Semantic Technologies for Healthy Lifestyle Monitoring

Mauro Dragoni(✉), Marco Rospocher, Tania Bailoni, Rosa Maimone, and Claudio Eccher

Fondazione Bruno Kessler, Trento, Italy
{dragoni,rospocher,tbailoni,rmaimone,cleccher}@fbk.eu

Abstract. People are nowadays well aware that adopting healthy lifestyles, i.e., a combination of correct diet and adequate physical activity, may significantly contribute to the prevention of chronic diseases. We present the use of Semantic Web technologies to build a system for supporting and motivating people in following healthy lifestyles. Semantic technologies are used for modeling all relevant information, and for fostering reasoning activities by combining real-time user-generated data and domain expert knowledge. The proposed solution is validated in a realistic scenario and lessons learned from this experience are reported.

1 Introduction

In the last decades, health care systems in many countries have invested substantial effort in informing people about the benefits of adopting healthy behaviors in their lives, including the prevention of several medical problems such as cognitive decline, obesity, disability, and death from major chronic diseases [1]. Given the increasing popularity of mobile and personalized applications and devices (e.g., smart watches), a natural follow up of this effort is the development of platforms capable of providing user tailored advices, motivating people to adopt healthy behaviors. Although Internet-based and mobile technologies allow to collect data from personal devices, off-the-shelf wearable sensors, and external sources, exploiting these data to generate effective personalized recommendations and to engage people in developing and maintaining healthier patterns of living, is a challenging task. To carry out this task, a system providing personalized healthy lifestyle support has to take into account and reason on a considerable amount of knowledge from different domains (e.g. user attitudes, preferences and environmental conditions, etc.), in order to generate effective personalized recommendations, and to adapt the message in response to the environment and the user status. Examples of such knowledge include food content and nutrients, physical activities accompanied by information concerning their categorization and effort, user attitudes and preferences, linguistic knowledge, and smart environment information (places, weather, etc.) Moreover, the creation, update and maintenance of the knowledge bases of the system by domain experts must be

D. Vrandečić et al. (Eds.): ISWC 2018, LNCS 11137, pp. 307–324, 2018.
https://doi.org/10.1007/978-3-030-00668-6_19

facilitated in order to obtain a flexible system that can be deployed in different domains, for different target users, and able to evolve in response to the continuous progress of the domain knowledge.

In this paper, we present the knowledge-based solution implemented into the PerKApp platform, a system providing personalized healthy lifestyle recommendations and advices. This work, follows the preliminary description of PerKApp [2], where the abstract architecture of the overall platform has been presented. Here, we put emphasis on presenting (i) the key knowledge component, responsible for generating healthy lifestyle advices and recommendations, (ii) the collaborative environment used by domain experts and knowledge engineers to model the knowledge exploited by the system, and (iii) the observation about how the overall platform, and in particular the knowledge part, behaves in a real-world environment. Then, we notably stress the role of the rule-based reasoner that is responsible of analyzing the compliance between data provided by users and prescription rules modeled by experts, and of providing feedback when the compliance is violated.

The rest of the paper is structured as follows. Section 2 provides a brief overview of available platforms for the monitoring of people's lifestyles and on the use of rule-based reasoners in real-world scenario. In Sect. 3 we discuss the challenges we want to address. While, in Sect. 4 we present the scenario where the PerKApp platform was used: namely the *Key to Health* project. Then, Sect. 5 presents in more detail the knowledge component. In Sect. 6 we exemplify PerKApp in a concrete scenario, while an evaluation of reasoning performances is reported in Sect. 7 together with the lessons learned from this experience. Finally, Sect. 8 concludes.

2 Related Work

The promotion of healthy lifestyle is a recent topic and the availability of systems working in this direction is limited. Nevertheless, some approaches based on Semantic Web technologies have been previously proposed.

In [3,4], intelligent systems recommending exercises tailored to the user profile are presented. Both exploit an ontology for representing information about the patient profile, goals and health data, and about the exercises and their effects. Inference rules are then combined with ontological data for providing health advices, i.e., for suggesting exercises to perform.

A further proposal is the Medical Decision Support System discussed in [5]. The presented solution allows (i) the collection of patients' relevant information via a mobile application prompting questions related to the patient's medical background, and (ii) the creation of customized advices based on the information previously collected and on the changes of patient's lifestyle. The system exploits a number of ontologies, including: (i) the Patient Profile Ontology describing general information, medical information, and medical measurements; (ii) the Questionnaire Ontology, formalizing concepts representing generic components of a questionnaire, sub-questionnaires, questions, potential answers, etc.; (iii) the Lifestyle Ontology; and, (iv) the Guidelines Ontology.

Finally, [6] presented an approach for designing a semantic reasoning engine for supporting coaching profiles. This system uses a web-based interface for collecting user data and an ontology for analyzing and processing such data. This way, created profiles can be used to optimize the coaching activities of professionals.

More in general, reasoning-based systems have been already adopted in the health-care domain [7–9].

With respect to the works presented in the literature, we provide a fully-fledged Semantic Web based solution supporting (i) the modeling and storing of all the data required to provide personalized healthy lifestyle support, as well as (ii) the definition and execution, via a reasoning engine, of a dynamic sets of rules performing real-time monitoring of people lifestyles. The output of the reasoning task is then used for suggesting people to change their habits in order to follow healthier behaviors. Our proposal fits in the context of ontology-centric decision support systems [10], as all the data processed (e.g., user profile, meals) and produced (e.g., violations) by the system are stored in an ontology-based knowledge base. To the best of our knowledge, our contribution is innovative with respect to the state of the art due the capability of integrating a multiple actor modeling environment with the possibility of performing a rule-based reasoning activities, at a very alimentary fine-grained level, for monitoring people behaviors in the healthcare context.

