

Ontology Based Framework for Interactive Self-Assessment of e-Health Applications

Wasin PASSORNPAKORN^{†a)}, *Nonmember* and Sinchai KAMOLPHIWONG^{†b)}, *Member*

SUMMARY Personal e-healthcare service is growing significantly. A large number of personal e-health measuring and monitoring devices are now in the market. However, to achieve better health outcome, various devices or services need to work together. This coordination among services remains challenge, due to their variations and complexities. To address this issue, we have proposed an ontology-based framework for interactive self-assessment of RESTful e-health services. Unlike existing e-health service frameworks where they had tightly coupling between services, as well as their data schemas were difficult to change and extend in the future. In our work, the loosely coupling among services and flexibility of each service are achieved through the design and implementation based on HYDRA vocabulary and REST principles. We have implemented clinical knowledge through the combination of OWL-DL and SPARQL rules. All of these services evolve independently; their interfaces are based on REST principles, especially HATEOAS constraints. We have demonstrated how to apply our framework for interactive self-assessment in e-health applications. We have shown that it allows the medical knowledge to drive the system workflow according to the event-driven principles. New data schema can be maintained during run-time. This is the essential feature to support arriving of IoT (Internet of Things) based medical devices, which have their own data schema and evolve overtime.

key words: *ontology-based knowledge model, ontology-based health information, ontology-based application programming interface, RESTful service composition, HATEOAS*

1. Introduction

Decision Support System (DSS) [1] has been developed over four decades to address a huge amount of information issues. In healthcare, DSS is the computer-based set of tools, which provides clinical knowledge and patient-related information to clinicians at appropriate time [2]. DSS has been applied in various fields of healthcare domain such as to improve diagnosis result [3], to visualize the diagnosis path and their criteria [4], and to validate drug formation [5]. These DSSs can be categorized by their objectives in clinical-oriented, and patient-oriented.

In clinical-oriented, DSS is commonly known as Clinical Decision Support System (CDSS) [6], which aims to address problems knowledge in the form of Clinical Intervention Guideline (CIG) [7]. There are various types of CIGs reviewed/discussed [8]. Most of these representations were based on ontology-based models, which aimed to represent

control-flow for clinical tasks in a form of hierarchy task network. These ontology-based models addressed interoperability problems, which were commonly found in clinical domain. These models covered various expression logic's requirements of medical knowledge representation such as intension-base task (goal-base), temporal reasoning (e.g., overlap, meet, and trend of data). Some efforts attempted to develop their representation languages, or to address some existing problems of CIGs for solving their specific problems in clinical domain [9]–[13].

In contrast with CDSS, patient-oriented CDSS aims to deal with patient's self-care process. Since patients have different requirements from physicians [14]. These requirements need a system to treat patients as a primary user by tracking patient's current health statuses and delivering the timely recommendations. For example, medical devices inside a small briefcase [15] were used to diagnose, categorize, and deliver health risk of patients to telemedicine call center by using mobile networks. Beyond healthcare tracking system [16] was able to collect lifestyle events and converted them to exercised suggestions according to their provided training plan. A concept of intelligent PHR (Personal Health Record: it is an electronic health record for each individual use purpose) [17] has been proposed, which focused on monitoring and triggering mechanism such as data collecting, data preprocessing, abnormal event detection, and linking the result to related medical devices. MONARCA 2.0 system [18] aimed to improve the health awareness of patients to disease insight. Their system estimated the trend of mood state for allowing physician to provide proactive and prevention treatment. StudentLife system [19] was proposed to assess the relationship between stress and life-style activities, while GRiST [20] tried to assess patient's mental health. Their system delivered expertise decision to specific patient group through different assessment setting. Embedding clinical guideline into patient-oriented CDSS [21] was developed for allowing clinical nurse to evaluate patient status and deliver healthcare intervention to patient.

These CDSSs addressed their primary goals such as integration with clinical workflow and support self-care process. However, these CDSSs did not focus on system dynamic environments such as flexibility, scalability, adaptability, and evolution of the system. Their systems lacked of the handling and interfacing capability for other software interactions. As a result, difficulty on communicating with external software modules will be faced.

To address these issues, we propose the ontology-based

Manuscript received June 3, 2015.

Manuscript revised September 11, 2015.

Manuscript publicized October 21, 2015.

[†]The authors are with the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand.

a) E-mail: 5410120047@psu.ac.th

b) E-mail: sinchai.k@psu.ac.th (Corresponding author)

DOI: 10.1587/transinf.2015NTI0001

framework for interactive healthcare self-assessment. Our framework allows medical knowledge and patient's context to drive various services to achieve the agreement goals. REST design pattern [22] is adopted to address the flexibility and evolution issue of services. However, allowing service working together in RESTful manner still has some big open challenges [23] such as self-description, composition modelling, and resource state management. We address these issues by adopting hypermedia driven interface and linked data protocol such as A Vocabulary for Hypermedia-Driven Web APIs (HYDRA) [24], and Linked Data Platform (LDP) [25].

The rest of this paper is organized as follows: First, we investigate related works in next section. In Sect. 3, we overview our proposed system. In Sect. 4, we describe our proposed domain data model. Next, in Sect. 5, we present the mapping model between our proposed domain model with HYDRA, and LDP. In Sect. 6, we demonstrate our test-case scenarios that allow various services working together, samples of e-health scenarios are given. In Sect. 7, some main benefits of our work will be discussed. Then, we summarize our paper in the last section.

