects the common characteristics of a typical e-mail interaction that has some unusual associations among the patterns. To achieve this, initially, the candidate patterns are transformed into an integrated set to make it comprehensively useful. Briefly explained, the module infers meaningful actions from the data collected in e-mail data, and then it splits the string of actions into periodic sequences based on some frequency support. Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information. So the extracted patterns reflect the habits and preferences of the user according to periodic trends that are building in frequency. Therefore, learning the user's lifestyle is an important step toward allowing Smart CDSS to provide personalized services more accurately and effectively. The patterns of interest after pattern pruning are converted into vMR format and passed to the Smart CDSS system through the social network adapter.

In this way, all the user's social interaction and experience are processed, including short text posts like Tweets, outdoor activities like lunch and shopping, and e-mail interactions, and meaningful information is extracted. All the extracted information that is useful for the Smart CDSS recommendations generation module is forwarded to Smart CDSS for manipulation and storage.

3.3 Knowledge repository

The knowledge repository (KR) is the backbone of Smart CDSS. One of the main components of Smart CDSS is to store and manage all the information flow to/from the proposed system. The information in the KR is modeled using an ontology and evolves with newly arriving activity and social media information. The information in the KR is semantically linked for subsequent processing using the inference engines of different modules at different stages of the proposed system. The abstract model for concept hierarchy in the build ontology is given below.

$$\begin{aligned}
O \equiv\ & \exists \mathrm{Activity}.(\mathrm{DetectedActivity} \sqcup \mathrm{SocialInteraction}) \\
& \sqcap \mathrm{Activity.relatesTo} = \exists \mathrm{Daily_Life_Activity} \\
& \sqcap \forall \mathrm{Activity.hasPerformer} \\
& = 1\mathrm{Actor.has}(\mathrm{Preferences} \sqcap \mathrm{SocialInteractions}) \sqcap \exists \mathrm{Activity.has} \\
& (\mathrm{Activity} \sqcup \mathrm{Effects} \sqcup \mathrm{Actions} \sqcup \mathrm{Location} \sqcup \mathrm{Duration} \sqcup \mathrm{Condition})
\end{aligned}$$

The KR stores raw data collected using different sensors and social media technology, and it stores real-time activities recognized by activity recognition engines and information retrieved from the social media raw data, activity history, user profile information, standard clinical data, expert knowledge, and expert recommendations. The information contained in knowledge repository is in machine understandable format (i.e., OWL) which is used to make inferences and generate smart recommendations.

The activity history and user profile information is used by the inference engine of the context-aware activity manipulation engine (CAME) to infer high-level activities that categorize the currently detected low-level activities. These activities and user information are represented in OWL format. From the activity repository, CAME, with the help of the rules and inference engine, manipulates these activities to infer the associated high-level activities [13]. Figure 8 shows the OWL representation (using N3 notation) of the "Walking" activity of a user named "Asad" in the knowledge repository.

In addition to the activity and user information, the KR also contains clinical information that is compliant with the HL7 standard. The internal workings for recommendations in Smart CDSS are based on the standard data format. So the expert knowledge and recommendations are also represented in standard representations; i.e., Arden syntax. Recommendations (rules) have two main components—condition and action—that make the recommendations easy to model. Representations of condition and action components are different under different rules. The Arden syntax uses *if* and *then* constructs; however, the definition of these constructs were imported from the *Data* and *Action* segment of the rule, while the rule itself is defined in the *Logic* segment. Figure 9 shows an example of the Arden syntax rule's logic.

3.4 Interface engine

The interface engine (IE) defines standard interfaces that allow external systems to interact with Smart CDSS. The

Fig. 8 OWL representation (using N3 notation) for walking activity including user information

```
activityOnto:Person_Instance_654
aactivityOnto:Person ;
activityOnto:hasactivityOnto:Patient , activityOnto:PhD , activityOnto:High_Age;
activityOnto:hasAge                  68;
activityOnto:hasDesignation   activityOnto:Professor;
activityOnto:hasID                   654;
activityOnto:hasName                  "Asad".

activityOnto:Activity_Instance_20090614140013347
aactivityOnto:Activity ;
activityOnto:hasConsequentAction     activityOnto:Rule;
activityOnto:hasID                   347;
activityOnto:hasName                 "Walking";
activityOnto:hasType"Motion";
activityOnto:isA                     activityOnto:Room_Instance_Living;
activityOnto:performedAtTime   2010:11:14:14:00:18;
activityOnto:performedBy       activityOnto:Person_Instance_654.
```

standard input interface allows external system input to be represented in standard HL7 vMR format. The standard output interface specifies recommendation, alerts, and guidelines in standard HL7 vMR output format. To ensure the validity of inputs and outputs, standard XML schemas are defined based on the HL7 vMR standard and applied as a contract to the web service WSDL contract. The proposed system is developed as a WCF (Windows Communication Foundation) web service, which properly implements standard input and output interfaces using HL7 vMR format.

HL7 vMR format allows the standardizing of interfaces for CDSS. It specifies the inputs and outputs in terms of patient information and observations as clinical statements. HL7 vMR is a comprehensive but simple representation, with which patient information is included as a root and related observations become sub-nodes of the root with the corresponding clinical statement. Clinical statements have been categorized as problem, observation (clinical findings), adverse event, encounter, substance administration (medications), goal (glucose level), procedure (surgery), and supply (dose of radiation).

3.5 Knowledge inference engine

The knowledge inference engine (KIE) is a core component that uses knowledge bases to generate recommendations and alerts based on consumer input. The knowledge broker facilitates in choosing the appropriate knowledge base for triggering recommendations and alerts. Two knowledge bases are incorporated to test the system on diabetic patient datasets: the Clinical Knowledge Base (CKB) and the Non-Clinical Knowledge Base (NCKB).

The CKB contains rules of actual practices that are taken from online medical resources (diabetic guideline resources) and physician experiences. The rules are represented as medical logic modules (MLMs) in HL7 Arden syntax. Diabetes rules are mapped to MLMs by defining one MLM for each medical rule. The NCKB contains rules derived from a dataset of patients formed with the case-based reasoning technique. For this paper, the described experiment only reflects execution of the CKB rules that provide recommendations for diabetes patients.

The Smart CDSS knowledge base contains rules published by clinicians in HL7 Arden syntax. Each rule constructs some recommendations, alerts, or guidelines, represented in MLMs. These modules have well-defined structures to represent rules. For example, the “maintenance” section defines the title of the MLM, the author of the MLM, and the date and time of the creation of the MLM. The “library” section contains information that describes the purpose of the MLM. The “knowledge” section holds the data and logic of the MLM that will generate recommendation or alerts. Figure 10 shows a partial MLM that is used to diagnose diabetes mellitus based on glucose observations.