3 Personalized Healthy Lifestyle Support

Systems for personalized healthy lifestyle recommendations and advices fall in the broad area of decision support. The goal of these systems is to help and guide users in taking healthy-informed decisions about their lifestyle, on aspects such as food consumption or everyday physical activities. Such systems have to take a decision (e.g., suggesting some physical exercises or conscious food consumption), similarly as a human expert would do, based on available data (e.g., nutrients ingested in the last meals, user health conditions), and to communicate these decisions to the users according to their preferred means and modalities.

Developing a healthy lifestyle personal assistant requires addressing several challenges from the knowledge perspective:

C1: to capture and model how experts work (“expert’s knowledge”) in assessing adherence to healthy lifestyle recommendations: among others, this involves representing different/heterogeneous kinds of data (e.g., food, nutrients, physical activities, user pathologies or needs) and how they interplay in defining a healthy lifestyle (e.g., the best practices an expert would recommend);

C2: to develop effective and efficient techniques implementing expert’s knowledge: these techniques apply expert’s knowledge on real-time users’ data, to assess how healthy is a user’s lifestyle and identify violations of healthy lifestyle recommendations;

C3: to embed these techniques in an efficient/scalable/flexible decision support system: the system has to acquire, process, apply expert's knowledge to provide motivational messages, on the data of a (possibly large) community of monitored users;

C4: to provide facilities supporting experts in specifying and revising the prescriptions for a healthy lifestyle applied by the systems: as healthy lifestyle best practices are not immutable, and sometimes not even universally shared, but they continuously evolve (e.g., new prescriptions for new typologies of users), there is a need to facilitate experts in defining and revising the prescription used in the systems.

In the next sections, we show how the use of semantic technologies is particularly suited to address these challenges, and can foster further research and development in the area of personalized health lifestyle support.

4 The PerKApp Platform

The PerKApp platform,[1] schematically summarized in Fig. 1, is a personalized healthy lifestyle recommendations and advices system implementing the 3-layer architecture described in Sect. 3.

The Input/Output layer consists in a mobile application[2] supporting the acquisition of user information about consumed foods and physical activities. Concerning foods, users are able to select them from a list of almost 6,000 basic foods and dishes; while for physical activities, we connected the PerKApp platform with the APIs of the most popular services and wearable devices (Google Fit, Misfit, Polar, and Garmin), or users can specify what they did among the more than 800 activities available. The Persuasive Layer consists in the development of a natural language generation component integrating a set of persuasive strategies supporting users in behavioral change. The analysis of the challenges linked with these layers (i.e. the effort necessary for providing data rather than the methodology for communicating with users effectively) and how they have been addressed is out of the scope of this paper.

Semantic Technologies in PerKApp. Ontologies and rules are used to address Challenge C1. Ontologies enables to represent in a connected and comprehensive way all the content relevant for the given domain. More precisely, the HeLiS ontology [11], comprises different interconnected modules formally representing expert's knowledge and the data needed by experts to assess the adherence of a user to healthy lifestyle best practices. The choice of adopting an ontological representation enables both to easily reuse existing resources, and to make available to other initiatives the results of the modeling performed in PerKApp.

[1] http://perkapp.fbk.eu/.

[2] The application is available at https://play.google.com/store/apps/details?id=eu.fbk.trec.lifestyle. The current version is only in Italian. We are in the process of translating it in English and German.

For instance, for the food domain PerKApp integrates content from the OpenFoodFacts database, which provides information about foods through an RDF dump or an API service. At the same time, as readily reusable resources about physical activities are currently lacking, their ontological modeling in PerKApp can be reused in other initiatives needing structured information about them.

Semantic rules are used to capture the checks performed by healthy life-style experts, to detect user's violations of healthy lifestyle best practices, and to decide whether some recommendation or motivational message has to be notified to the user (the rules do not decide on the way or terminology to be used in the message, a task addressed in the Persuasion Layer). The collection, modeling, and revision of such rules (Challenge C4) by domain experts is supported by the integration of specific user facilities and informal templates into MoKi [12], a collaborative tool that has already been used for supporting the work of domain experts in the health domain [13].

The application and evaluation of these rules on real-time user data (Challenge C2) is performed by an inference engine module, powered by RDFpro [14], a tool enabling several RDF and Named Graph manipulation tasks, including rule-based reasoning. In particular, RDFpro has been configured with specific entailment rules featuring custom SPARQL functions to support the specific requirements of the presented scenario.

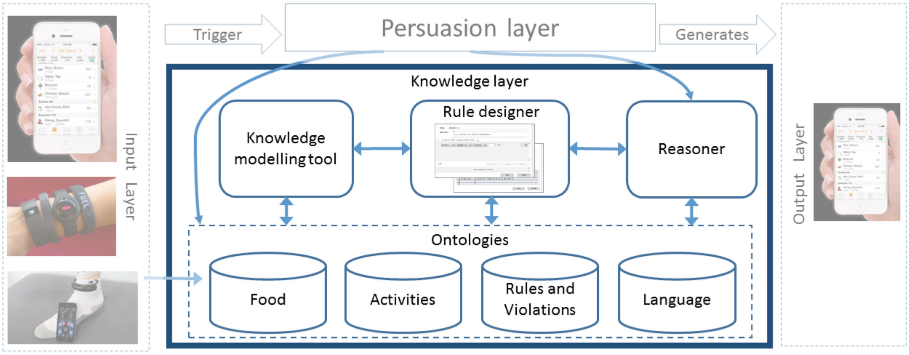


Fig. 1. The overview of the architecture implemented for the PerKApp project.

Finally (Challenge C3), all the data management part is supported by state-of-the-art triplestores, using named graphs to efficiently organize data (e.g., by user) and efficiently tuned SPARQL queries for all data access, manipulation and analysis tasks. As a result, PerKApp provides an evaluation testbed for the joint use of these technologies at scale and, as mentioned in Sect. 2, it contributes filling the current gap about the realization of resources and services integrating knowledge and reasoning capabilities exploiting food, physical activity, and user personalized information in synergy.