2. Related Works

2.1 Service-Oriented Clinical Decision Support System

SOA (Service-oriented Architecture) enables various benefits in software development and maintenance [26]. However, there are still few implementations of SOA in CDSS. Therefore, various efforts establish the standard and guideline for developing the CDSS in SOA manner. For example, HL7 working group identified basic component services of service-oriented CDSS [27], and drafted the required information and functionality for decision support service [28]. Health eDecisions (HeD) [29] defined interaction model for stateless interaction between CDSS guidance requester and CDSS guidance supplier.

According to the above efforts, some works adopted these standards and guidelines. For example, MobiGuide architecture [30] provided ubiquitous functionality for patient guidance system. They adopted SOA to facilitate independent software development. They addressed some issues in the standard [27] through their PHR mapping service.

Some efforts applied existing service technologies in healthcare domain. For example, Weitzel [31] developed a medical protocol as a service. Their system allowed services to adapt themselves to meet the requirements of healthcare experts. Their system adopted OpenSocial API [32] with HL7 standard format [33]. This adaptation provided channels for accessing patient's information through standard management interfaces (i.e., OpenSocial API). Ontology-based information and knowledge management framework for chronic disease care management [34] was proposed as a reminding and alerting system. Their system was based on SOA, which provided the data schema editor, and rule-editing interface. Aside from traditional service-oriented,

the extension framework of European Telecommunications Standards Institute (ETSI) was presented in [35]. Their proposed framework relied on ETSI for resources gathering process. They extend ETSI with reasoning mechanism, adaptation, event-handler, and patient's profile management.

Some efforts focused on CDSS context-aware system. Their system could adapt the system behavior according to input contexts. For example, Paganelli [36] proposed ontology-based context aware for supporting home-based continuing care with various serviced supports such as information sharing, multichannel information accessing, patient health monitoring, and alarm handling. Ongenae [37] proposed automatic knowledge adaptable behaviors of context-aware applications for reducing the user adaptation. From aforementioned SOA-based system in healthcare, we found that most of them focused on features of systems. Most of them did not consider the issues about REST principles [22], especially Hypermedia as the Engine of Application State (HATEOAS) concept. The HATEOAS together with other REST constraints can reduce out-of-band information and coupling between services.

2.2 Restful Service Composition

Usually, SOA is comprised of several services working together. Each service may re-use other services for archiving certain goals [38]. To allow various services working together, a predefine agreement needs to be defined. As a result, these services will be difficult to change in the future.

REST constraints [22] alleviate these problems, since it requires the service to provide necessary information for processing each request through link or reference to related information. These constraints allow the service to change or increase their features without breaking the existing clients during the run-time period, since clients can claw from existing resource to their interested resource. However, REST cannot remove all out-of-band information. Some agreements need to be defined on hard coding such as the location of operations and their expected inputs/outputs. These agreements can be called as media-type of resource. To reduce the coupling between client and services, the media type should not be changed frequently. Therefore the standardize media type is more suitable than customized media-type. In addition, the media type should fit to hypermedia control for allowing the service to work together automatically. That is, it must be able to present the functional operation of each resource. Therefore, we extend the design in our previous work [39] with HYDRA and LDP.

HYDRA [24] is the set of vocabulary for defining media-type of hypermedia service interface. HYDRA focuses on functionality description of resources such as operations, hint for their expected inputs and possible outputs. These description describe through `hydra:operation`, `hydra:expect`, and `hydra:return` respectively. These descriptions allow service to chain together by mapping between service outputs and other service inputs. LDP [25] is W3C

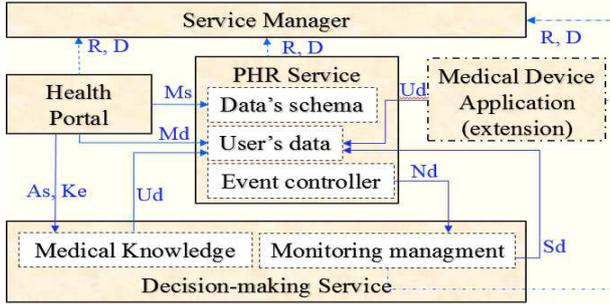


Fig. 1 Overview of proposed system

recommendation that defines the set of rules and vocabulary for management of ontology-based information through HTTP protocol. The LDP will be the most importance part for management of the medical knowledge. Since, the ontology-based information is often used as the encoded format of medical knowledge. This combination allows ontology-based medical knowledge to drive the service composition process.

3. Ontology Based Interactive Self-Assessment Service Model for e-Health Services

The patient needs are often diverse, uncertain, personalized, and dynamic. To satisfy these requirements, the framework should be atomic, extensible, and flexible enough. Therefore, the components in this system are designed as RESTful service. As shown in Fig. 1, our proposed system consists of service manager, health portal, decision-making service (DM), personal health record service (PHR), and external services (e.g. medical device applications).