The KIE executes rules compiled from Arden syntax to executable C# classes. Each MLM has a counterpart C# class, as shown in Fig. 11, which has an index with the KIE based on event type. So, for each input from an external application, KIE schedules all related MLMs for execution. MLMs are

Fig. 9 Example of the Arden syntax rule’s logic

```
if (Catheter is present AND time of Catheter is at least 7 days before Urine_culture_time) then
        ifOne_urine_culture then
        conclude true;
        endif;
        else if Two_urine_cultures then
        conclude true;
        endif;
endif;
```

Fig. 10 Diabetes MLM for diagnosing diabetes

```
maintenance:
title: Diabetes Mellitus
library:
purpose: "Finding VPGC[Venous Plasma Glucose Concentration] to identifyDiabetes Mellitus (DM)
knowledge:
logic:
          If [fasting and VPGC =126 (mg/dL)] OR
                    [2-hour post 75 g glucose load and VPGC = 200 (mg/dL)]
          then
                    Diabetes Mellitus (DM)
          Else if [fasting and VPGC = 100 and < 126 (mg/dL)] AND
                    [2-hour post 75 g glucose load and VPGC < 140 (mg/dL)]
          then
                    Impaired Glucose Tolerance (IGT)
          Else if[fasting and VPGC < 126 (mg/dL)] AND
                    [2-hour post 75 g glucose load and VPGC = 140 and < 200 (mg/dL)]
          then
                    Impaired Fasting Glycaemia (IFG)
```

executed one by one, and the results are accumulated into a single recommendation as standard output using HL7 vMR format derived from vMR ontology.

3.6 Adapter interoperability engine

One of the key aspects of Smart CDSS is its interoperability with health management information systems (HMISs), which are compliant with various, differing health-care standards. Smart CDSS service can only be utilized by HMISs if processing of information is performed in their respective interpretable formats. The adapter interoperability engine (ARIEN) resolves the heterogeneities among HMIS formats so they interact easily with Smart CDSS and utilize its services. The ARIEN is a subcomponent of the Smart CDSS adaptability engine and behaves as a mediator between Smart CDSS and HMISs. Because they are compliant with different health-care standards, HMISs understand only standardized formats, such as HL7 CDA, openEHR, and CEN 13606, while Smart CDSS can only process virtual medical record (vMR) format. The ARIEN provides bridge services that use ontology matching techniques to generate mappings between heterogeneous health-care standards for automatic transformation of information to enable interoperable communication among health-care systems. A demonstration of resolving heterogeneities between HL7 and openEHR standards at the model level was presented in our previous work; please refer to [28].

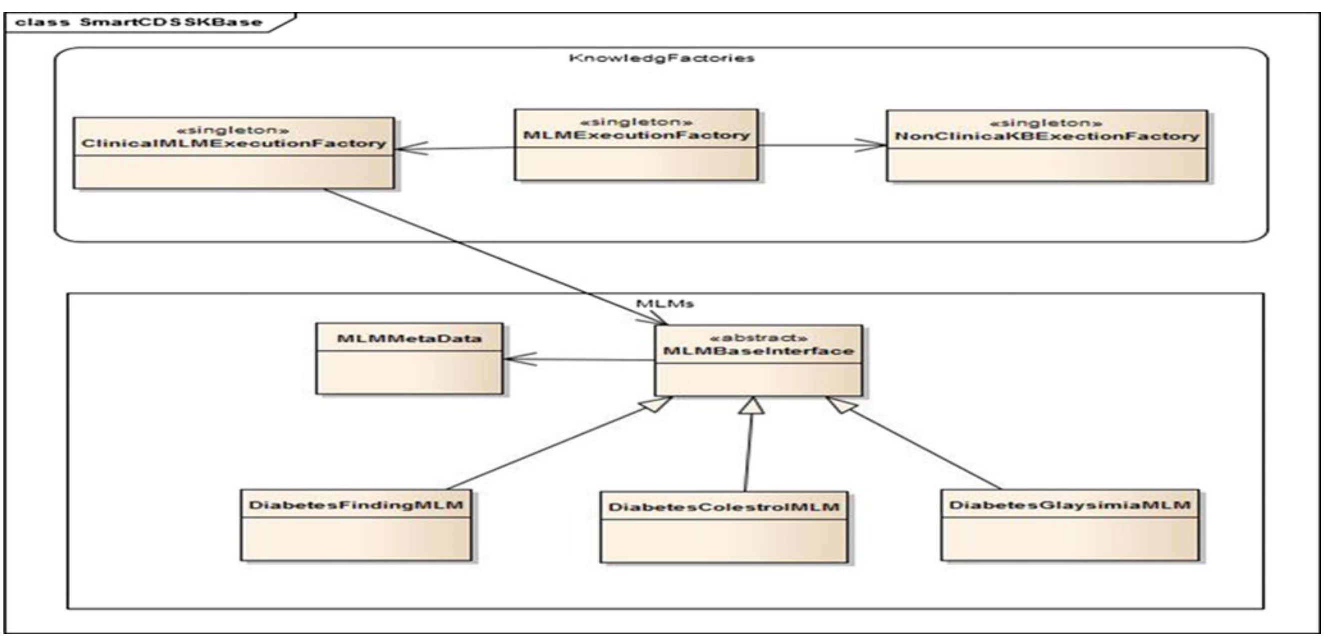

Fig. 11 Smart CDSS knowledge base abstract design

The ARIEN takes input in a standard form (e.g., HL7 CDA) and transforms it into vMR format using already generated and stored ontology mappings in the repository. This transformed vMR format is processed by Smart CDSS for generating recommendations. After processing, the information must then be converted back from vMR to the HMIS-compliant standard format. The AREIN performs this conversion and passes the information to the HMIS in standard format. With these procedures, transformation between HL7 CDA and vMR is successfully completed, with some deficiencies, while openEHR and vMR transformation is underway. However, complete mappings cannot be generated using ontology-matching techniques; therefore, manual mappings are also used to increase the accuracy of mappings in transformations.

The abstract architecture of the ARIEN system is shown in Fig. 12, with the modules of the ARIEN that are used for achieving interoperability. The ARIEN system performs three main steps. First, the ontology repository is used to store standard ontologies that are used by the accuracy mapping engine, which generates ontology mappings to be stored in the mapping file repository. Second, the standard ontology change management module identifies changes in ontologies from the ontology repository, generates mappings based on changes, and embeds those changes in the mapping file repository. Finally, when the HMIS wants to use Smart CDSS service and is compliant with a particular standard (HL7 CDA in our case), the information is provided to the transformation engine that uses already-stored mappings from the mapping file repository to transform the information into vMR format for further processing.

The main objectives of the ARIEN are accuracy and continuity of mappings. Accuracy is achieved using ontology matching with a manual matching scheme; continuity of mappings is realized through ontology change management.

Because final recommendations and guidelines are provided to clinicians and users by Smart CDSS with the information in vMR format, all incoming information must be converted into vMR format. The fusion adapter is responsible for integrating the incoming data from diverse sources into a single vMR. Figs. 13, 14, and 15 show the social media information from Tweet, Trajectory, and interaction analysis, respectively, in vMR format.

For the scenario provided in Fig. 13, patient information extracted from Tweets regarding depression and diabetes are represented as "Problem" clinical statements. The problems are represented in SNOMED CT code to achieve interoperable exchange between the health-care system and CDSS. In the same way, observations extracted from Trajectory analysis and interactions analysis are represented as "Problem" clinical statements, as shown in Figs. 14 and 15.

3.7 Cloud-based Smart CDSS

The "modular build" of Smart CDSS requires a reliable infrastructure that provides scalability with usage, distributed data processing, and security and privacy for the individual's data. Cloud computing, with its service and deployment models, can support these needs of Smart CDSS. To demonstrate this, we show Smart CDSS being deployed under the principles of the hybrid cloud model. Components with accessibility to private data are deployed over a private cloud, while components providing user and social interaction are deployed over a public cloud.