Key to Health. As a specific case study, PerKApp has been deployed and evaluated in the context of the project *Key to Health* on workplace health promotion (WHP) inside our institution (FBK). WHP, defined as "the combined efforts of employers, employees, and society to improve the mental and physical health and well-being of people at work",[3] aims at preventing the onset of chronic diseases related to an incorrect lifestyle through organizational interventions directed to workers. Actions can concern the promotion of correct diet, physical activity, and social and individual well-being, as well as the discouragement of bad habits, such as smoking and alcohol consumption. Within the Key to Health project, PerKApp has been used by 92 FBK's workers (both researchers and employers) as a tool to persuade and motivate them to follow WHP recommendations. This study represented a first step within the "Trentino Salute 4.0" digital health initiative aiming at extending the availability of the PerKApp platform at the whole Trentino Province before the end of 2018 and at Italian National level during the two-year period 2019–2020.

5 Knowledge Layer Components

In this Section, we focus on the components that are mainly involved in the Knowledge Layer of PerKApp. We show how the platform supports the domain experts in defining monitoring rules (Sect. 5.1). Then, we describe how reasoning is implemented to evaluate these rules (Sect. 5.2). Here, we skip a presentation of the underlying ontology which details are described in [15]. However, we report a schema including the main concepts in Fig. 2.

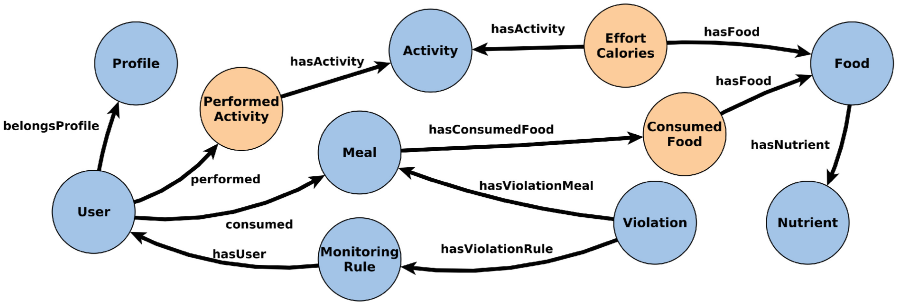


Fig. 2. The HeLiS ontology.

[3] Luxembourg Declaration on Workplace health promotion in the European Union. 1997.

5.1 Experts Support Facilities

As mentioned in Sect. 3, one of the challenges we wanted to address was the development of flexible facilities supporting domain experts in defining rules for monitoring user behaviors. Moreover, one of the requirements is to support the collaborative work of domain experts. Indeed, the definition of dietary rules may lead to disagreement between experts in deciding, for example, the amount of a specific food that a person has to eat during a day and the weekly frequency.

Domain experts' facilities have been implemented as an extension of MoKi [12]. The modeling of concepts and individuals related to the `Food`, `Activity`, and `User` branches is supported by views and forms enabling create, read, update, and delete (CRUD) operations. MoKi provides domain experts with several collaboration facilities (e.g., discussion pages, revision history), whose effectiveness was empirically confirmed [16]. Here, we want to focus on two flexible facilities offered by the tool: the definition of rules and the monitoring of user's behavior.

The definition of rules is supported by an interface that permits domain experts to create, view, or revise rules in a collaborative way, covering both the dietary and activity domains. Each form element is associated with one of the properties of a `MonitoringRule` instance and can be populated with actual values supplied by expert to instantiate the rule template, avoiding the need to learn a formal language for expressing rules. Within the interface, domain experts are able to select the timing dimension that the rule refers to, the type of check that the rule has to perform on user data (e.g., the presence of a specific food or nutrient), the type of entity that the rule has to monitor (i.e., `Food`, `Nutrient`, or `Activity`), and the name identifying that specific entity. Then, a comparison operator has to be selected together with the type of operation to be performed on the acquired data. Finally, other information, like a priority used for reporting, can be associated with the rule.

The second facility we want to highlight is the real-time monitoring of users' behaviors where experts can explore the reports generated by the tool. In this particular case, experts can observe, for a specific meal, the number of associated violations and their related details. The usefulness of this report is two-fold: on the one hand, experts can observe how much a user is respecting the rules assigned; on the other hand, the provided information is useful also for the experts themselves. Indeed, they can analyze if the evaluation of the modeled rules in a real-world context is correct or not. This is a critical aspect in a collaborative environment and, in particular, in the medical domain due to possible disagreements between experts in validating the results of the monitoring activity. In Sect. 7.4, we report a brief discussion about this point by presenting the results of a comparative analysis between experts on the output provided by the reasoner in four different real-world scenarios.

5.2 Rule-Based Reasoning

Reasoning in PerKApp has the goal of verifying that a user's lifestyle is consistent with the monitoring rules defined by domain experts, detecting and possibly materializing violations in the knowledge base, upon which further actions may be taken. Reasoning can be implemented via the fixed point, forward chaining evaluation of IF-THEN *entailment rules* (cf. *monitoring rules*, which are RDF individuals in the HeLiS ontology) that implement the semantics of OWL 2 RL (to account for TBox declarations in the ontology) and of monitoring rules by matching non-conforming patterns in RDF data and asserting corresponding `Violation` individuals.