From Fig. 1, firstly, all of the services need to register (R) their root URLs to the service manager. This process allows various services in our framework to discover their interested resources at run-time (D). To illustrate the functionality of each service, we construct health portal for allowing human user to manage resources in each service. The health portal allows user to manage their health data (Md) and data type (Ms) through PHR service's APIs. The health portal also provides assessment interface (As) and medical knowledge editor (Ke) by connecting to medical knowledge interface of DM. During decision process, the DM may need to subscribe necessary data (Sd) at PHR service. When the data is available, the PHR service will send notification to the DM for their decision process. The result of DM service is updated to the user's data storage (Ud) at PHR service.

Our proposed framework allows an interoperability with Internet of Things (IoT) based medical monitoring devices, for example blood pressure, heart rate, weighing scale, or glucose meter. Data of these devices can be flowed to the DM for analyzing and updating a new result according to related algorithms and current patient states.

4. Domain Data Model

This section describes our domain data model, which con-

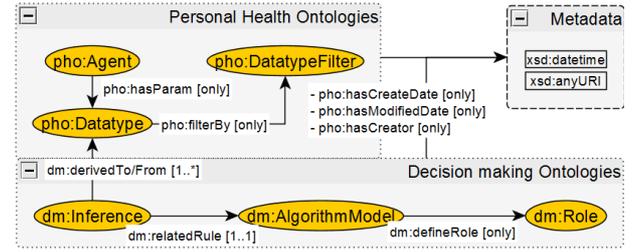


Fig. 2 Top-level model of proposed data model

sists of personal health ontologies, and decision-making ontologies. In addition from Fig. 2, both classes and their instances can have their metadata through various annotation properties such as pho:hasCreateDate, pho:hasModifiedDate, and pho:hasCreator.

4.1 Personal Health Data Model

The personal health ontologies (PHO) are a representation of patient's data. These ontologies allow various services to understand the meaning and constrains of each data type in the consistency way. They consist of two main classes such as pho:Agent, and pho:Datatype. The instances of these classes are used for representation of user's instance, and user's parameter type, respectively. The instance of pho:Agent can link with instance of pho:Datatype through object property named, pho:hasParam.

4.2 Decision-Making Data Model

Decision-making model has its own data types that relate to medical decision such as event, condition, and its enumeration value. These data types are the extension part of pho:Datatype from Fig. 2. Aside from pho:Datatype extension, the decision model consists of dm:Inference, dm:Role, and dm:AlgorithmModel. The dm:Inference is responsible to link the input, output, and decision node together. The dm:Role is used to represent rule's characteristic. In addition, it is used to map with algorithm implementation such as SPARQL rule. The dm:AlgorithmModel contains properties of algorithm structure (e.g., sub/sibling algorithm) and their metadata. These three classes describe functionality facet of algorithm, which allow decision-making process to map with service's resource.

There are various subclasses of dm:Role such as dm:IfStmt, dm:ThenStmt, dm:Derive, dm:Assert, dm:StartTimeMax, and dm:AtLeastN operators. These subclasses are used to represent different characteristics of rules, for example, dm:Role structure is described in Fig. 3. From dm:MatchStmt Role in Fig. 3, it is used to map the instance of dm:AlgorithmModel with SPARQL simple statement such as subject, predicate, object, filter comparator, and union. The dm:MatchStmt has two subclasses: dm:IfStmt, and dm:ThenStmt. These two subclasses are used to translate instance into statement and construct statement of SPARQL, respectively. Each instances

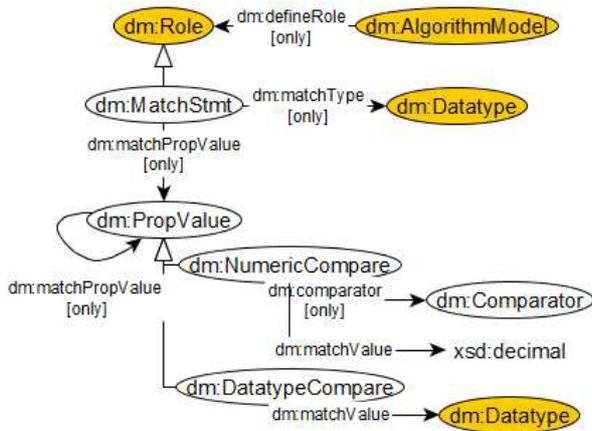


Fig. 3 Example of Role: MatchStmt descriptions

of `dm:MatchStmt` has `dm:matchType` property, which represents data type of subject. The predicate and object are represented through `dm:matchPropValue` that links to `dm:PropValue` class. The `dm:PropValue` class can be linked to other instances of `dm:PropValue`. The `dm:PropValue` class has two subclasses: `dm:NumericalCompare`, and `dm:DatatypeCompare`, which represent criterion filtering for object numerical value and object data type value respectively. There are other example of `dm:Role`. For example, `dm:Derive` and `dm:Assert` are used for updating the existing data. The `dm:StartTimeMax` are used to specify location of input's time metadata, which allows the system to generate time metadata according to input's data.

The instance of `dm:AlgorithmModel` will be used for generating SPARQL rule. The generating rule then will be embedded into decision-making model as the spin rule according to SPIN (SPARQL Inferencing Notation) vocabulary [40] (i.e., `spin:rule`). These spin rule will be used for reasoning process. After instance of `dm:AlgorithmModel` is defined, the instance of inference classes can link the related rule with their inputs and their outputs together. Both inputs and outputs are the instance of `pho:Datatype`, which are generated from property `dm:matchType` of `dm:MatchStmt` instance. These relations allow the initialization of gathering process, which describe at Sect. 6.2.

