Recommendation Service for Smart Space-based Personalized Healthcare System

Ilya Paramonov, Andrey Vasilyev
Petrozavodsk State University, Petrozavodsk, Russia
P.G. Demidov Yaroslavl State University, Yaroslavl, Russia
{ilya.paramonov, andrey.vasilyev}@fruct.org

Abstract—**The paper describes architecture of a recommendation service that provides advice on how to change the lifestyle to prevent appearance or progression of cardiovascular deceases. The service is an extension of the personalized healthcare system that has access to measurements of vital signs and the patient's electronic health record. The system is designed according to the smart spaces approach and implemented on top of the Smart-M3 platform. The paper presents the ontology used for representation of data of an electronic health record and the procedure of its retrieval with subsequent generation of sensible recommendations with the use of the rules engine. Justification of usefulness of the proposed approach is illustrated by implementation of several instructions from the seventh report of the Joint National Committee on prevention, detection, evaluation, and treatment of high blood pressure as rules for the recommendation service.**

I. Introduction

Personalized healthcare is a big aid for the outgrowing world population. With the help of this approach it becomes possible not only to preemptively detect the life threatening states, but to prevent them by pointing out negative trends coming from the lifestyle. People may seem to be aware of healthcare guidelines in general but providing personal recommendations that take into account current state of a person may increase the adoption rate of healthy habits.

Nowadays personalized healthcare systems constitute a rapidly developing area with its most prominent sector involving extensive usage of mobile technologies (m-Health systems). Their most common scenario is to provide a person with the mobile vital signs sensor, gather its data using the mobile device, and then process them on the device or in the cloud to provide actual information on health state directly to the person or her doctor.

One of a promising architecture for development of such systems is based on the smart spaces approach that allows to seamlessly integrate software components and provides a common view of the current situation that facilitates the reasoning. Prototypes of smart space-based m-Health systems were described in multiple scientific publications [1], [2], [3] covering different aspects of development such solutions. But an important aspect of providing recommendations to patients in smart space-based personalized healthcare system is not profoundly researched.

This paper makes a step towards solving a problem of the most effective utilization of the all the array of data about a patient including information from senors, electronic health record (EHR), and direct feedback from the user in smart space-based healthcare system. In our research we rely on a reference model of the personalized healthcare system providing digital assistant services for emergency situations [4]. We extend this system to support the prevention of emergency situations strategy by providing recommendations to the user on how to change the lifestyle. The recommendation service architecture is designed to be opened for seamless rule extension by an expert to support new situations and vital signs measurements.

The rest of the paper is organized as follows. Section II briefly describes the architecture of the personalized healthcare system we use in our research. Thereafter, Section III describes a reference scenario for the recommendation service and shows how it fits the whole system. In Section IV the ontologies necessary for the recommendation service's correct functionality are described, whereas the architecture of this service is presented and discussed in Section V. Section VI is devoted to recommendation rules organization and justification of usefulness of the proposed recommendation service. Conclusion summarizes the main results of the paper and highlights future proposals.

II. System architecture

The personalized healthcare system we supplemented with the recommendation system is designed to enhance the quality of life for the people with cardiovascular deceases. Currently, the majority of people diagnosed with such a disease restrict themselves in a environment (a home or a hospital) where they can receive a medical help in a case of emergency. Therefore, they limit their professional and personal interests, give up many opportunities. The aim of the system is to detect emergency situations as early as possible and supplement the provision of the first aid.

The system provides the services for patients, hospitals, emergency departments and volunteers. Patients use mobile devices to provide information about their health by responding to the questionnaire, input their latest measurements of vital signs or call for help. If prescribed by the physician they also wear an ECG sensor that constantly tracks the state of the cardiovascular system. The mobile device receives these measurements, processes them and sens to the servers for further in-depth analysis.