Aspects of Interest. To effectively implement monitoring rules via rule-based reasoning, the following aspects need to be taken into accounts:

- *Non-monotonic reasoning.* The evaluation of monitoring rules involves aggregations (e.g., total calories of a meals) and negation-as-failure (e.g., a food is not consumed if there is no `ConsumedFood` for it), which make reasoning a non-monotonic process where acquiring new data may invalidate previous inferences.
- *Generation of new individuals.* The materialization of detected violations implies the creation of new individuals in the knowledge base, whose identity is tightly linked to the one of existing individuals (e.g., a meal for a violation affecting it).
- *Interplay with RDFS/OWL reasoning.* The evaluation of monitoring rules (e.g., that a meal must contain cereals) relies on the prior computation of OWL 2 inferences (e.g., to determine that `RiceFlakes` are `Cereals`), while at the same time resulting violation assertions can be the subject of further RDFS/OWL inferences.
- *Static/dynamic data.* The *Food*, *Activity*, and *Monitoring* branches of the ontology are largely static and are available up front, allowing to pre-compute inferences over this data and/or to optimize the entailment rules used real-time on user data.
- *Per-user reasoning.* As monitoring rules operate on a single user, reasoning can be done on a per-user basis achieving scalability via the parallel/distributed processing of multiple users at the same time.

Reasoning Workflow. We perform reasoning in PerKApp using RDFpro [14], a tool that allows us to consider the aforementioned aspects by supporting out-of-the-box OWL 2 RL and the fixed point evaluation of `INSERT... WHERE...` SPARQL-like entailment rules that leverage the full expressivity of SPARQL (e.g., `GROUP BY` aggregation, negation via `FILTER NOT EXISTS`, derivation of RDF nodes via `BIND`). We organize reasoning in two *offline* and *online* phases as shown in Fig. 3.

Offline, a one-time processing of the *static* part of the ontology (*Food*, *Activity*, *Monitoring* branches) is performed to materialize its deductive closure, based on OWL 2 RL and some additional pre-processing rules that identify the most specific types of each `Nutrient` individual (this information greatly helps in aggregating their amounts). The resulting *closed ontology* is then used to *expand* a set of PerKApp-specific entailment rules that evaluate monitoring rules on the *dynamic* user data described by the *User* branch of the ontology. The expanded rules are responsible of: (i) computing the OWL 2 RL deductive closure of user data; (ii) assert `Violation` individuals; and (iii) enrich these individuals with various triples according to the `Violation` model in the ontology. The expansion process consists in pre-computing the matches of specific rule body atoms that can only appear in the static ontology (e.g., `rdfs:subClassOf` statements), producing variable bindings that are replaced in entailment rules to derive a new (larger) set of simpler rules that can be evaluated more efficiently.

Online, each time the reasoning is triggered (e.g., a new meal or performed activity is entered), the user data is merged with the closed ontology and its deductive closure based on the *expanded rules* is computed. This process can occur both on a per-user basis or globally on the whole knowledge base. The resulting `Violation` individuals and their RDF descriptions are stored back in the knowledge base.

Rule Example. We report below an example of PerKApp entailment rule responsible of detecting violations of *DAY-Rules* constraining the daily amount of calories consumed.

```
INSERT { ?v :hasViolationRule ?rule; :hasViolationQuantity ?cal;
            :hasViolationUser ?user; :hasTimestamp ?checkTs. }
WHERE { { SELECT  ?rule ?user ?timestamp (SUM(?a / ?e * ?fc) AS ?cal)
          WHERE { ?rule a :MonitoringRule; :timing :Day; :command "hasCalories";
                        :appliesTo/:containsUser? ?user.
                  ?meal :hasUser ?user; :hasTimestamp ?ts; :hasConsumedFood ?cf.
                  ?cf :amountFood ?a; :hasFood [:amountCalories ?fc; :ediblePart ?e]
                  BIND (:computeTimestamp(:Day, ?ts) AS ?checkTs) }
          GROUP BY ?rule ?user ?checkTs }
        ?rule :hasOperator ?op.
        OPTIONAL { ?rule :hasMonitoredValue ?val }
        OPTIONAL { ?rule :hasMonitoredValueInterval ?interval.
                   ?interval :lowerBound ?lower; :upperBound ?upper. }
        FILTER (?op = "equal" && ?cal != ?val   || ?op = "less" && ?cal >= ?val ||
                ?op = "greater" && ?cal <= ?val || ?cal < ?lower || ?cal > ?upper)
        BIND (:mintViolation(?rule, ?user, ?checkTs) AS ?v) }
```

The nested `SELECT` query takes a monitoring rule and groups affected meals by user and by the *check* timestamp where daily rules for that meal should be evaluated (computed via a SPARQL custom function implemented in Java). The sum of calories is computed for each monitoring rule, user, and check timestamp. The rest of the `WHERE` clause evaluates the monitoring rule operator against the specified target value or value interval. If a violation is detected, a URI for a

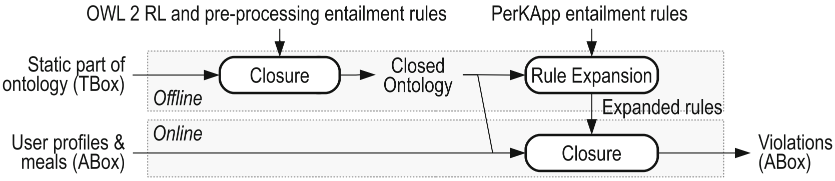


Fig. 3. Reasoning workflow in PerKApp.

new Violation individual is minted and that individual and its core violation data are asserted by the INSERT clause.

Scheduling. We leverage the distinction among *UB-Rules*, *DAY-Rules*, and *WEEK-Rules* to schedule the online reasoning phase at different times to improve efficiency:

- *UB-Rules* are evaluated in real-time on a meal or performed action immediately after they are received on the server, to enable the possibility of providing an immediate feedback to users. This kind of reasoning suffers of the possibility of high concurrency due to the amount of people providing their data during a small time interval. Hence, by restricting to evaluate only the necessary *UB-Rules* after each meal or activity, we are able to manage potential bottlenecks in elaborating data.
- *DAY-Rules* and *WEEK-Rules* are evaluated in background, respectively on a daily or weekly basis. Their evaluation implies the collection and aggregation of relevant amounts of data, requiring significant time for being analyzed. Therefore, the evaluation of these rules is scheduled at night or in other time slots when the system is relatively idle, so to avoid affecting the performance of the whole system.