5. Hypermedia Interface Model

To allow various services working together with minimum out-of-band information, we adopt Hypermedia-Driven Web APIs (HYDRA) vocabulary and Linked Data Protocol (LDP). The HYDRA is the vocabulary, which is adopted to present the supported operation and their inputs/outputs to client. On the other hand, the operation type that will be presented is based on LDP.

We design our APIs by adopting `hydra:Collection` as the entry point for management of our domain resource. Each instance of `hydra:Collection` supports two main operations: view and create operations through `hydra:filter`, and `hydra:operation/hydra:method="POST"` respectively. The

Table 1 Example of resource and its constraints.

Resource	POST Constraints
<code>phoapi:DatatypeCollection</code>	[OR]* PrimitiveDT, ComplexDT, EnumDT, QuantityDT
<code>phoapi:UserParameters</code>	<code>property = rdf:type</code> <code>allowedValue** = phoapi:Datatype-Collection</code>
<code>phoapi:Triggerings</code>	[OR]* OnInstanceCreate, OnInstanceUpdate, CustomNotification
<code>dmapi:AlgorithmCollection</code>	<code>property = rdf:type</code> <code>allowedValue** = dmapi:Algorithm</code>
<code>dmapi:Algorithm</code>	<code>property = rdf:type</code> <code>allowedValue** = phoapi:Datatype-Collection</code>

*[OR]: the correct resource must has its constraint match with one of these constraints. Each constraint may contains `valueType`, `allowedValue`, `parent`, and `hasValue` property.

**`allowedValue`: the property's value must be one member of the collection. The collection is the value of `allowedValue`. The member of collection describes through `hydra:member` property.

view operation allows client to retrieve the resource according to some URL templates. The create operation allows client to perform the POST operation with input's constraints described as `hydra:expect`. In addition, since HYDRA vocabulary does not provide enough information for constraint definition. Therefore, we extend HYDRA vocabulary with the constraints concept similar to developing vocabulary named `shape` [41]. Our extended vocabulary is the property of `hydra:SupportedProperty`, which can be placed on any classes to describe constraints of class's instance. Usually, they can be used together with the `hydra:expect` to guide the client about the constrains of newly create resource. This vocabulary consists of `valuetype`, `allowedValue`, `parent`, and `hasValue`. The `valuetype` references to `xsd:datatype` or resource that has property `hydra:supportedProperty`. The `allowedValue` references to `hydra:Collection`. The `parent` property describes the nested properties, which is one of the issues in HYDRA community. Lastly, the `hasValue` describes the property and its value that must be presented.

We adopt `hydra:Collection` for management of the following domain data: 1) `pho:Datatype` and its subclass managed through `phoapi:DatatypeCollection`. 2) The instance of `pho:Agent` with `pho:hasParam` property managed through `phoapi:UserParameters`. 3) The instance of `pho:Agent` with `pho:hasTriggering` property managed through `phoapi:Trigerrings` 4) `dm:AlgorithmModel` and its subclass managed through `dmapi:AlgorithmCollection`. Aside from collection management, we provide the `dmapi:Algorithm` for initialization of decision-making process. This resource allows client to enter health's data through POST operation. More detail descriptions of resource's constraints are shown in Table 1.

In addition to resource's constraints shown in Table 1, some resources contain `hydra:supportedOperation`, which is used to specify `hydra:operation` on instance that has type as the current resource. For example, the

phoapi:UserParameters has hydra:supportedOperation for PUT and DELETE operations. Therefore, all instances of phoapi:UserParameters (i.e., patient's health parameters) will have PUT and DELETE operations for allowing patients to manage their existing data. This property also describes on the dmapi:Algorithm for management of medical algorithm.

6. Test Scenarios

6.1 The Management of PHR Schema and Decision-making Knowledge

Both construction processes of PHR schema and decision-making knowledge at the client side are similar. Since, health portal acts as a generic client that performs the operation according to HYDRA and LDP vocabulary. However, there are some differences in server side process. In the PHR schema management, a new schema needs to be validated with the schema constraints that define at PrimitiveDT, ComplexDT, EnumDT, and QuantityDT. In the medical knowledge management, there are two needed mechanisms to perform such as filtering and classification mechanisms. The filtering is used for dealing with duplicated or similar rules. The classification is used for medical knowledge management, which consists of grouping, merging, and hierarchy. These mechanisms perform through ontology-based reasoning module such as OWL-RL of SPIN library or OWL-DL classification of Pellet.