Another mobile device is used by the volunteers that can help the person in need in the case of emergency. They should provide the assistance before the arrival of the ambulance and during transportation to the hospital if needed. The software

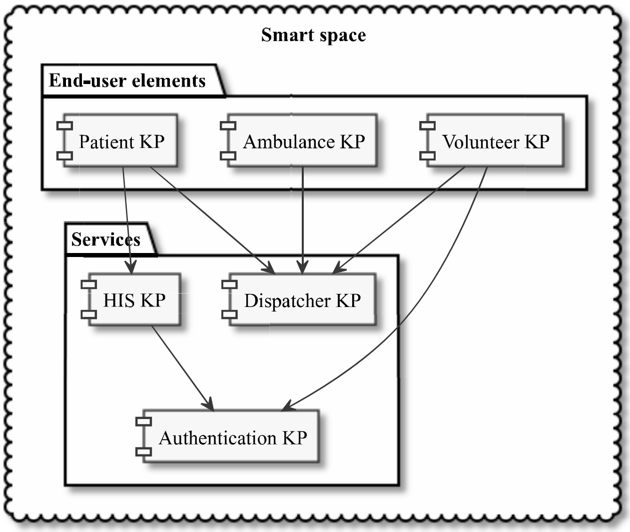

Fig. 1. The principal architecture of the Personalized Healthcare System

on the mobile device allows to publish current location, accept or reject the request for the help.

Server-side of system consists of the authentication service, dispatching service and electronic health record access service. First service authenticates another components of the system and provide them corresponding privileges. The dispatching service is responsible for providing an ambulance and a volunteer to handle an emergency. The EHR service provides access to a great number of medical documents that are stored in the hospital information system. These documents may be used by medical stuff to provide a proper treatment and by diagnostic software to increase the quality of predictions.

The system is developed using the Smart-M3 platform that enables communication between components by manipulation of shared knowledge [5]. The knowledge is represented in the form of data graph that can be expressed in the form of OWL classes and RDF triples. This data is stored in the component called Semantic Information Broker (SIB). Other components that interact with it and modify the graph are called Knowledge Processors (KPs). KP may also track a state of a subgraph and receive notifications when it changes. This feature enables the bidirectional data flow between components.

The architecture of the system is shown in Fig. 1. Patient KP runs on the mobile device and provides the information services for the patient. Volunteer KP and Ambulance KP run on mobile device of the corresponding parties and notify them in case of emergency. Authentication KP, HIS KP and Dispatcher KP run on the servers of the hospitals. The complete description of the interaction between components during the emergency situation is profoundly described in [4].

III. Reference scenario for recommendation KP

One of the ways to further improve the quality of life for people with cardiovascular deceases is to decrease the probability of emergency situations. According to [6] it may be possible by improvement of the life style via continuous monitoring of vital signs and provision of automatic recommendations on how to improve certain traits. These recommendations are not meant to substitute the visits to the doctor and ones decisions but to support the prescribed treatment.

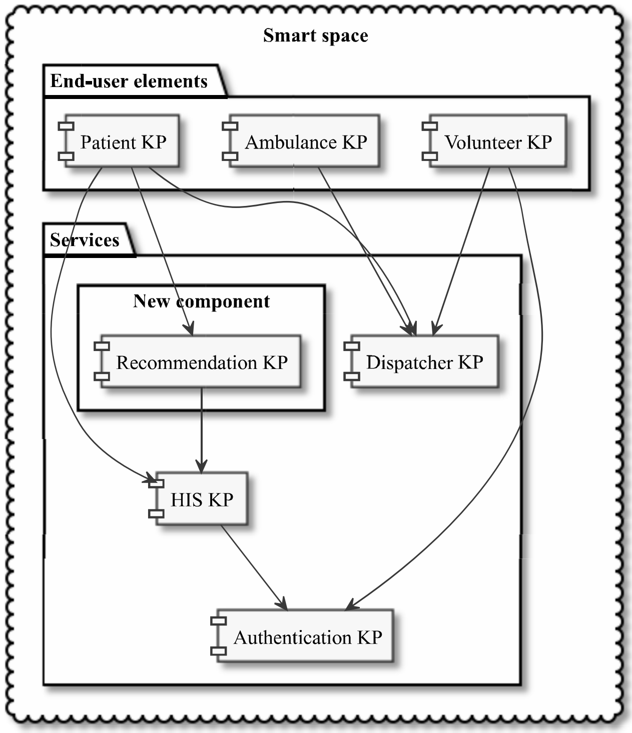

Fig. 2. Architecture of the system with the recommendation KP

The complete scenario for the user is rather straightforward. The personal mobile device notifies the patient at regular intervals that he or she should input the latest measurements of the vital signs and supplementary readings. When the measurements are done they are added to the EHR via the mobile device of the patient. The process of entering the measurements may be greatly simplified if they are performed using tools that may automatically transfer readings via wireless channels, but this requirement is not mandatory.