6 The PerKApp Platform in Action

In order to explain in a clearer way which are the structures of the monitoring rules generated through the actions of the domain experts and the data acquired by the users, we describe here a simple scenario that will help the comprehension of how the reasoner, described in Sect. 5.2, works.

Let us consider the following scenario: Michelle is a sales agent of an important company. Her working days are very stressful especially in the morning, where she often has to drive for long distances for reaching her customers. After her workday, she uses to run as relaxing practice. In the last period, she started to have some dizziness during the morning and, sometimes even while driving. After a colloquium with her doctor, he noticed that she is used to have very light and quick breakfast and that the level of physical activity she does is maybe too much with respect to the diet she is following. For this reason, her doctor asked her to use the PerKApp application for monitoring what she eats in the morning and which is the actual level of physical activity she does. The doctor ask her to follow three rules: (i) the amount of calories for breakfast has to be higher than 250; (ii) the breakfast always has to contain at least 80 grams of cereals; and, (iii) she does not have to run more than 45 min.

At first stage (cf. Figure 4), the doctor creates a user associated to Michelle, and details her profile. Then, he defines the three rules that the PerKApp system has to validate every day.

For the first two days, Michelle correctly inserts the data about her breakfast and physical activity as shown in Fig. 5.

Based on this data, the ontology integrated within the PerKApp platform and the RDF encoding of the configured monitoring rules, the PerKApp reasoner

```
:MorningDizziness a :Profile.

:Michelle a :User; :hasUserId "4953"^^xsd:int; :belongsProfile :MorningDizziness.

:MR1 a :MonitoringRule; :appliesTo :MorningDizziness; :timing :Breakfast;
   :monitoredEntityType :Food; :monitoredEntity :Cereal; :command "contains";
   :hasOperator ">"; :hasMonitoredValue "80"^^xsd:int; ...
:MR2 a :MonitoringRule; :appliesTo :MorningDizziness; :timing Breakfast;
   :command "hasCalories"; :hasOperator ">"; :hasMonitoredValue "250"^^xsd:int; ...
:MR3 a :MonitoringRule; :appliesTo :MorningDizziness; :timing Day;
   :monitoredEntityType :Activity; :monitoredEntity :Running; :command "contains";
   :hasOperator "<"; :hasMonitoredValue "45"^^xsd:int; ...
```

Fig. 4. Example of a user profile with the description of the associated rules.

```
:Michelle :consumed :Breakfast-4953-1, :Breakfast-4953-2.
:Michelle :performed :Running-4953-1, :Running-4953-2.

:Breakfast-4953-1 a :Breakfast; :hasTimestamp "2017-04-14T07:19:00Z"^^xsd:dateTime;
   :hasConsumedFood [ :hasFood :GoatMilk; :amountFood "250"^^xsd:int ],
                    [ :hasFood :RiceFlakes; :amountFood "100"^^xsd:int ].
:Breakfast-4953-2 a :Breakfast; :hasTimestamp "2017-04-15T07:34:00Z^^xsd:dateTime";
   :hasConsumedFood [ :hasFood :TeaInCups; :amountFood "300"^^xsd:int ],
                    [ :hasFood :Apple; :amountFood "150"^^xsd:int ].

:Running-4953-1 :hasTimestamp "2017-04-14T17:45:00Z^^xsd:dateTime";
    a :PerformedActivity; :hasActivity :Running; :activityDuration "42"^^xsd:int .
:Running-4953-2 :hasTimestamp "2017-04-15T18:20:00Z^^xsd:dateTime";
    a :PerformedActivity; :hasActivity :Running; :activityDuration "38"^^xsd:int .
```

Fig. 5. Example of data provided by users and transformed in their structured representation.

determines that the amount of calories of the two breakfasts and the amount of performed physical activity satisfy the monitoring rules MR1 and MR3, respectively. However, the reasoner determines that no cereals where consumed in the second breakfast, leading to a violation of the cereals rule (MR1). The Violation individual shown in Fig. 6 is thus asserted in the knowledge base.

```
:violation-4953-1-20170415 a :Violation;
   :hasViolationRule :MR1; :hasTimestamp "2017-04-15T07:34:00Z"^^xsd:dateTime;
   :hasViolationUser :Michelle; :hasViolationMeal :Breakfast-4953-2;
   :hasViolationQuantity "0"^^xsd:int; :hasViolationExpectedQuantity "80"^^xsd:int.
```

Fig. 6. Example of violation (excerpt).

As a result, Michelle did not receive any feedback from the system after the first breakfast as she respected all the monitoring rules provided by the doctor. The same happened also after the registration of her physical activities as she did not exceed the amount of minutes suggested by the doctor. Whereas, she receives a textual feedback from the PerKApp application after the second breakfast, communicating that she did not consume the suggested quantity of cereals. This kind of feedback aims to remember to Michelle that she has to follow doctor's suggestions for avoiding the arise of dizziness during the morning. At the same time, the doctor is able to see the violations of Michelle. This way, by combining the generated violations and Michelle's feedback about her health status after using the PerKApp application for a certain period of time, the doctor can verify if his hypothesis were true.

7 Evaluation

In this Section, we report the evaluation activities we performed on the PerKApp platform with a particular emphasis on the scalability aspect. Indeed, our main goal is to validate the suitability, and potential drawbacks, concerning the deployment of the PerKApp platform in bigger scenarios. We start by presenting an analysis of the data we collected during the first forty-five days of the *Key to Health* project (Sect. 7.1) and by extending our observations to a synthetic scenario simulating wider contexts (Sect. 7.2). Then, we report some lessons learned from the development and deployment of the PerKApp platform (Sect. 7.4).