6.2 Data Monitoring and Triggering Process

From Fig.4, DM process can be started by notification triggering rules in PHR (1.1), or by patient entering abnormal symptom (1.2). In the first case, firstly, DM needs to subscribe the interested events to PHO service. When the matching event occurs, the notification will be send to DM. In the latter case, patient needs to POST his/her symptoms according to hydra:expect property's value of dmapi:Algorithm. After receiving the input, e.g. pho:VomitOccur, the service will create the user model (e.g. instance of pho:Agent) according to the received input, and will send them to analyzing module. This module is the

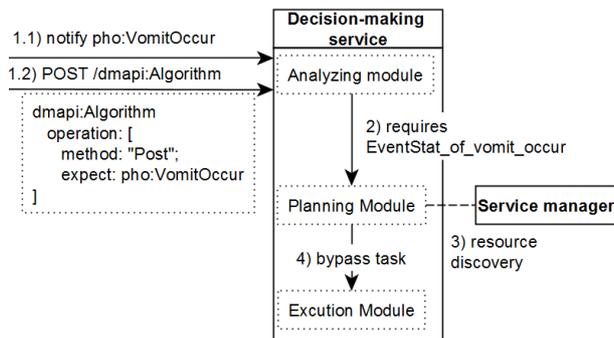


Fig. 4 Sample use-case of data monitoring and triggering process

rule-engine, which consists of several SPARQL based rules that are able to produce the set of dm:Inference's instances. These instances contain some numbers of related inputs, outputs, and rules. According to dm:AlgorithmModel described in Sect.4.2, the input and output of each algorithm can be presented. Therefore, the planning module can find the missing data of each algorithm by matching algorithm descriptions to the existing user's data model. These missing data are the resources that need to gather for further analyzing processes. In this example, the missing data is EventStat_of_vomit_occure (2). This resource is the summary information of pho:VomitOccur event that requires system to monitor pho:VomitOccur event overtime. However, it does not contain in the current user's model. Therefore, the analyzing module will mark on them and send them to the planning module. The planning module is responsible for location discovery of missing data (3). This discovery process can be achieved through resource discovery of service manager. The service manager is able to discover certain resource by matching the constraints of EventStat_of_vomit_occured with constraints of hydra:return in operation of each registered service APIs. These constraints are specified by HYDRA vocabulary and constraints extension, which describe in Sect.5. Assume that the constraints of EventStat_of_vomit_occured consists of property = pho:eventFreqOf, and valueType = pho:VomitOccur. Therefore, the hydra:return value of matching service must have pho:eventFreqOf as the property's value; and its valueType must be pho:VomitOccur or super class of pho:VomitOccur (e.g., pho:Event). In this case, the service manager finds EventStat resource on one operation of Event Stat service (ESS) match with EventStat_of_vomit_occured constraints. Therefore, the planning module will pair URL of ESS's resource with its related operation, and then sends it to the execution module for running the gathering process (4).

From Fig. 5, DM starts a gathering process by performing POST on the ESS with necessary data described in the

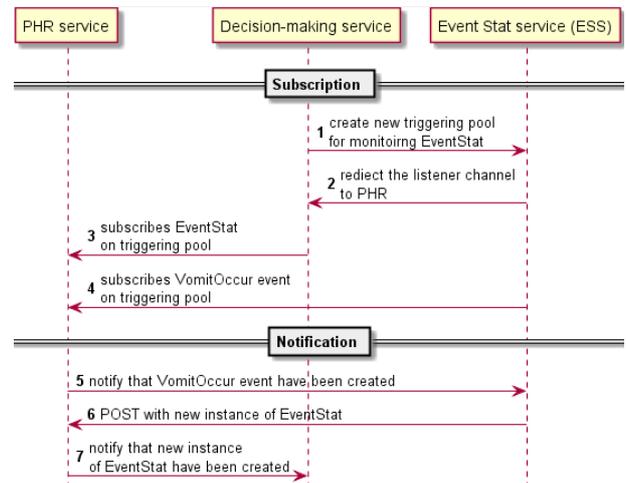


Fig. 5 Event Subscription and Event notification

constraints value of `hydra:expect` such as the type of input data, (e.g., `EventStat_of_vomit_occur`). The ESS will return the listener URL to DM, e.g., triggering pool of `EventStat` at PHR. Therefore, DM will subscribe on such triggering pool URL. The process of ESS then starts by subscribing the `pho:VomitOccur` at PHR service. When the `pho:VomitOccur` is updated, the PHR will notify ESS which allows ESS to calculate the `EventStat` of `pho:VomiOccur` and updates them on PHR. When the `EventStat` is updated, DM will receive the notification, which allows DM to resume its analyzing process.

7. Discussion

In the Personal e-Health service, the schema of data can be extended and changed overtime. The proposed schema covers various types of information such as patient health information, clinical knowledge, and event matching. Unlike data types of exiting personal health records (e.g. SMART platform [42], Indivo [43], FHIR [44]) where they are unchangeable and difficult to extend. Moreover, each data type can have their own metadata such as author, date, and label. New data types may inherit constraints from existing data types. These features achieve through the implementation of `hydra:supportedProperty` in their classes, which allow them to specify data constrains of their instances.

In our decision-making module, we have implemented clinical knowledge through the combination of OWL-DL and SPARQL rules. SPARQL rule has some advantages over SWRL [45] and Jena built-in rule [46]. For example, SPARQL's rule syntaxes allow new instances and metadata (e.g. author, time stamp) to generate, and can be embedded into OWL class through SPIN vocabulary.