Upon receiving new records, the system analyzes them, already stored data and formulates a set of recommendations. These recommendations then become available for the user in the text from. The personal mobile device may notify the user once a day to review them. This will ensure that recommendations will be seen by the users. Doctors may also make a review of these recommendations, discuss them with the patients and extend them when needed.

In order to implement such a service in the personalized healthcare system we added a new component to the system — Recommendation KP that produces the recommendations, and updated the existing Patient KP. The new component runs on the server and accesses the EHR of the patient via HIS KP and collects questionnaires responses from the smart space. Updated architecture of the system is shown in Fig. 2. Generated recommendations are then published into the smart space. Details on how this data is organized in the smart space is discussed in the following section.

IV. Ontologies

A. EHR Ontology

During the development of the EHR ontology we studied the standards on interoperability between medical information systems, mainly the H7 protocol suite, and presumed that EHR can be considered as a set of documents that describe the interaction of the patient with the medical facility. These documents can describe appointments, results of the laboratory analysis, demographic data, and so on.

Also, such documents can include records that were gathered directly by the user using non-professional sensors. These records can include the glucose level, blood pressure, ECG measurements, activity status, food consumption, and so on. Though they are not clearly the complete medical data, but they provide a lot more data on person's health and may provide a reasonable enough data to detect the pathological state.

The ontology representation of the medical documents is shown in Fig. 3. The parent class for all other document types is *HISDocument*. It contains the date of document creation and the author that either may be one of the medical staff or the patient herself. These fields are then present in all the other documents. The date of submission is the mandatory parameter for all documents.

Entities of the *DemographicData* class describe the demographic data that is recorded during the submission of the patient to a hospital or her visit to a doctor. If the demographic data changes between submissions then the new document of this type is added to the patient's EHR. Previous document stays in the EHR, so previously added records stay consistent.

The *DoctorExamination* class describes the results of a visit to the doctor. The *examinationReason* and *visitOrder* fields contain the reason of appointment to the doctor. These fields are mandatory for each document, all the other fields are optional and filled by doctor when needed. The document can also describe the measurements performed using professional medical equipment. Currently only *height* and *weight* parameters are depicted in current version of the ontology but clearly they can be easily expanded to support more scenarios.

Entities of the *LaboratoryAnalysis* class contain data that comes from different laboratories and specialized facilities. Displayed set of fields contain core parameters coming from the general blood pressure analysis, a commonly used one. This class contain more data specific to the particular analysis. It is clear that this class cannot describe all possible types of laboratory analyses but it is meant to represent a general class of documents that can be stored in HIS.

The *Measurement* class of the ontology is not supposed to be used directly, it is required to distinguish records performed by the patient (using the non-professional equipment) and the professional ones. Particular measurement types are represented as separate classes: *BloodPressureMeasurement*, *WeightMeasurement*, *PhysicalActivityMeasurement*. Each of them contains the fields specific to represent corresponding data.

The *BloodPressureMeasurement* and *WeightMeasurement* classes describe the scalar results of measurements of corresponding values. The *PhysicalActivityMeasurement*, on the other hand, contains aggregate data on daily physical activity of the person.

B. Document request process

Another ontology required for recommendation service functioning is devoted to management procedures, espesially it describes the retrieval of the documents related to a particular person. All corresponding classes of the ontology are shown in Fig. 4.

When HIS KP connects to the smart space it publishes the instance of the *HospitalInformationSystem* class to reveal the availability of such a service. This instance can be connected to the instance of the *Hospital* class to indicate the medical facility this system belongs. The service can give an optional name in the corresponding field if it process only particular document types.

In order to initiate the request process the client KP adds an instance of the *HISRequest* class and connects it to the instance of *HospitalInformationSystem*. It is also necessary to specify the identity of the patient whose data should be retrieved. It is done by linking the request with the individual of the *Patient* class .

The fields of the *HISRequest* class allow to specify concrete parameters of the request. The *dateFrom* and *dateTo* fields specify the period of time the documents should belong to. The *requestsDocument* field specifies the type of the documents to be requested from HIS. The request can also specify the filter upon one of the fields of the document. The *fieldName* property specifies the name of the field to filter and the *searchString* property specifies the substring to search in the specified field.