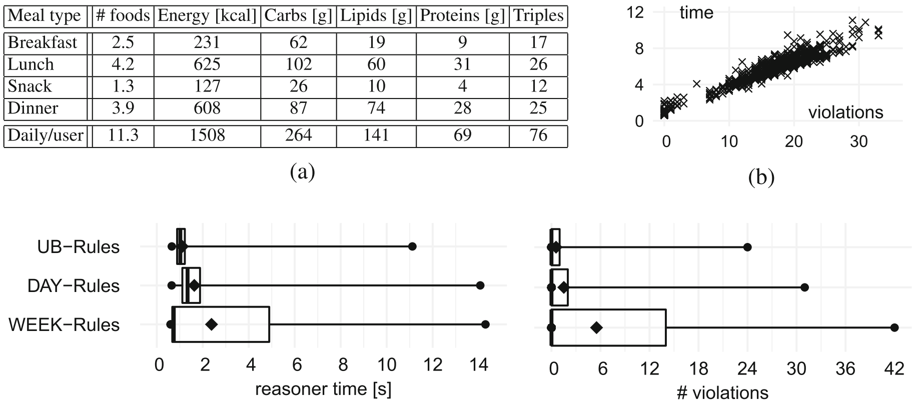
7.1 Analysis of Real Data

Within the *Key to Health* project, PerKApp has been used in a period of 45 days by 92 users , 49 of which have reported data about their meals on a regular basis, and about their physical activities only occasionally. We thus focus on the meal data provided by those 49 users, analyzing it as well as the reasoning process evaluating monitoring rules.

Figure 7a provides an average characterization of the meals inserted by users based on their type (breakfast, lunch, snack, dinner), in terms of number of composing foods, calories and main nutrients (carbs, lipids, proteins), and the number of triples necessary to encode this meal data in the triplestore; a daily per-user aggregate is also reported. The data give evidence (on average) of a 1500 Kcal daily diet, although users may have omitted some consumed food or underestimated its amount, either unintentionally (i.e., they forget to enter a meal) or intentionally (e.g., to "hide" the consumption of unhealthy foods). The number of triples needed for a meal is in the order of few tens, suggesting that the representation of meals in the ontology is compact and thus makes easier to store and manipulate large numbers of meals. On average, 76 triples per user per day are currently needed, meaning that a small PerKApp deployment supporting 1B triples would manage to store one month worth of meal data for over 400 K users.

The box plots in Fig. 7c summarize the distributions of reasoning time and the number of output violations for the three different times reasoning is performed:[4] after each single meal to check *UB-Rules*; at the end of each day for *DAY-Rules*; and at the end of each week for *WEEK-Rules*. Both reasoning time and the number of violations increase moving from a single meal to a whole week, as more input data is being processed and more rules can be violated. Each

[4] Deployment machine: 2 CPUs Intel Xeon X5690 3.47GHz, 8GB RAM, 160GB HDD.



Meal type	# foods	Energy [kcal]	Carbs [g]	Lipids [g]	Proteins [g]	Triples
Breakfast	2.5	231	62	19	9	17
Lunch	4.2	625	102	60	31	26
Snack	1.3	127	26	10	4	12
Dinner	3.9	608	87	74	28	25
Daily/user	11.3	1508	264	141	69	76

(a)

(b)

(c)

Fig. 7. Evaluation on *Key to Health* scenario: (a) Consumed foods, nutrients and RDF triples for user-supplied meals, by type and aggregated per user/day; (b) Correlation between reasoning time and output violations (WEEK-Rules); and (c) Reasoning time and output violations distributions.

violation corresponds on average to the assertion of 18 triples in the triplestore. Reasoning takes around 1 s in half of the cases (medians in box plots), but other cases require much more time (up to 14 s for *WEEK-Rules*) and lead to an increase in the average reasoning time (diamond means in box plots). This variability is the result of the interplay of different factors. In particular, we observe a significant correlation between reasoning time and the number of produced violations, shown in the scatter plot of Fig. 7b, that may be partly justified by the fact that additional entailment rules are triggered to populate detected violations.

7.2 Assessing Reasoning Performance

We performed some scalability tests to assess the reasoning performance under different settings. We focused our tests on two aspects. Firstly, we wanted to observe the time needed by the reasoner for completing the inference process and for creating the instances of the `Violation` class, where necessary. Concerning this validation, we recall that the reasoning is performed per-user (as mentioned in Sect. 5.2)

Secondly, we desired to analyze the throughput of the system with the aim of discovering which is the maximum load bearable by the system per unit of time. In particular, we focused our exploration on knowing how many users can be analyzed per minute by the system. Collected values may be indicative for estimating possible hardware requirements for making our platform deployable into different, larger scenarios, as planned in the "Trentino Salute 4.0" digital health initiative.

The configuration of the evaluation procedure deals with two parameters: (i) the number of days monitored for each user, and (ii) the number of monitoring rules contained in each profile. Concerning the number of days, we used values suggested by the domain experts as significant for monitoring purposes: 1, 7, 14, and 30 days. For the rules, we considered profiles with 1, 10, 50, and 100 rules, to simulate both `simple` profiles (1, 10 rules — e.g., typical of the dietary monitoring for healthy people) as well as `complex` ones (50, 100 rules — representative of more sensitive situations, with users to be monitored due to particular health conditions).

The graph shown on the left in Fig. 8 reports the observations of the time required for performing the reasoning activity.[5] On the x-axis, we reported the number of days for which user's data are provided, while on the y-axis, we reported the time necessary for completing the reasoning. There are two aspects that we may notice. Firstly, the timing necessary for reasoning in the more complex scenario (30 days and 100 rules) is acceptable for the implementation of the reasoner in a real-time system. Moreover, we can appreciate how the time necessary for completing the reasoning by considering simple profiles (1 or 10 rules) remains almost constant, independently of the number of the considered days. Secondly, we may notice a significant increment of the computational time when complex profiles are used.