Figure 16 illustrates the deployment stack of Smart CDSS on a private cloud infrastructure. This infrastructure is scaled over a particular location like smart spaces and smart homes, equipped with various sensors. An individual living in the smart space can interact with the private cloud via sensors and services. The private cloud provides context-based (CAME) and activity recognition (AR) services. Data collected by CAME and AR over a certain period of time are sensitive and private in nature. These data persist and are partially catalogued over the private cloud's storage resources. However, the private cloud's limited resources and associated costs do not qualify it to be a permanent storage solution. For data cataloguing, the public cloud's less expensive and much larger storage resources are utilized.

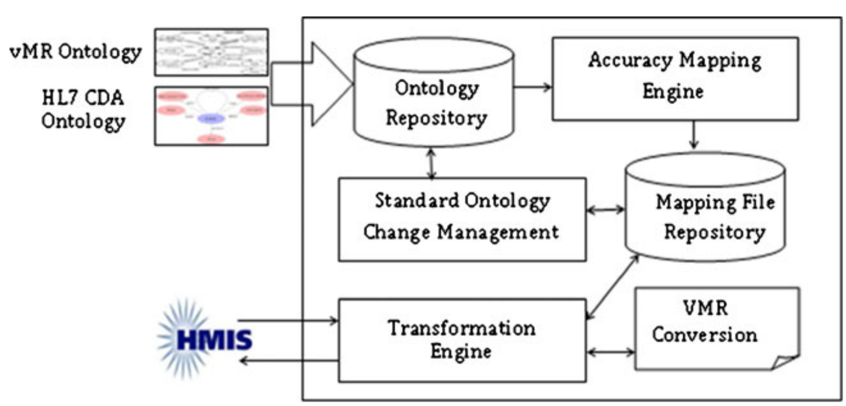

Fig. 12 ARIEN system architecture

```
<vmrInput>
  <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
  <patient>
    <clinicalStatements><!--Health problem>
      <problem> <!--Depression -->
        <id root="TW_107c2dc0-67a5-11db-bd13-0800200c9a66"/>
        <problemCode codeSystem="2.16.840.1.113883.6.96"
      codeSystemName="SNOMED CT" code = "291746001" />
      </problem>
      <problem> <!-- diabetes-->
        <id root="TW_10dca459-3249-29db-bd31-0800200c9a66"/>
        <problemCode codeSystem="2.16.840.1.113883.6.96"
      codeSystemName="SNOMED CT" code = " 73211009 "/>
      </problem>
    </clinicalStatement>
  </patient>
</vmrInput>
```

Fig. 13 vMR format for tweet analysis

Making sensitive data available via a public cloud increases the chance of potential information leakage because of the cloud's accessibility and ubiquity. In the worst case, an untrustworthy cloud provider can become a source of information leaks. To ensure privacy even on a public cloud, anonymization services (discussed later) decouple data from the user's identity.

Recommendations generation components and data catalogues including knowledge bases are hosted over the public cloud infrastructure, as illustrated in Fig. 17. With their scalability over large resources and inexpensive cost model, public clouds are the ideal fit for the components of Smart CDSS. The public cloud provider market is rapidly growing, led by many high-tier technology companies, for example, Amazon's EC2, Microsoft Azure, and Google's AppEngine. Adoption of a public cloud provider involves far more than economics. Clearly, development platform compatibility is among the most important factors for public cloud adoption. The proposed Smart CDSS is implemented using Microsoft's development tools and technologies, and the public cloud

```
<vmrInput>
  <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
  <patient>
      <clinicalStatements><!--Health problem>
  <problems>
  <problem> <!-- Abdominal weakness-->
      <id root="107c2dc0-67a5-11db-bd13-0800200c9a66"/>
      <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"
code = "162239000" />
  </problems>
<substanceAdministrationEvents>
  <substanceAdministrationEvent>
      <substance> <!—Exercise assessment not completed-->
          <id root="2d813912-a3d1-464c-9567-4d92c75a0c23"/>
          <substanceCodecodeSystem="2.16.840.1.113883.6.96" codeSystemName=
"SNOMEDCT" code = "413161009" />
      </substance>
      <documentationTimelow="20120216"high="20120216"/>
  </substanceAdministrationEvent>
</substanceAdministrationEvents>
      </clinicalStatement>
  </patient>
</vmrInput>
```

Fig. 14 vMR format for trajectory analysis

```
<vmrInput>
    <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
    <patient>
        <clinicalStatements><!--Health problem>
            <problems>
            <problem> <!--Muscle Fatigue-->
                <id root="107c2dc0-67a5-11db-bd13-0800200c9a66"/>
                <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED
            CT" code = "56946009" />
            </problem>
            <problem> <!-- Depression -->
                <id root="10dca459-3249-29db-bd31-0800200c9a66"/>
                <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED
            CT" code = " 291746001 "/>
            </problem>
        </problems>
        </clinicalStatement>
    </patient>
</vmrInput>
```

Fig. 15 vMR format for interaction analysis

infrastructure is provided by Microsoft's Azure Platform. The Smart CDSS components on the public cloud are classified into four categories:

1. Web applications and services that include Smart CDSS UI, Smart CDSS Web services, and anonymization service. These components are deployed in web roles over Windows Azure.
2. Service libraries that include Smart CDSS component engines (inference engine, CAME, interoperability engine and adaptability engine). These components are deployed in worker roles over Windows Azure.
3. A system-level service bus that includes knowledge inference and broker services. These components are deployed over Azure AppFabric.
4. Knowledge bases, ontologies, and stored procedures are deployed over SQL Azure and accessed via data services between Windows Azure and SQL Azure.

3.8 Security and anonymization

Cloud computing is an epitome of on-demand computing. It provides virtualized computing resources (i.e., processing power, storage facility, application services) over the