When the instance of the *HISRequest* class is added to the smart space HIS KP retrives it and parses the parameters. If there is an error in the request then HIS KP adds an instance of *HISResponse* class linked to the request via *hasResponse* and *hasError* properties. It allows to distinguish requests that have already been processed from the new ones.

If there were no errors in the request it is executed upon the HIS document storage and resulted set of documents is added to the smart space. Then the instance of the *HISResponse* class is added and linked to the corresponding documents with the *hasDocument* property.

The requesting KP should track the apperance of the individual of the *HISResponse* class linked with the individual of *HISRequest* that have been added to the smart space. Upon receiving the corresponding notification KP should extract all required data from the smart space. Due to asynchronous nature of communication between the requester and HIS KP, the former must manage all documents added during the request-response procedure.

The proposed procedure allows to process requests from several clients simultaneously. It can be adapted to all situations when the service manages a shared resource that several

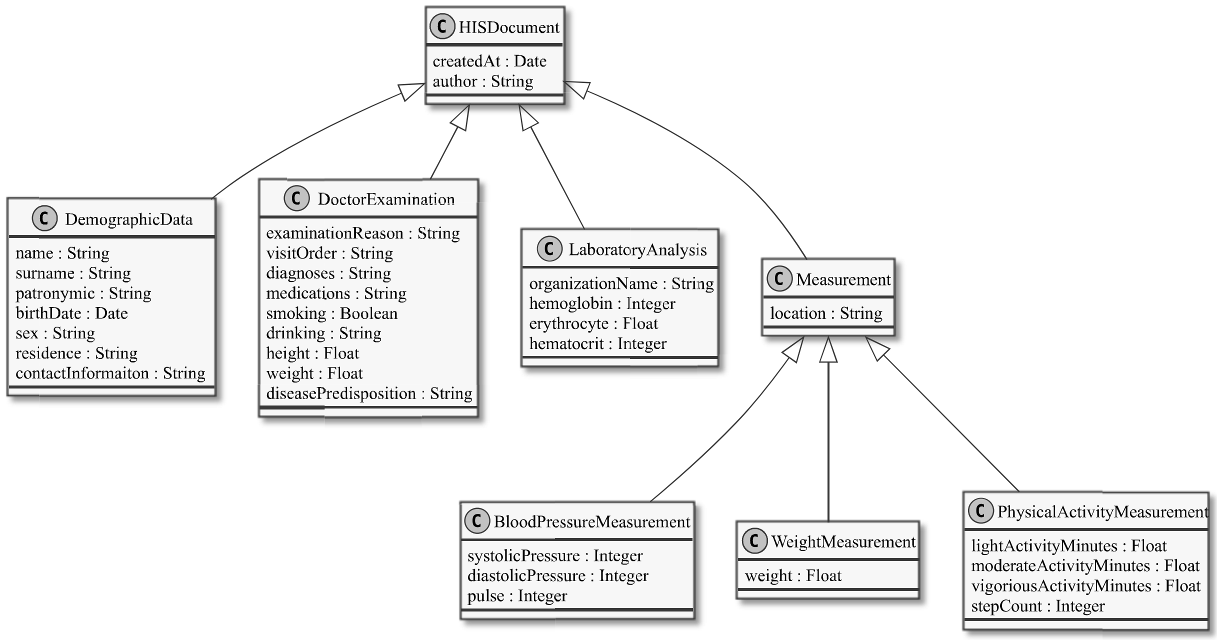

Fig. 3. Hierarchy of common HIS document classes

clients want to gain access to. One of such clients is discussed in the next section.

V. Architecture of recommendation service

For generation of recommendation we use Drools—a production-quality open source business rules management system solution [7]. It includes a business rules engine as well as additional tools for knowledge management, data modeling, and administration. It is also worth mentioning that Drools was successfully used in many e-Health applications (e.g., [8], [9]) that makes it a good choice for the task in question.

When using Drools rules engine all the data involved in reasoning (including its input and output) are enclosed inside Java classes, usually in the form of properties. The classes used in our prototype recommendation service are shown in Fig. 5.

The *Patient* class contains identification data of the patient (in the patientId property) and deduced recommendations for the patient as a sting array (the *recommendations* property).