The second aspect we wanted to highlight is the throughput of the system. The graph shown on the right in Fig. 8 reports the trend of the throughput of the system with an increasing number of rules. On the x-axis, we reported the number of days for which user's data are provided, while on the y-axis, we reported the number of users per-minute that the system is able to process. By performing reasoning operation on single-day data, the throughput is acceptable also for elaborating complex profiles. Indeed, on average the system is able to process approx. 456 user/minutes in the case that all of them are associated with profiles with 100 rules. On the contrary, by increasing the number of considered days, the support for a real-time reasoning starts to become unfeasible. Indeed, a throughput of less than 100 users per minute could be critical in a crowded scenario. However, this throughput can be arbitrarily increased using multiple machines to cope with large scale deployments.

7.3 Usability Evaluation of Experts Tool

The usability of the facilities used by the domain experts for building the monitoring rules has been evaluated through the System Usability Scale (SUS), analyzing the intuitiveness and simplicity of the interfaces. The evaluation protocol consisted in a multiple use sessions and followed the five steps below:

1. Training meetings with the experts involved in the evaluation for an introductory explanation of the functionalities available in the tool.
2. Definition of the first bunch of the Mediterranean Diet rules (95 rules).

[5] Testing machine: 8 CPUs Intel Core i7 870 2.93GHz, 16GB RAM, 1TB HDD.

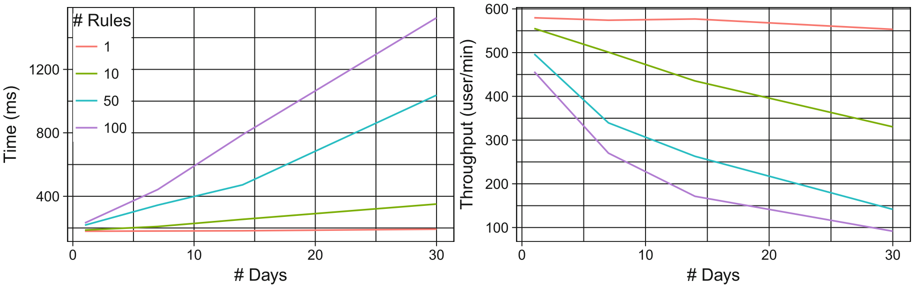


Fig. 8. Time (left) and throughput (right) of the reasoner on the scalability test.

3. Meetings with the experts for collecting questions about functionalities. Refinement of the interface integrating some improvements requested by the experts after the first four days of usage were deployed.
4. Definition of the second bunch of the Mediterranean Diet rules (126 rules)
5. Final meetings with the experts and distribution of the evaluation questionnaires.

The usability test of the tool involved six domain experts participated to the setup of the PerKApp platform for the *Key to Health* project. According to the usability test requirements provided by [17], the number of users involved in the test granted the discovery of 91% of the usability problems. The average score obtained from the SUS was 81.5, that, according to the adjective rating scale proposed by [18], corresponds to "excellent". Finally, the experts agreed about the capability of provided operators to cover the definition of rules necessary for supporting the overall monitoring activity of the Mediterranean Diet.

7.4 Lessons Learned

The design and development of the PerKApp platform provided interesting insights related to the challenges that we needed to address in our experience.

Concerning Challenge C1, we needed to address the lack of detailed structured data about both the food and physical activity domains. In our experience, we noticed that available resources for the food domain were not granular enough. While, concerning the physical activity domain, structured resource representing, not only an activities classification schema, but also other detailed information like the energy consumption and the effort level, were not available. Thus, despite the number of resources available in the Semantic Web community, our scenario demonstrated the necessity of further effort for the construction of resources with a level of detail useful for supporting the development of smart applications.

Regarding Challenge C2, we validated and discussed earlier in this section the effectiveness of the reasoning process implemented into the PerKApp platform. Our results derived from the optimization of rules design and rules evaluation schedule. In a first stage, we designed few complex rules for covering all monitoring activities. On the one hand, we were able to cover several constraints with one rule. But, on the other hand, the computational time required for evaluating these rules was high. Hence, in a second stage, we opted for splitting the rules in more simpler ones and, at the same time, to schedule their evaluation depending on their timing property. This strategy led to an improvement of the overall reasoning performance and allowed us to have an easier control on the overall reasoning process (exactness of the `Violation` instances, debugging operations, etc.). In the scenario addressed by the current deployment of PerKApp, reasoning operations are performed on sets of triple describing users' specific events. Part of the future work will be the investigation of stream reasoning strategies when it is necessary to monitor a continuous flow of information as well as to exploit learning strategies for suggesting, if possible, new rules or adaptations of existing ones. An example within the health domain is the real-time monitoring of the glycemic index.

Concerning Challenge C3, we observed the platform's behavior after the integration of an open source solution, namely RDF4J[6], for supporting data storage. The performance analysis highlighted that this solution was feasible for a real-world scenario. In particular, we observed the capability of managing the parallel data management operations in an efficient way. Thus, we may state that the engine implemented into RDF4J satisfies the scalability requirements of the PerKApp platform. Even if this finding sounds trivial, we believe that it is important to mention this result as a feasibility proof concerning the integration of open source solutions into real-world environments.

Finally, we mentioned in the Challenge C4 the necessity of supporting experts in specifying and revising users' prescriptions through the use of a collaborative tool. One of the potential issues is that the collaborative definition of these constraints may lead to possible disagreements between experts. In order to evaluate if the risk of having these disagreements truly exists, we built a set of 4 scenarios from the food consumption data collected in our triplestore, and we sent them to a group of 12 experts that were not involved in the setup of the PerKApp platform[7]. For each scenario, we proposed a list of possible violations based on the prescriptions modeled into the PerKApp platform and we asked them to confirm if such violations were correct or not. By analyzing the provided responses[8], we can observe that a full agreement was never reached. From this result, we may infer that a tool for supporting experts in the rule definition activity is necessary for trying to limit the disagreement between them. The necessity of reaching an agreement between experts is clear when they work for the same organization: if a healthcare organization has to monitor and support patients, the implemented guidelines should not be expert-dependent. Thus, possible disagreement between them has to be reconciled.