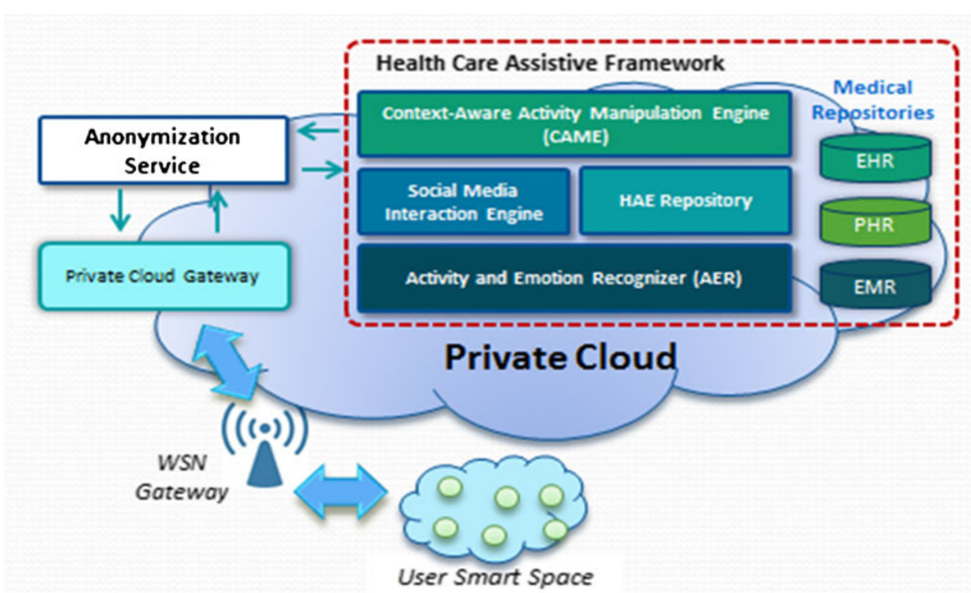

Fig. 16 Smart CDSS private cloud deployment model

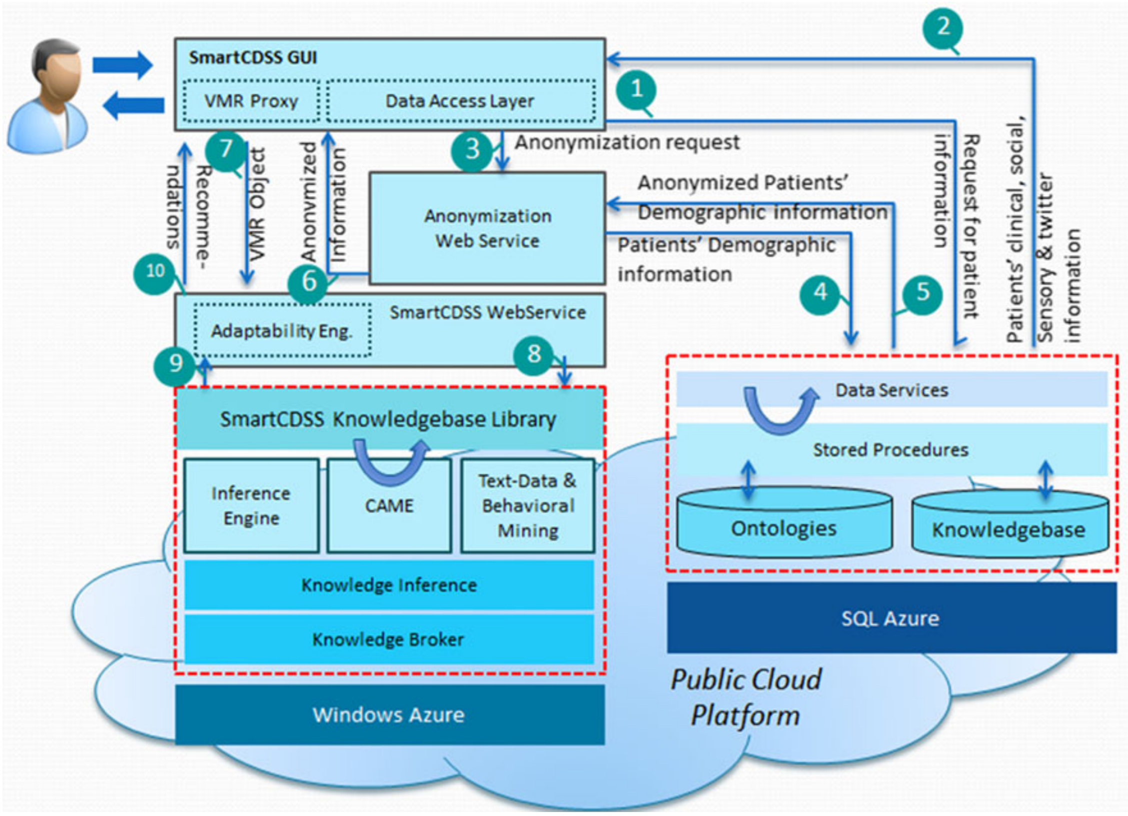

Fig. 17 Public cloud deployment model for the proposed Smart CDSS components

internet. However, as cloud is owned and managed by a third party, the risk of privacy infringement escalates when confidential data of the proposed Smart CDSS is outsourced to an un-trusted domain of cloud service provider. The most common approach to ensure data confidentiality is through encryption.

Although encryption ensures data confidentiality; however, it restrains data processing (i.e., searching, processing, and modification) over encrypted data. For Smart CDSS, to process data residing within the un-trusted domain of a cloud service provider, the system is left with two options. First, share data decryption key with the cloud service provider, once processing is completed data is re-encrypted with new encryption key. Second, download the entire encrypted data to a private cloud, decrypt it with valid data decryption key and then process it.

Both of the options discussed earlier are not feasible for data intensive applications, such as Smart CDSS, where huge amount of data is required to be processed. In order to ensure data privacy for Smart CDSS there is a need to anonymize the data. Anonymization ensures that data residing within the un-trusted domain of a cloud service provider cannot be linked to actual users (i.e., diabetic patients). With anonymous data outsourced, there is no need to encrypt data in order to achieve data privacy. Figure 18 illustrates the conceptual model of our proposed anonymization for Smart CDSS service on hybrid cloud configuration.

3.8.1 Working of anonymization

Anonymization service de-identifies the data, by replying actual identities with pseudo identities. This ensures that the user cannot be linked with the data residing within in a cloud service. Anonymization service provides mapping between the real identities and pseudo identities of the user. This mapping is stored on secure location in order to avoid risk of privacy infringement in any adverse event. Figure 19 shows the architecture of anonymization service, which provides a bridge between the mapping stored locally on secure server and data outsourced to public cloud for clinical processing.

Anonymization service is further composed of four cohesive components ensuring that data migrating to cloud service is properly de-identified and cloud service provider cannot exploit it to compromise privacy of a diabetic patient. Anonymization service is composed of Identity Management, Consent Management, and Consent Validator.

Identity management Identity management is responsible for de-identifying the user information. It generates pseudo

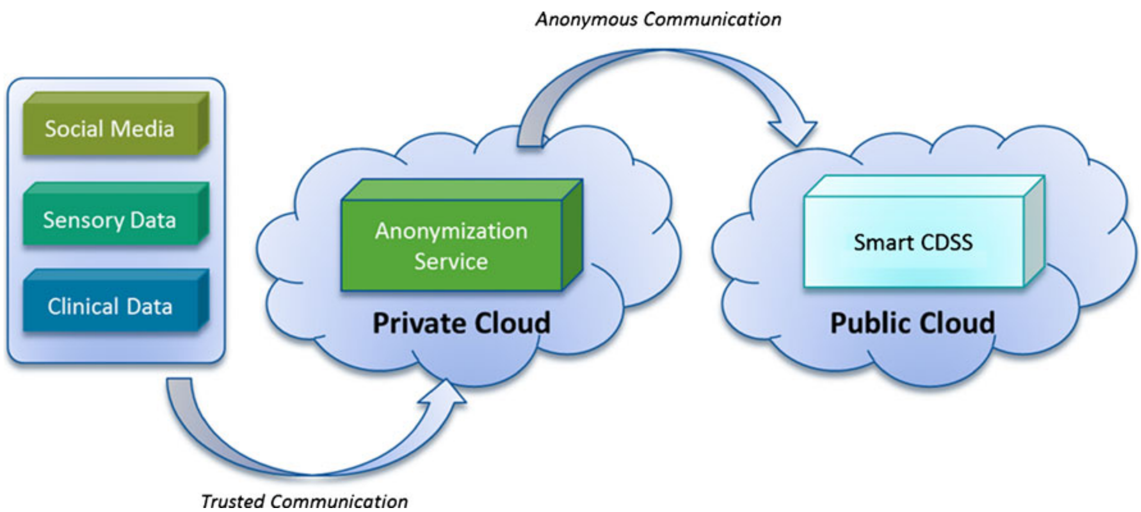

Fig. 18 Conceptual model of anonymization service for the proposed Smart CDSS

identity (i.e., virtual identity) for each user. It then securely stores the mapping between the virtual identities and real identities. These mappings are only known to identity management component. This ensures that whenever Smart CDSS requests data access, its real identities can only be swapped with virtual identity by identity management component. Identity management is composed of two sub components, virtual identity generator and identity mapping (see Fig. 19). Virtual identity generator is a secure random string generator which generates randomized identity for each request. This ensures that for a single patient new pseudo random identity is generated on each request to Smart CDSS's services. This restrains the capability of malicious cloud service provider to link Smart CDSS request to a particular category user. Identity mapping is a repository that securely stores the mapping between the virtual identities and real identities of diabetic patients. These mapping are stored in encrypted form to ensure that even if the anonymization service is compromised, an attacker cannot gain access to the mapping repository.