The initial data for reasoning can originate from various data sources, including the patient's electronic health record (represented as the *EHR* class), data retrieved directly from the patient in the form of a questionnaire (the *Questionnaire* class), data from an external body sensor network ensuring continuous monitoring of vital signs (the *RealtimeSensorData* class), and so on. Each of these classes can either store the corresponding data or simply provide access to an external storage.

In this paper we consider only the EHR data source that acts as a gateway between the smart space, which delivers patient's medical data according to the ontology described in Section IV-A, and the Drools workspace, where the rule-based reasoning is made to generate proper recommendations to the patient.

The *EHR* class acts as KP in the smart space and retrieves requested data from EHR KP. In the simplest case this class would just translate requests for raw data into SSAP-requests used to communicate between KPs. However, generation of recommendation has a specificity: usually recommendations are formed from statistically processed data for certain periods of time. Due to this fact literal copying data from the smart space to the Drools workspace would lead to proliferation of the volume of the workspace and, as a result, to the performance issues during the reasoning. To prevent such unfavourable scenario we implemented primary processing of the data originated from EHR inside the *EHR* class and made the methods of this class calculate some accumulate functions applied to a data set over a period of time that ends by the time of the request:

- the *averageValueForRecentPeriod()* method returns the average value of a certain parameter over all documents of a specified type for the specified period (similarly work the methods with other prefixes, e.g., *totalValueForRecentPeriod()*);
- the *containsDocumentInRecentPeriod()* method returns true if there is a document with a specified type dated inside the specified period of time;
- the *containsValueInRecentPeriod()* method returns true if there is a parameter with a specified value in a document of a certain type dated inside the specified period of time.

As it is shown in the following section, such selection of methods makes them useful for recommendation generation, especially the way of the period determination reflects the fact that the most useful data for recommendations are usually the most recent ones.

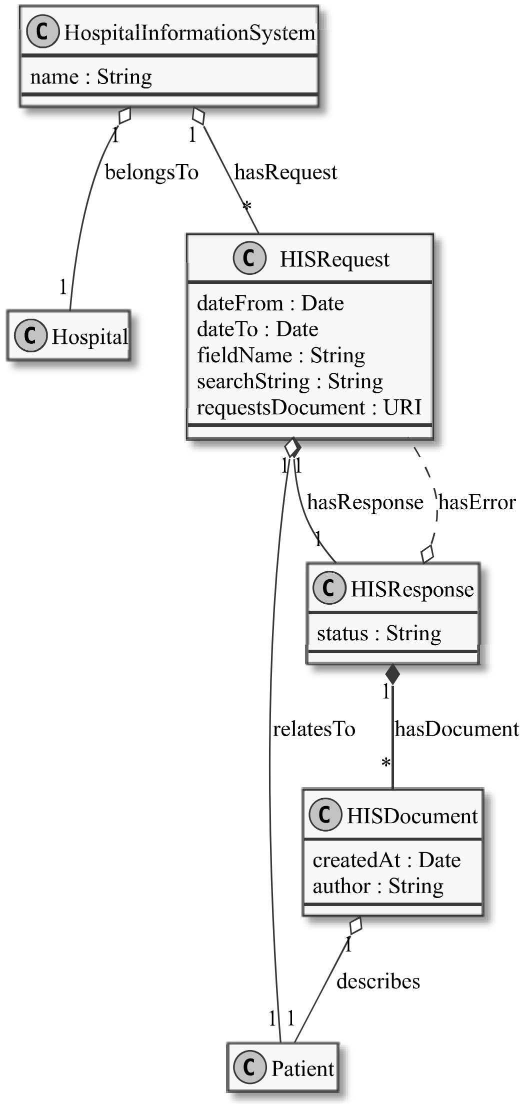

Fig. 4. Classes involved in document retrieval procedure

Another important responsibility of the *EHR* class is caching of the retrieved data to avoid repeated access to EHR KP. This feature as well as primary data processing allows to eliminate most of the performance issues in the recommendation service.

The reasoning may emit some intermediate facts regarding the patient's health state. In our prototype we involve a single type of facts designated as instances of the *SimpleFact* class with just one property *name* that describes the fact, e.g., "prehypertension" or "lifestyle management required". Involving such facts is important for medical professionals as it helps them suggest proper rules for the recommendation service using familiar terminology. Likewise, more complicated facts with greater number of properties can be easily supported.