However, SPARQL-based rule is difficult to maintain because of their flexibility characteristic. The failure of rule construction may cause the reasoning process running infinitely. Therefore, we provide SPARQL rule templates (e.g., `dm:AlgorithmModel`) and management interface (e.g., `dmapi:AlgorithmCollection`). The rule templates allow the algorithm to be traced and allow physician or patient to ensure the correction of their decision results. In addition, the rule template can be used for explaining the missing resources that need to be gathered for further analysis. This approach is similar to SPARQL template in [47]. However, our approach attempts to describe the functionality facet of algorithm and apply into generic use-cases. The rule management interface is ease for developing of knowledge management by providing supported constraints that can be defined in each rule.

After the knowledge has been provided, the suggestion can be sent to patient. The suggestion and action result come from various services working together. For example, Alice is a pregnant woman with vomit symptom employs our framework for dealing with her symptom consulting. This initial condition may require various processes such as to estimate the vomiting severity, to provide the suggestion according to diagnosis result, to follow-up the symptom and

related conditions, and to provide the map and transportation of a hospital if needed. In order to allow each component to evolve overtime, these components can be deployed as a separated service in our framework.

In our proposed framework, these scenarios achieved through various services working together such as the service manager that registers available resources of each service, the event-subscription service that monitors the data, and the decision-making service that executes each service according to medical knowledge and service requirements. In order to let these services working together, each service must expose the functionality of its resource. This issue can be achieved through HYDRA vocabulary such as `hydra:operation`, `hydra:expect`, and `hydra:return`. For example, a patient enters vomit symptom into the DM. The DM then needs to run its analyzing process, which requires more data to run its algorithm. These required data are the comparing result between `dm:Algorithm` and current user's data repository. For gathering data, the DM needs to find another service's operations which have `hydra:return` matching to such data. When service is found, the DM needs to invoke the service's operation by providing necessary data according to `hydra:expect`. To provide such necessary data, the DM needs to invoke another service. This invoking loop allows the DM to chain various services together according to initial required data of the DM.

After the services are able to working together, these services may be changed or evolved in the future. For example, a developer of ESS decides to change their services according to Separation of Concern Principle (SPC). That is, instead of letting ESS queries the necessary input from patient's URL, the client of ESS must provide all necessary information to ESS. Therefore, ESS changes its interface from two input parameters to one input parameter. This input parameter is the URL referred to collection of target event, e.g., `pho:VomitOccur` collection. This change will break the coordination process between ESS and DM. Since, hard coding of DM is used to interface with ESS, therefore, every modification of each service requires code changing. This issue is hard to maintain if the system consists of various service working together.

To address this issue, we design and develop our framework according to REST. The importance constraints of REST that address this issue is self-descriptive and HATEOAS. These two constraints require service to deliver the message with necessary information for performing the process according to such message. Therefore, the client can perform the recursive retrieval operation from current resource, until the desired resource is found. For example, when DM needs to gather `EventStat_of_vomit_occur` resource, which is the resource of Event Stat service (ESS). DM requires service manager to discovery this resource from its registered service. The discovery process in non-HATEOAS approach requires fixed information at the service manager such as resource locations and their supported operations. If the ESS is changed, the ESS needs to inform the service manager to change the registered information.

On the other hand, in HATEOAS approach, this mechanism can be avoided. Since, the service manager requires only the root URL of the registered services, i.e., ESS root URL. The service manager then can perform recursive retrieval operation (e.g., GET) from root URL, until it finds Event-Stat.of.vomit_occurred. In this case, the service manager does not need to fix any information in its implementation, except the root URL of each service. This approach allows a service to change its implementation overtime, as long as the semantic of resources has not changed. The limitation of this approach is it requires client to support the clawing process. However, this clawing process does not need to change overtime. Since, the clawing process is based on the standardize media-type (e.g., HYDRA and LDP) of resource that are not often changed.

8. Conclusions

This paper presents a design ontology-based framework for interactive self-assessment of e-Health services. Our work focuses on the flexibility of service interface that allows coordination among various services to achieve interactive self-assessment results. This flexibility achieves through the design and implementation based on HYDRA, LDP, and REST principles. The proposed system bridges the gap between the medical knowledge and medical management service interfacing. We have demonstrated how to apply our framework for self-assessment in e-health applications. Furthermore, our framework allows the medical knowledge to drive the system workflow according to event-driven principles. In addition, our framework considers an integration process of various services in IoT (Internet of Things), while allows them independently involve.

Acknowledgments

This work was supported by the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission (under the funding no. MED540548S at Prince of Songkla University). Thanks to Prof. Veerapol Chandeying, Faculty of Medicine, University of Phayao, for his valuable advising on medical knowledge to this project.