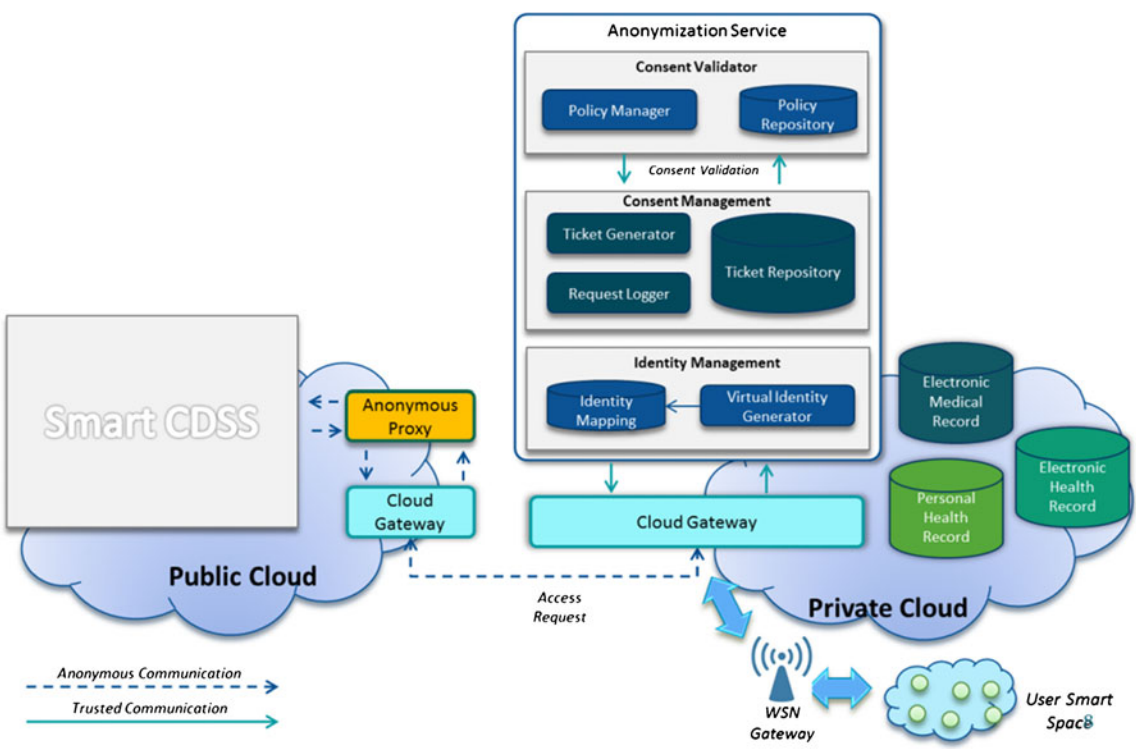

Fig. 19 Anonymization service for Smart CDSS

Consent management Consent management is responsible for enforcing access control policies during data access requests to the cloud storage. It not only enforces access control policies but also ensures that requests are generated by authorized subscribers and components. Consent management comprises a ticket generator, request logger, and ticket repository (see Fig. 19). Ticket generator is responsible for evaluating the access control policy and making decision on the outcome of access control policy evaluation. If policy is evaluated successfully, a valid access ticket is generated and digitally signed by the consent management; otherwise, the access request is rejected,, restraining unauthorized access to the outsourced data. The access ticket contains information about the data access request. It also encodes the validity of the access ticket to ensure that it cannot be used multiple times, to prevent reply attacks. The request logger logs each data access request to ensure a proper audit trail. The log includes a time stamp of when a request is generated, the resource being requested, information of the requester (i.e., subscriber information), and the outcome of the access control policy's evaluation. This detailed log of data access requests ensures that the cause of failure or malicious activity can be effortlessly traced by a system administrator, in the case of any adverse event (i.e., security attack, system failure). Tickets generated by the ticket generator are maintained (persisted) in the secure ticket repository. Persisted tickets are used to de-anonymize the data once a request is processed by Smart CDSS in a public cloud. This ensures that in the trusted domain of a private cloud, processed data can be linked with users in order to execute Smart CDSS workflows properly.

Consent validator The cloud service provider is a non-trusted entity. Thus, to ensure authorized data access and request execution rights, Smart CDSS cannot rely on cloud service provider. Otherwise, the provider could authorize a malicious subscriber to access outsourced data and compromise the privacy of patients. The consent validator is responsible for governing access to computing resources in the public cloud (i.e., Smart CDSS services) and outsourced data persisted in public cloud storage service. The consent validator is further composed of a policy manager and a policy repository (see Fig. 18). The policy manager is responsible for evaluating access control policy on each data access request. This ensures that whenever a subscriber seeks access to the outsourced data, its access privileges are evaluated, and access is granted according to the requester's access rights to the outsourced data. Access control policies need to be evaluated within a trusted domain, so they are stored in a policy repository. This ensures that a system administrator can properly manage them, and if there is a change in any access control policy, it can be realized and enforced immediately.

4 A case study

Smart CDSS is a cloud-based service that accepts user input as activity, social interaction, and clinical information in standard HL7 vMR format and produce reminders, guidelines, and recommendations based on the stored information in the knowledge repository generated by clinicians and other domain experts. The ultimate consumer of Smart CDSS service can be a user (patient) monitoring system in a smart home, external EMR/EHR, PHR, social media or computerized physician order entry system (CPOE). The previous section discussed the main components of Smart CDSS service with information collection, flow, processing, and cloud infrastructure. To maintain the privacy of user data, some basic information of the user (patient) is anonymized before sending it for recommendations using the anonymization service. In this section, we present a detailed case study of our proposed system that illustrates the overall working of the different components and their integration and interaction with other components in the proposed system.