Taking all these considerations into account, the algorithm of the recommendation service functioning can be represented as follows:

1) The recommendation service retrieves a request from the smart space for recommendations for a particular patient.
2) The instance of the *Patient* class with the data identifying the patient and empty *recommendations* array is created and placed into the Drools workspace.
3) The instances of the data source classes related to the patient are created/retrieved from an external service and placed into the Drools workspace.
4) The Drools engine is started to fill in the workspace with facts regarding the patient's health state, and collect the final recommendations in the *recommendations* property of the *Patient* class.
5) On finish of the reasoning process the recommendation service fills in a response with the generated recommendations and send it to the smart space.

VI. Organization of recommendation rules

To illustrate the way of organization of the recommendation rules and justify usefulness of these rules for patients with hypertension or prehypertension we selected several recommendations from the seventh report of the Joint National Committee on prevention, detection, evaluation, and treatment of high blood pressure (JNC 7) issued by US National Heart, Lung, and Blood Institute [10], and represented them as Drools rules involving the recommender service classes described in the previous section.

A. Identification of prehypertension/hypertension

JNC 7 defines prehypertension as systolic blood pressure ≥ 120 mmHg and < 140 mmHg and diastolic blood pressure ≥ 80 mmHg and < 90 mmHg. Unfortunately, there is no strict definition of the period during which these values should be observed. We considered the interval of one month to be a good choice for the rule:

```
rule "BP category"
when
  ehr : EHR()
  $monthlyAvgSysBP : ehr.averageValueForRecentPeriod(
      "BloodPressureMeasurement", "systolicPressure", 30)
  $monthlyAvgDiaBP : ehr.averageValueForRecentPeriod(
      "BloodPressureMeasurement", "diastolicPressure", 30)
  eval($monthlyAvgSysBP >= 120 &&
      $monthlyAvgSysBP < 140 &&
      $monthlyAvgDiaBP >= 80 &&
      $monthlyAvgDiaBP < 90)
then
  insert(new SimpleFact("prehypertension"))
end
```

When prehypertension is identified according to the rule above the corresponding fact about the patient's health state is added to the workspace for further reasoning.

Hypertension can be identified in much the same way when systolic blood pressure ≥ 140 mmHg, and diastolic blood pressure ≥ 90 mmHg.

B. Negative trend detection

Early detection of negative trends in vital signs is essentially important feature of the personalized medicine as