References

- [1] K. Vizecky and O. El-Gayar, "Increasing research relevance in dss: Looking forward by reflecting on 40 years of progress," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pp.1–9, IEEE, 2011.
- [2] M. Osheroff, A. Jerome, M. Teich, J.M. Fhimss, M. Levick, F. Mba, M. Saldana, M. Velasco, T. Ferdinand, and D.F. Fhimss, *Improving outcomes with clinical decision support: An implementer's guide*, HIMSS Pub., 2012.
- [3] J.L.M. Amaral, A.J. Lopes, J.M. Jansen, A.C.D. Faria, and P.L. Melo, "Machine learning algorithms and forced oscillation measurements applied to the automatic identification of chronic obstructive pulmonary disease," *Computer methods and programs in biomedicine*, vol.105, no.3, pp.183–193, 2012.
- [4] F.L. Seixas, B. Zadrozny, J. Laks, A. Conci, and D.C.M. Saade, "A bayesian network decision model for supporting the diagnosis of dementia, alzheimer's disease and mild cognitive impairment," *Computers in biology and medicine*, vol.51, pp.140–158, 2014.
- [5] N. Chalortham, P. Leesawat, T. Ruangrajitpakorn, and T. Supnithi, "A framework of ontology-based tablet production supporting system for a drug reformulation," *IEICE Trans. Inf. & Syst.*, vol.E94-D, no.3, pp.448–455, 2011.
- [6] M.A. Musen, B. Middleton, and R.A. Greenes, "Clinical decision-support systems," in *Biomedical informatics*, pp.643–674, Springer, 2014.
- [7] M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *Journal of biomedical informatics*, vol.46, no.4, pp.744–763, 2013.
- [8] P. Fraccaro, D. O-Sullivan, P. Plastiras, H. O-Sullivan, C. Dentone, A.D. Biagio, and P. Weller, "Behind the screens: Clinical decision support methodologies—a review," *Health Policy and Technology*, vol.4, no.1, pp.29–38, 2015.
- [9] D. Riaño, F. Real, F. Campana, S. Ercolani, and R. Annicchiarico, "An ontology for the care of the elder at home," in *Artificial intelligence in medicine*, vol.5651, pp.235–239, Springer, 2009.
- [10] D. Riaño, F. Real, J.A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico, and C. Caltagirone, "An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients," *Journal of biomedical informatics*, vol.45, no.3, pp.429–446, 2012.
- [11] D.A. Alexandrou, I.E. Skitsas, and G.N. Mentzas, "A holistic environment for the design and execution of self-adaptive clinical pathways," *IEEE Trans. Inf. Technol. Biomed.*, vol.15, no.1, pp.108–118, 2011.
- [12] C. Bratsas, P. Bamidis, D.D. Kehagias, E. Kaimakamis, and N. Maglaveras, "Dynamic composition of semantic pathways for medical computational problem solving by means of semantic rules," *IEEE Trans. Inf. Technol. Biomed.*, vol.15, no.2, pp.334–343, 2011.
- [13] C. Bratsas, V. Koutkias, E. Kaimakamis, P.D. Bamidis, G.I. Pangalos, and N. Maglaveras, "Knowbasics-m: An ontology-based system for semantic management of medical problems and computerised algorithmic solutions," *Computer methods and programs in biomedicine*, vol.88, no.1, pp.39–51, 2007.
- [14] T. Andersen, J. Bansler, F. Kensing, J. Moll, and K.D. Nielsen, "Alignment of concerns: A design rationale for patient participation in ehealth," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp.2587–2596, IEEE, 2014.
- [15] A. Ahmed, A. Rebeiro-Hargrave, Y. Nohara, E. Kai, Z.H. Ripon, and N. Nakashima, "Targeting morbidity in unreached communities using portable health clinic system," *IEICE Trans. Commun.*, vol.E97-B, no.3, pp.540–545, 2014.
- [16] F. Daniel, F. Casati, P. Silveira, M. Verga, and M.N. Alin, "Beyond health tracking: A personal health and lifestyle platform," *Internet Computing, IEEE*, vol.15, no.4, pp.14–22, 2011.
- [17] G. Luo, "Triggers and monitoring in intelligent personal health record," *Journal of medical systems*, vol.36, no.5, pp.2993–3009, 2012.
- [18] M. Frost, A. Doryab, M. Faurholt-Jepsen, L.V. Kessing, and J.E. Bardram, "Supporting disease insight through data analysis: Refinements of the monarca self-assessment system," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp.133–142, ACM, 2013.
- [19] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A.T. Campbell, "Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp.3–14, ACM, 2014.
- [20] C.D. Buckingham, A. Ahmed, and A. Adams, "Designing multiple user perspectives and functionality for clinical decision support systems," in *Computer Science and Information Systems (FedCSIS)*,