4.1 Use case: request for generating guidelines/reminders/alerts

Smart CDSS service accepts requests in standard HL7 vMR format. To adapt Smart CDSS to work with standard interfaces, consumers should transform their proprietary format into HL7 vMR format. The adaptability engine provides a space to all consumer applications to derive the corresponding adapter. Examples of such adapters include sensor adapters, social networks adapters, and interoperability engine adapters for standard clinical data. Moreover, to collect user (patient) data from different resources, the fusion adapter is designed to merge the input of diverse sources for a single user's (patient's) context into standard vMR format. In home care, an elderly person's data can be collected using the activity recognition system (i.e., using sensory data) and clinical information prescribed by a physician to monitor the patient's health condition. Figure 20 depicts the data capture procedure for a patient from three different sources: sensory data from activity recognition, clinical data from physicians, and patient records from social media like Twitter and PatientLikeMe.[2]

4.2 Request transformation to standard input: sensory input

To keep this example simple and understandable, we consider transforming patient sensory information into standard Smart CDSS input. Figure 21 shows some

[2] http://www.patientslikeme.com/

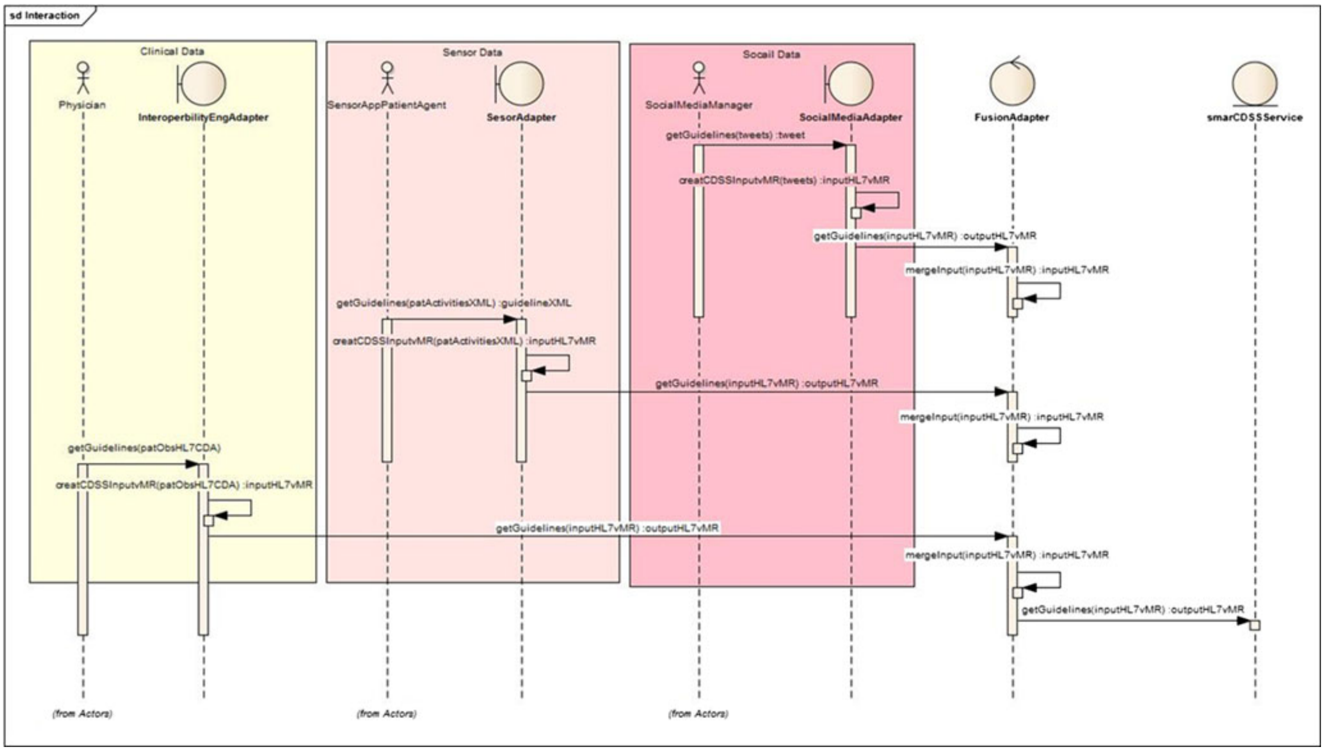

Fig. 20 Smart CDSS standards-based request from different data sources

partial information of patient activity in XML format. The corresponding standard input for Smart CDSS is shown in Fig. 22. The normal activity information has been transformed into standard format using the standard terminology of SNOMED CT.

Fig. 21 Patient activities log

```
<?xml version="1.0" encoding="UTF-8"?>
        <activities>
                <activity type="Motion">
                        <detectedBy>Motion Sensor</detectedBy>
                        <hasName>Mr J</hasName>
                        <activityName>Leaving Bedroom</activityName>
                        <id>1</id>
                        <time>2011:05:16:01:00:00</time>
                </activity>
                ........... ...........
                <activity type="Medicine">
                        <detectedBy>Wearable Sensor</detectedBy>
                        <hasName>Mr J</hasName>
                        <activityName>Taking Medication</activityName>
                        <id>5</id>
                        <time>2011:05:16:01:10:40</time>
                </activity>

                ........................
                </activities>
```

4.3 Request transformation to standard input: from clinical dataset

Clinical data of 100 patients were collected in unstructured format from Saint Mary's Hospital. The unstructured data

```
<vmrInput>
          <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
          <patient>
          ..........
                    <clinicalStatements>
                              <!-- current problems -->
                              <problems>
                                        <problem>
                                                  <id root="d7ebd80c-a28f-438f-9457-d3f92ea124ad"/>
                                                  <!-- Diabetes -->
                                                  <problemCodecodeSystem="2.16.840.1.113883.6.96"
                                                  codeSystemName="SNOMED CT" code="73211009"/>
                                        </problem>
                              </problems>

                              <!-- current medications -->
                              <substanceAdministrationEvents>
                                        <substanceAdministrationEvent>
                                                  <id root="54277620-9128-4c13-8fc8-623a38532627"/>
                                                  <substance>
                                                            <id root="2c803900-c8d1-457d-9567-4c92d75a0e23"/>
                                                            <!-- Morning after diabetic pill -->
                                                            <substanceCodecodeSystem="2.16.840.1.113883.6.96"
                                                            codeSystemName="SNOMED CT" code="102954005" />
                                                  </substance>
                                                  <documentationTime low="20110516" high="20110516"/>
                                        </substanceAdministrationEvent>
                              </substanceAdministrationEvents>
                    </clinicalStatements>
          </patient>
</vmrInput>
```

Fig. 22 Patient activities log in Smart CDSS HL7 vMR format

were converted into standard format by designing and implementing HL7 CDA (clinical document architecture) for patient encounters. The HL7 CDA template derived from the HL7 CDA specifications is used to represent patient encounters at Saint Mary's Hospital in clinical encounters for diabetes patient. As an example, the CDA document generated for one patient is shown in Fig. 23.

The HL7 CDA for this patient encounter at Saint Mary's Hospital is transformed into HL7 vMR format before forwarding it to Smart CDSS for decision support. The equivalent representation of the HL7 CDA document that is shown in Fig. 23 in HL7 vMR format is depicted in Fig. 24.

4.4 Use case: generating guidelines

MLM execution is a critical process of the Smart CDSS knowledge base. Based on the triggered events, knowledge base reasoning loads all corresponding MLMs and executes it for possible recommendations. The recommendations are sent back to the consumer of Smart CDSS in standard vMR format. Figure 25 depicts the sequence of interactions to generate guidelines or reminders.