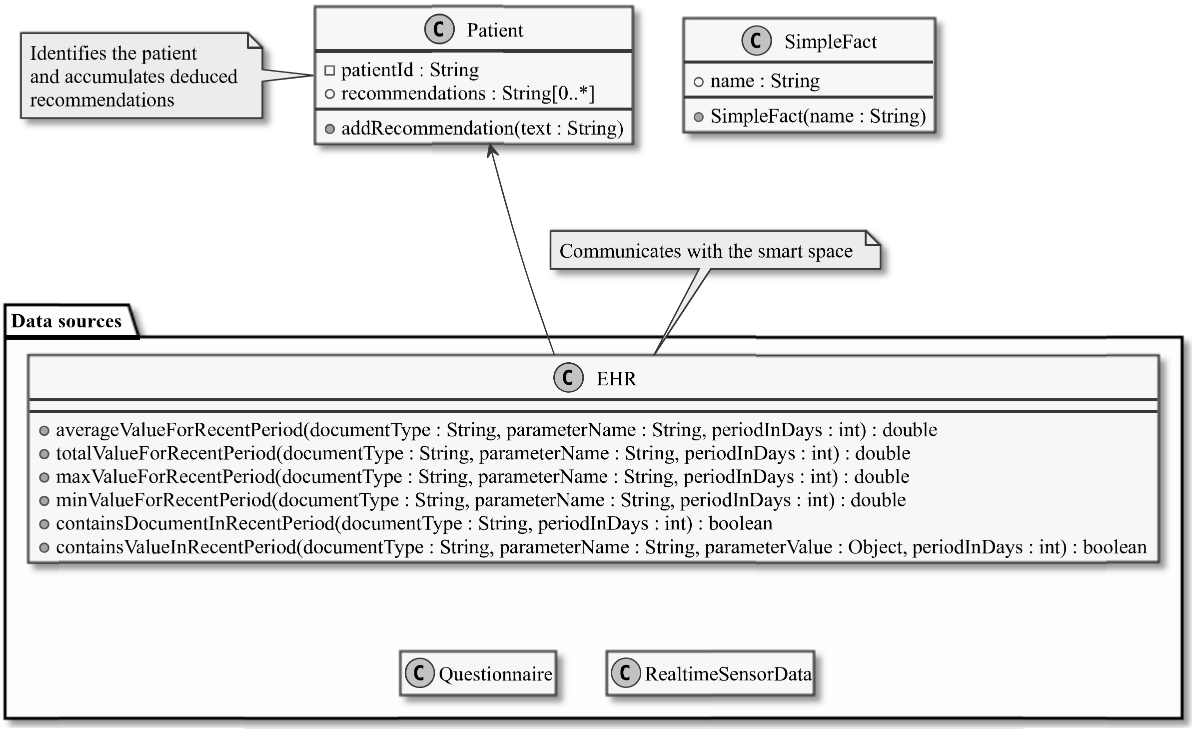

Fig. 5. Classes involved in Drools-based recommendation generation

it allows to prevent serious deceases by performing proper measures at the first signs of the negative trend.

The following rule identifies the negative trend when the average blood pressure over the last week surpasses the average value over the last month by more than 5 mmHg and adds the corresponding fact “BP negative trend” to the workspace:

```
rule ”BP negative trend identification”
when
  ehr : EHR()
  $monthlyAvgSysBP : ehr.averageValueForRecentPeriod(
      ”BloodPressureMeasurement”, ”systolicPressure”, 30)
  $monthlyAvgDiaBP : ehr.averageValueForRecentPeriod(
      ”BloodPressureMeasurement”, ”diastolicPressure”, 30)
  $weeklyAvgSysBP : ehr.averageValueForRecentPeriod(
      ”BloodPressureMeasurement”, ”systolicPressure”, 7)
  $weeklyAvgDiaBP : ehr.averageValueForRecentPeriod(
      ”BloodPressureMeasurement”, ”diastolicPressure”, 7)
  eval($weeklyAvgSysBP > $monthlyAvgSysBP + 5 or
       $weeklyAvgDiaBP > $monthlyAvgDiaBP)
then
  insert(new SimpleFact(”BP negative trend”))
end
```

Once the negative trend is identified, it can be used for further blood pressure management: patients with prehypertension lifestyle management actions are suggested, whereas hypertensive patients are urged to visit the doctor:

```
rule ”prehypertension + negative trend”
when
  patient: Patient()
  SimpleFact(name = ”prehypertension”)
  SimpleFact(name = ”BP negative trend”)
then
  insert(new SimpleFact(”Lifestyle management required”))
end

rule ”hypertension + negative trend”
when
  patient: Patient()
  SimpleFact(name = ”hypertension”)
  SimpleFact(name = ”BP negative trend”)
then
  patient.addRecommendation(”Consult the doctor to improve ” +
      ”your blood pressure management”)
end
```

C. Lifestyle management: physical activity

When lifestyle management is required and the patient’s BMI is more than 25 weight reduction is highly desirable for the patient. It can be achieved using various strategies. For example, if the physical activity per week is not enough giving a recommendation to exercise more looks relevant:

```
rule ”weight management (physical activity)”
when
  SimpleFact(name = ”Lifestyle management required”)
  EHR(averageValueForRecentPeriod(”WeightMeasurement”,
      ”BMI”, 30) > 25 &&
      totalValueForRecentPeriod(”PhysicalActivity”,
      ”moderateActivityMinutes”, 7) < 150)
then
  patient.addRecommendation(”Exercise more”)
end
```

Otherwise, a diet correction can be a good strategy.

D. Quit smoking

A very simple kind a recommendation can be given to the patient when its EHR says that he/she is a smoker:

```
rule "smoking"
when
  patient: Patient()
  EHR(containsValueInRecentPeriod("DoctorExamination",
      "smoking", true, 180))
then
  patient.addRecommendation("Quit smoking")
end
```