- 2013 Federated Conference on, pp.211–218, IEEE, 2013.
- [21] P.J. Fortier, B. Puntin, and O. Aljaroudi, “Improved patient outcomes through collaborative monitoring and management of subtle behavioral and physiological health changes,” in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pp.1–10, IEEE, 2011.
- [22] R.T. Fielding, “Architectural styles and the design of network-based software architectures,” University of California, Irvine, 2000.
- [23] Q.Z. Sheng, X. Qiao, A.V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, “Web services composition: A decade’s overview,” *Information Sciences*, vol.280, pp.218–238 2014.
- [24] M. Lanthaler and C. Gütl, “Seamless integration of restful services into the web of data,” *Advances in Multimedia*, vol.2012, pp.1–14, 2012.
- [25] S. Speicher, J. Arwe, A. Malhotra, “Linked Data Platform 1.0,” <http://www.w3.org/TR/ldp/>, accessed Feb. 28, 2015.
- [26] R. Dobrescu and V. Purcarea, “Impact of information technology on the quality of health services,” in *Service orientation in holonic and multi-agent manufacturing control*, vol.402, pp.307–319, Springer, 2012.
- [27] K. Kawamoto, J. Jacobs, B.M. Welch, V. Huser, M.D. Paterno, G. Del Fiol, D. Shields, H.R. Strasberg, P.J. Haug, and Z. Liu, “Clinical information system services and capabilities desired for scalable, standards-based, service-oriented decision support: Consensus assessment of the health level 7 clinical decision support work group,” in *AMIA Annual Symposium Proceedings*, p.446, American Medical Informatics Association, 2012.
- [28] HL7 CDS and SOA working group, “HL7 Decision Support Service (DSS),” http://www.hl7.org/implement/standards/product_brief.cfm?product_id=12, accessed Feb. 3, 2015.
- [29] S&I Community & Staff: HeD, “Health eDecisions CDS Guidance Service,” <http://wiki.siframework.org/HeD+Use+Case+2+Consensus+-+CDS+Guidance+Service>, accessed Feb. 3, 2015.
- [30] M. Peleg, T. Broens, A. González-Ferrer, and E. Shalom, “Architecture for a ubiquitous context-aware clinical guidance system for patients and care providers,” *KR4HC’13/ProHealth’13*, pp.161–167, 2013.
- [31] M. Weitzel, A. Smith, S.D. Deugd, and R. Yates, “A web 2.0 model for patient-centered health informatics applications,” *Computer*, vol.43, no.7, pp.43–50, 2010.
- [32] OpenSocial, <http://www.w3.org/blog/2014/12/opensocial-foundation-moves-standards-work-to-w3c-social-web-activity/>, access Feb. 3, 2015.
- [33] G.W. Beeler, “HI7 version 3—an object-oriented methodology for collaborative standards development,” *International Journal of Medical Informatics*, vol.48, no.1-3, pp.151–161, 1998.
- [34] M. Buranarach, N. Chalortham, Y.M. Thein, and T. Supnithi, “Design and implementation of an ontology-based clinical reminder system to support chronic disease healthcare,” *IEICE Trans. Inf. & Syst.*, vol.E94-D, no.3, pp.432–439, 2011.
- [35] M. Fengou, G. Mantas, D. Lymberopoulos, N. Komninos, S. Fengos, and N. Lazarou, “A new framework architecture for next generation e-health services,” *Biomedical and Health Informatics, IEEE Journal of*, vol.17, no.1, pp.9–18, 2013.
- [36] F. Paganelli and D. Giuli, “An ontology-based system for context-aware and configurable services to support home-based continuous care,” *Information Technology in Biomedicine, IEEE Trans. Inf. Technol. Biomed.*, vol.15, no.2, pp.324–333, 2011.
- [37] F. Ongenaë, M. Claeys, T. Dupont, W. Kerckhove, P. Verhoeve, T. Dhaene, and F. De Turck, “A probabilistic ontology-based platform for self-learning context-aware healthcare applications,” *Expert Systems With Applications*, vol.40, no.18, pp.7629–7646, 2013.
- [38] C. Pautasso, “Composing restful services with jopera,” in *Software Composition*, vol.5634, pp.142–159, Springer, 2009.
- [39] W. Passornpakorn, S. Kamolphiwong, T. Kamolphiwong, and V. Chandeeying, “Design framework for ontology based interactive e-health services,” in *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, pp.389–394, IEEE, 2013.
- [40] “SPARQL Inferencing Notation,” <http://spinrdf.org/>, accessed March 1, 2015.
- [41] E. Prud’hommeaux and H. Knublauch, “SHACL (Shapes Constraint Language) Primer,” <http://w3c.github.io/data-shapes/data-shapes-primer>, accessed May 10, 2015.
- [42] K.D. Mandl, J.C. Mandel, S.N. Murphy, E.V. Bernstam, R.L. Ramoni, D.A. Kreda, J.M. McCoy, B. Adida, and I.S. Kohane, “The smart platform: Early experience enabling substitutable applications for electronic health records,” *Journal of the American Medical Informatics Association*, vol.19, no.4, pp.597–603, 2012.
- [43] B. Adida, A. Sanyal, S. Zabak, I.S. Kohane, and K.D. Mandl, “Indivox: Developing a fully substitutable personally controlled health record platform,” in *AMIA Annual Symposium Proceedings*, p.6, American Medical Informatics Association, 2010.
- [44] “FHIR: Fast healthcare interoperability resources,” <http://hl7.org/implement/standards/fhir/>, accessed Feb. 20, 2015.
- [45] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Groszof, and M. Dean, “Swrl: A semantic web rule language combining owl and ruleml,” *W3C Member submission*, vol.21, p.79, 2004.
- [46] Jena, <http://jena.apache.org/>, access Feb. 3, 2015.
- [47] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, “Metaq: A knowledge-driven framework for context-aware activity recognition combining sparql and owl 2 activity patterns,” *Pervasive and Mobile Computing*, 2015.



Wasin Passornpakorn received the B.S. degrees in Computer Engineering from Prince of Songkla University in 2011. During 2011–2015, he stayed with Centre for Network Research (CNR), Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand. He is now studying M.Eng. degree.



Sinchai Kamolphiwong received Ph.D. from the University of New South Wales, Australia. He is now an Associate Professor in the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Thailand. He publishes some of 100 technical papers. His main interest research areas are: NGN/NGI, Multimedia Communications, Tele-Medicine, Network Mobility, P2P, and Performance Evaluation.