5 Smart CDSS evaluation

Smart CDSS service has been evaluated for accuracy of recommendations generated based on an existing knowledge base, in the form of MLMs and performance of scheduled MLMs. For simple and meaningful recommendations, the knowledge base was populated with five basic MLMs that provided recommendations regarding hypertension and cholesterols levels, with some basic recommendations for diabetes management of these observations. The trial data

```
<ClinicalDocumentxmlns="urn:hl7-org:v3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" classCode="DOCCLIN" moodCode="EVN">
    <id root="2.16.840.1.113883.1122" extension="375913" displayable="false"/>
    ...........
    <component typeCode="COMP" contextConductionInd="true">
    <section classCode="DOCSECT" moodCode="EVN">
    <title representation="TXT" mediaType="text/plain">Objective</title>
    <entry typeCode="COMP" contextConductionInd="true">
    <observation classCode="OBS" moodCode="EVN">
    <code code="ObservationType" codeSystem="getCodeSystem"/>
    <entryRelationshiptypeCode="CAUS" contextConductionInd="true">
    <observation classCode="OBS" moodCode="EVN">
    <code code="271649006" displayName="Systolic BP" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"/>
    <effectiveTimexsi:type="IVL_TS">
    <low inclusive="true" value="20080220102200+0300"/>
    <high inclusive="true" value="20080220102200+0300"/>
    </effectiveTime>
    <value xsi:type="PQ" value="127" unit="mm[Hg]"/>
    </observation>
    </entryRelationship>
    ...........
    </entry>
    </section>
    </component>
    </structuredBody>
    </component>
</ClinicalDocument>>
```

Fig. 23 HL7 CDA for Saint Mary's Hospital patient encounter

```
<?xml version="1.0" encoding="UTF-8"?>
<in:cdsInputxmlns:in="org.opencds.vmr.v1_0.schema.cdsinput">
          <templateId root="2.16.840.1.113883.1122"/>
          <cdsContext>
                    <cdsSystemUserPreferredLanguage code="en" codeSystem="1.2.3" displayName="English"/>
          </cdsContext>
          <vmrInput>
                    <patient>
                              ............
                              <clinicalStatements>
                                        <!--lab results, using optional nested structure for panel-->
                                        <observationResults>
                                                  <observationResult>
                                                            <templateId root="2.16.840.1.113883.10.20.1.32"/>
                                                            <id root="107c2dc0-67a5-11db-bd13-0800200c9a66"/>
                                                            <observationFocus code="43789009" codeSystem="2.16.840.1.113883.6.96"
                                                            displayName="Diabetes Mellitus"/>
                                                            <observationEventTime low="20110803" high="20110803"/>
                                                            <observationValue/>
                                                            <relatedClinicalStatement><!--Blood Pressure-->
                                        <observationResult>
                                        <templateId root="2.16.840.1.113883.10.20.1.31"/>
                                        <id root="33d27880-eb74-11e0-9572-0800200c9a66"/>
                                        <observationFocusdisplayName=" Systolic BP  codeSystem="
                                         2.16.840.1.113883.6.96" code=" 271649006 "/>
                                        <observationEventTime low=" 20080220102200+0300 "
                                                                                high=" 20080220102200+0300 "/>
                                        <observationValue>
                                        <physicalQuantity value=" 127 " unit="mm[Hg] "/>
                                        </observationValue>
                                        </observationResult>
                                        </relatedClinicalStatement>
                              ............
                              </clinicalStatementRelationships>
                    </patient>
          </vmrInput>
</in:cdsInput>>
```

Fig. 24 HL7 vMR input for Smart CDSS using Saint Mary's Hospital patient encounter HL7 CDA

used as input was collected from clinical observations at Saint Mary's Hospital for 100 patients. According to the diagnosis history of the hospital, 20 patients had type-1 diabetes, 40 had type-2 diabetes, and 40 were examined for diabetes but found to have no evidence of the disease. During invocation of Smart CDSS for this diabetes data for 100 patients, the system was provided with each patient's encounter information one by one. The invocation caused the scheduling of two basic MLMs to trigger "evidence of diabetes" based on cholesterol level and glucose level.

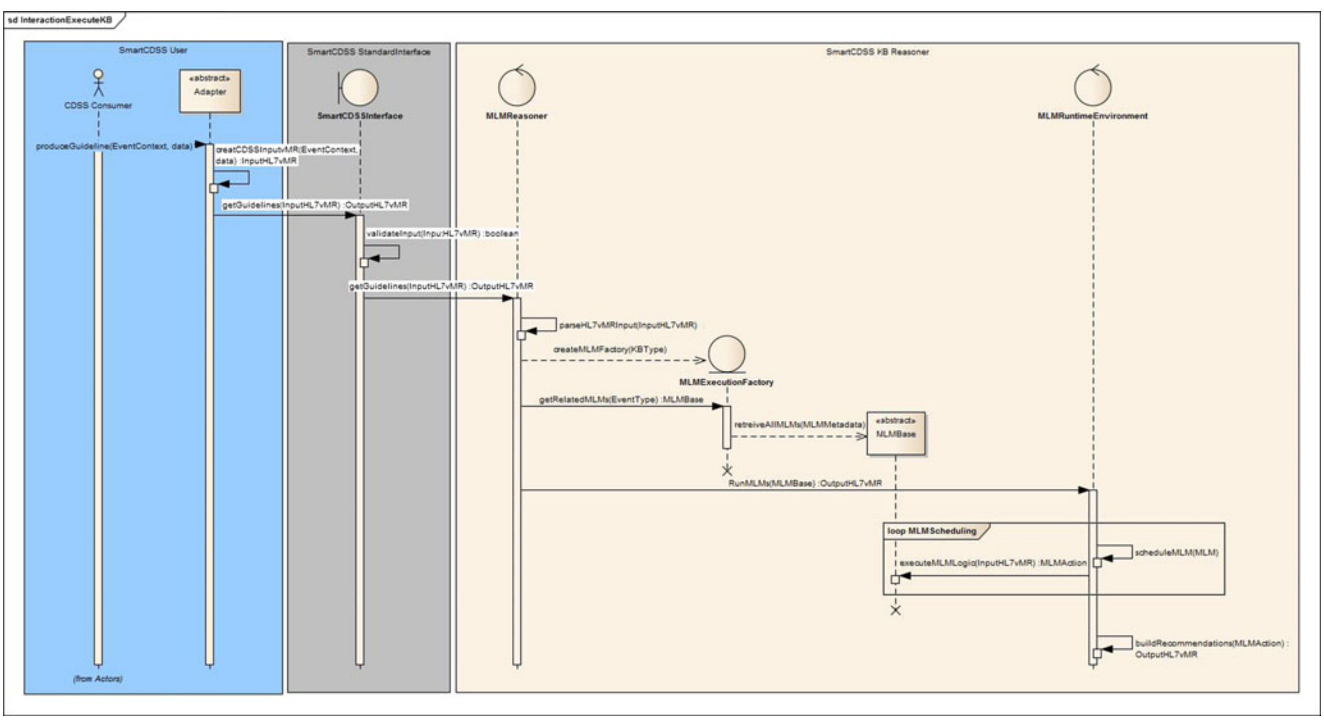

Fig. 25 Smart CDSS recommendation

Smart CDSS identified three patients as suspicious for type-1 diabetes, who had not been diagnosed with diabetes. Moreover, two patients diagnosed with type-1 diabetes were placed in the type-2 category. Investigating this incorrect placement revealed that a physician had found the patient to have comorbidities that were not included in the existing MLMs. This limitation discovery led to the conclusion that the MLMs should be restructured to handle the results for other diseases. The outcome of these results was to enrich the knowledge base by including more possible trends of symptoms for diagnosing diabetes.