[6] http://rdf4j.org/.

[7] The selected experts are nutritionists of the Trento and Milan healthcare districts.

[8] https://perkapp.fbk.eu/downloads/ISWC2018_inuse_open_issues_dieta_mediterranea.csv.zip

8 Conclusions

In this paper, we presented the PerKApp platform and its key component, i.e., the rule-based reasoner adopted for monitoring users' behaviors in order to support the promotion of health lifestyles. We discussed the role of the knowledge base within the PerKApp platform and how it is equipped with monitoring rules inserted by domain experts through an easy-to-use interface integrated as an extension of the MoKi tool. Then, through a running example, we described how the reasoner operates on the data provided by the users, and we evaluated the reasoner performances by discussing: (i) the time required for processing users' data under different settings; (ii) the throughput of the system with the aim of inferring possible hardware requirements for deploying the PerKApp platform into different environments; and (iii) how the complexity of the users' profiles (i.e. the number of monitoring rules associated with each user) affects the overall efficiency of the system. Results demonstrated the possibility of adopting the system in real-world scenarios, and the reported lessons learned provide insights for future developments, in order to improve the overall efficiency, and thus allowing the deployment of the PerKApp platform in more challenging environments.

References

1. Intille, S.S.: Ubiquitous computing technology for just-in-time motivation of behavior change. In: UbiHealth 2003: 2nd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications (2003)
2. Dragoni, M., Bailoni, T., Eccher, C., Guerini, M., Maimone, R.: A semantic-enabled platform for supporting healthy lifestyles. In: Seffah, A., Penzenstadler, B., Alves, C., Peng, X., (eds.) Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, 3–7 April 2017, pp. 315–322. ACM (2017)
3. Izumi, S., Kuriyama, D., Itabashi, G., Togashi, A., Kato, Y., Takahashi, K.: An ontology-based advice system for health and exercise. In: IMSA, pp. 95–100 (2006)
4. Nassabi, M.H., op den Akker, H., Vollenbroek-Hutten, M.M.R.: An ontology-based recommender system to promote physical activity for pre-frail elderly. In: Mensch and Computer Workshopband, pp. 181–184 (2014)
5. Benmimoune, L., Hajjam, A., Ghodous, P., Andres, E., Talha, S., Hajjam, M.: Ontology-based medical decision support system to enhance chronic patients' lifestyle within e-care telemonitoring platform. Stud. Health Technol. Inform. **213**, 279–282 (2015)
6. Mikolajczak, S., Ruette, T., Tsiporkova, E., Angelova, M.,V.,B.: A semantic reasoning engine for lifestyle profiling in support of personalised coaching. In: International Conference on Global Health Challenges, pp. 79–83 (2015)
7. Jawaid, H., Latif, K., Mukhtar, H., Ahmad, F., Raza, S.A.: Healthcare data validation and conformance testing approach using rule-based reasoning. In: Yin, X., Ho, K., Zeng, D., Aickelin, U., Zhou, R., Wang, H. (eds.) HIS 2015. LNCS, vol. 9085, pp. 241–246. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19156-0_25
8. Hommersom, A., Lucas, P.J.F.: An Introduction to Knowledge Representation and Reasoning in Healthcare. In: Hommersom, A., Lucas, P.J.F. (eds.) Foundations of Biomedical Knowledge Representation. LNCS (LNAI), vol. 9521, pp. 9–32. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-28007-3_2

9. Mamykina, L., Levine, M.E., Davidson, P.G., Smaldone, A.M., Elhadad, N., Albers, D.J.: Data-driven health management: reasoning about personally generated data in diabetes with information technologies. JAMIA **23**(3), 526–531 (2016)
10. Rospocher, M., Serafini, L.: An ontological framework for decision support. In: Takeda, H., Qu, Y., Mizoguchi, R., Kitamura, Y. (eds.) JIST 2012. LNCS, vol. 7774, pp. 239–254. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37996-3_16
11. Bailoni, T., Dragoni, M., Eccher, C., Guerini, M., Maimone, R.: Healthy lifestyle support: the PerKApp ontology. In: Dragoni, M., Poveda-Villalón, M., Jimenez-Ruiz, E. (eds.) OWLED/ORE -2016. LNCS, vol. 10161, pp. 15–23. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54627-8_2
12. Dragoni, M., Bosca, A., Casu, M., Rexha, A.: Modeling, managing, exposing, and linking ontologies with a wiki-based tool. In: LREC, pp. 1668–1675 (2014)
13. Eccher, C., Ferro, A., Seyfang, A., Rospocher, M., Miksch, S.: Modeling clinical protocols using semantic MediaWiki: the case of the oncocure project. In: Riaño, D. (ed.) K4HelP 2008. LNCS (LNAI), vol. 5626, pp. 42–54. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03262-2_4
14. Corcoglioniti, F., Rospocher, M., Mostarda, M., Amadori, M.: Processing billions of RDF triples on a single machine using streaming and sorting. In: ACM SAC, pp. 368–375 (2015)
15. Dragoni, M., Bailoni, T., Maimone, R., Eccher, C.: HeLiS: an ontology for supporting healthy lifestyles. In: The Semantic Web - ISWC 2018–17th International Semantic Web Conference. Lecture Notes in Computer Science, Monterey, California, USA, 8–12 October 2018. Springer (2018)
16. Di Francescomarino, C., Ghidini, C., Rospocher, M.: Evaluating wiki collaborative features in ontology authoring. IEEE Trans. Knowl. Data Eng. **26**(12), 2997–3011 (2014)
17. Nielsen, J., Landauer, T.K.: A mathematical model of the finding of usability problems. In: Proceedings of ACM INTERCHI 1993 Conference, pp. 24–29 (1993)
18. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: adding an adjective rating scale. J. Usability Stud. **4**, 114–123 (2009)