E. Visiting the doctor's office

Although the patient can (and even is suggested to) measure her blood pressure at home visiting the doctor's office for control rechecks should be performed regularly. Particularly, JNC 7 proposes to visit the doctor yearly for people with prehypertension and at least bi-monthly for hypertensive patients. The corresponding rules looks as follows:

```
rule "regular visiting the doctor's office (prehypertension)"
when
  patient: Patient()
  SimpleFact(name = "prehypertension")
  not(EHR(containsDocumentInRecentPeriod("DoctorExamination",
      365)))
then
  patient.addRecommendation("Visit the doctor's office " +
    "to recheck your BP")
end

rule "regular visiting the doctor's office (hypertension)"
when
  patient: Patient()
  SimpleFact(name = "hypertension")
  not(EHR(containsDocumentInRecentPeriod("DoctorExamination",
      60)))
then
  patient.addRecommendation("Visit the doctor's office " +
    "to recheck your BP")
end
```

VII. Conclusion

In this paper we proposed a prototype solution of the recommendation service integrated into personalized healthcare system built with the use of the smart spaces approach. The service is based on Drools—a production-quality open source business rules management system, and allows to transform the data stored in the patient's electronic health record as well as in other sources to recommendations on improvement of the patient's health state. The usefulness of the developed solution was illustrated by implementation of several instructions from the seventh report of the Joint National Committee on prevention, detection, evaluation, and treatment of high blood pressure as rules for the recommendation service.

The main direction of future research is integration of the recommendation service with real-time sources of medical data, particularly body sensor networks. The data gathered with the use of continuous monitoring provide wide spectrum of possibilities for more fine-grained and personalized recommendations, as well as new challenges for implementation.

Another important direction would be simplification of representation of the rules that can be achieved by using a Drools feature to define a domain-specific language. Such a decision would allow medical professionals with no programming skills to review or edit the rules, which is crucial for further improvement and maintenance of the whole system.

Acknowledgment

This research is financially supported by the Ministry of Education and Science of the Russian Federation within project #14.574.21.0060 (RFMEFI57414X0060) of Federal Target Program "Research and development on priority directions of scientific-technological complex of Russia for 2014–2020".

References

[1] F. Vergari, T. S. Cinotti, A. D'Elia, L. Roffia, G. Zamagni, and C. Lamberti, "An integrated framework to achieve interoperability in person-centric health management," *International journal of telemedicine and applications*, vol. 2011, 2011.

[2] C.-C. Chen and Y.-F. Chang, "Smart healthcare environment: design with RFID technology and performance evaluation," *Journal of Medical and Biological Engineering*, vol. 33, no. 4, pp. 427–432, 2013.

[3] D. G. Korzun, A. V. Borodin, I. V. Paramonov, A. M. Vasilyev, and S. I. Balandin, "Smart spaces enabled mobile healthcare services in internet of things environments," *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 6, no. 1, pp. 1–27, 2015.

[4] D. G. Korzun, A. V. Borodin, I. A. Timofeev, I. V. Paramonov, and S. I. Balandin, "Digital assistance services for emergency situations in personalized mobile healthcare: Smart space based approach," in *International Conference on Biomedical Engineering and Computational Technologies (SIBIRCON)*. IEEE, 2015, pp. 62–67.

[5] S. Balandin and H. Waris, "Key properties in the development of smart spaces," in *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*, ser. Lecture Notes in Computer Science, C. Stephanidis, Ed. Springer, Heidelberg, 2009, vol. 5615, pp. 3–12.

[6] E. I. Georga, A. N. Papadopoulos, and D. I. Fotiadis, "mHealth environments for chronic disease management," in *Handbook of Research on Trends in the Diagnosis and Treatment of Chronic Conditions*. IGI Global, 2016, ch. 1, pp. 1–32.

[7] M. Bali, *Drools JBoss Rules 5.0 Developer's Guide*. Packt Publishing Ltd, 2009.

[8] D. Li, S. Murthy, D. Sottara, C. G. Chute, S. M. Huff, J. Pathak, C. Endle, D. Suesse, and C. Stancle, "Modeling and executing electronic health records driven phenotyping algorithms using the NQF Quality Data Model and JBoss® Drools Engine," in *AMIA Symposium*, 2012, pp. 532–541.

[9] J. Lardon, H. Asfari, J. Souvignet, B. Trombert-Paviot, and C. Bousquet, "Improvement of diagnosis coding by analysing EHR and using rule engine: Application to the chronic kidney disease," *Digital Healthcare Empowering Europeans: Proceedings of MIE2015*, vol. 210, pp. 120–124, 2015.

[10] National High Blood Pressure Education Program *et al.*, "The seventh report of the Joint National Committee on prevention, detection, evaluation, and treatment of high blood pressure," 2004.