Additionally, in the case examined, the system provided useful general recommendations, such as exercise, diet management, and other diabetes education, that complemented the physician guidelines. For general recommendations, the system was evaluated for one patient who had made 11 physician visits over a two-year span of diabetes history.

For performance of the MLM execution, it was demonstrated that, for scheduling two MLMs to be fired for a particular event, it takes less than half a second to produce all the related recommendations. So the system expecting ten MLMs to be scheduled at a time would take about 3 s, which is very acceptable for chronic disease recommendations.

6 Conclusions and future directions

Building a health-care system with low cost and better services in which the information is coming from diverse modalities is a collaborative and challenging issue. In this research paper, we describe the design and development of a Smart Clinical Decision Support System (Smart CDSS) that provides standards-based health-care services. To achieve the desired objectives, we collected user activity data using different sensors, user experience information based on social media data, and user medical history from clinical records. In addition, medical expert knowledge was used to infer the user/patients' situations and generate appropriate recommendations for them. All the information is handled in HL7 standard format, and the guidelines and recommendations have been strictly reviewed by medical doctors and domain experts. A secure cloud-based working case study of the proposed system is provided that demonstrates the overall working of the proposed Smart CDSS.

In future research, we plan to incorporate online published medical information into the proposed system. The knowledge base will evolve with extracted guidelines from online resources. We are also developing an authoring tool for the domain expert so that they can build new guidelines with an easy-to-use user interface and so they can verify the automatically generated guidelines.

Acknowledgments This work was supported by a National Research Foundation of Korea (NRF) grant that was funded by the Korean government (MEST) (No. 2011-0030823).

This research was also supported by the MKE (Ministry of Knowledge Economy), Korea, which is under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-(H0301-12-1004).

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

References

1. Blumenthal D, Tavenner M. The "meaningful use" regulation for electronic health records. N Engl J Med. 2010;363(6):501–4.
2. Evidence on the costs and benefits of health information technology; 2008. Publication No. 2976. http://www.cbo.gov/sites/default/files/cbofiles/ftpdocs/91xx/doc9168/05-20-healthit.pdf, Accessed: September 23, 2012.
3. Kern LM, Wilcox A, Shapiro J, Dhopeshwarkar RV, Kaushal R. Which components of health information technology will drive financial value? Am J Manage Care. 2012;18(8):438–48.
4. Wright A, Sittig DF, Ash JS, Sharma S, Pang JE, Middleton B. Clinical decision support capabilities of commercially-available clinical information systems. J Am Med Inform Assoc. 2009;16(5):437–644.
5. Wright A, Sittig DF. A four-phase model of the evolution of clinical decision support architectures. Int J Med Inform. 2008;77(10):641–9.
6. Sittig DF, Wright A, Osheroff JA, Middleton B, Teich JM, Ash JS, et al. Grand challenges in clinical decision support. J Biomed Inform. 2008;41(2):387–92.
7. HL7 Organization Home. http://www.hl7.org/, Accessed: September 24, 2012.
8. Kawamoto K, et al. Multi-national, multi-institutional analysis of clinical decision support data needs to inform development of the HL7 virtual medical record standard. AMIA Annual Symposium Proceedings, 2010, pp. 377–82.
9. Samwald M, De Fehre K, Bruin J, Adlassnig KP. The Arden syntax standard for clinical decision support: experiences and directions. J Biomed Inform. 2012. doi:10.1016/j.jbi.2012.02.001.
10. Sordo M, Boxwala AA, Ogunyemi O, Greenes RA. Description and status update on GELLO: A proposed standardized object-oriented expression language for clinical decision support. Medinfo 2004: Proceedings of the 11th World Congress on Medical Informatics, pp 164–9, 2004.
11. Khattak AM, Truc PTH, Hung LX, Vinh LT, Dang V, Guan D, et al. Towards smart homes using low level sensory data. Sensors 2011;11(12):11581–604.
12. Microsoft Azure. http://www.windowsazure.com/en-us/. Accessed: September, 2012.
13. Khattak AM, Pervez Z, Lee SY, Lee YK. Intelligent healthcare service provisioning using ontology with low level sensory data. Trans Internet Inf Syst (TIIS). 2011;5(11):2016–34.
14. Pervez Z, Khattak AM, Lee SY, Lee YK, Huh EN. Oblivious access control policies for cloud based data sharing systems. Springer, 2012.
15. Gietzelt M, Goltz U, Grunwald D, Lochau M, Marschollek M, Song B, et al. Arden2ByteCode: a one-pass Arden syntax compiler for service-oriented decision support systems based on the OSGi platform. Comput Methods Prog Biomed. 2011.
16. Karadimas HC, Chailloleau C, Hemery F, Simonnet J, Lepage E. Arden/J: an architecture for MLM execution on the Java platform. J Am Med Inform Assoc. 2002;9(4):359–68.

17. Hung LX, Lee SY, Truc PTH, Vinh LT, Khattak AM, Han MH, et al. Secured WSN-integrated cloud computing for u-life care. 7th Annual IEEE CCNC 2010 Conference http://www.ieee-ccnc.org/2010/, Las Vegas, USA, January 9–12, 2010.
18. Fahim M, Fatima I, Lee S, Lee YK. EEM: evolutionary ensembles model for activity recognition in smart homes. Int J Appl Intell, pp 1–11, 2012.
19. Wang F, Turner KJ. An ontology-based actuator discovery and invocation framework in home care systems. In: Proceedings of the7th International Conference on Smart Homes and Health Telematics, pp. 66–73, Berlin, June 2009.
20. Pollack ME, Brown LE, Colbry D, McCarthy CE, Orosz C, Peintner B, et al. Autominder: an intelligent cognitive orthotic system for people with memory impairment. Robot Auton Syst 2003;273–82.
21. Sohn T, Li K, Lee G, Smith I, Scott J, Griswold W. Place-its: A study of location-based reminders on mobile phones. In: Ubicomp, pp. 232–50, 2005.
22. Du K, Zhang D, Zhou X, Mokhtari M, Hariz M, Qin W. HYCARE: A hybrid context-aware reminding framework for elders with mild dementia, ICOST, 2008.
23. Chen L, Nugent C, Wang H. A knowledge-driven approach to activity recognition in smart homes. IEEE Trans Knowl Data Eng. 2011;24(6):961–74.
24. Fischhoff IR, Sundaresan SR, Cordingley J, Larkin HM, Sellier MJ, Rubenstein DI. Social relationships and reproductive state influence leadership roles in movements of Plains zebra, Equusburchellii. Anim Behav. 2007;73(5):825–31.
25. Collins JJ. The importance of medication reminders. http://ezinearticles.com/?The-Importance-of-Medication-Reminders&id=5129852.
26. Press Release. Text message prescription reminders significantly improve patient adherence to oral diabetes medication, May 24, 2012. http://www.marketwatch.com/story/text-message-prescription-reminders-significantly-improve-patient-adherence-to-oral-diabetes-medication-2012-05-24.
27. KEA. http://www.nzdl.org/Kea/, Accessed: April 2011.
28. Khan WA, Khattak AM, Lee SY, Hussain M, Amin B, Latif K. Achieving interoperability among healthcare standards: building semantic mappings at models level. In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ACM 2012.