A Hypothetical Reasoning System for Mobile Health and Wellness Applications

Aniello Minutolo(✉), Massimo Esposito, and Giuseppe De Pietro

Institute for High Performance Computing and Networking, ICAR-CNR, Via P. Castellino, 111, 80131 Naples, Italy
{aniello.minutolo,massimo.esposito, giuseppe.depietro}@icar.cnr.it

Abstract. In the last years, rule-based systems have been used in mobile health and wellness applications for embedding and reasoning over domain-specific knowledge and suggesting actions to perform. However, often, no sufficient information is available to infer definite indications about the action to perform and one or more hypothesis should be formulated and evaluated with respect to their possible impacts. In order to face this issue, this paper proposes a mobile hypothetical reasoning system able to evaluate set of hypotheses, infer their outcomes and support the user in choosing the best one. In particular, it offers facilities to: (i) build specific scenarios starting from different initial hypothesis formulated by the user; (ii) optimize them by eliminating common domain-specific elements and avoiding their processing more than once; (iii) efficiently evaluate a set of logic rules over the optimized scenarios directly on the mobile devices and infer the logical consequences by providing timely responses and limiting the consumption of their resources. A case study has been arranged in order to evaluate the system's effectiveness within a mobile application for managing personal diets according to daily caloric needs.

Keywords: Hypothetical reasoning · Rule-based systems · Mobile health and wellness applications

1 Introduction

In the last years, the growing availability of mobile phones and wearable devices has enabled the development of new mobile health and wellness applications able to continuously support individuals anywhere and anytime. These applications are often designed on the top of rule-based systems, since these latter enable the reproduction of explicit deductive reasoning mechanisms on the basis of the explicit formalization of logic production rules built on the top of domain-specific knowledge.

A production rule is usually made of a conjunction of condition elements to satisfy, in its left-hand side (LHS), and a set of action elements in its right-hand side (RHS), respectively. A rule-based system checks whether LHSs hold against a knowledge base including a representation of domain-specific elements, named facts, and, in case the LHS of a rule is satisfied, the corresponding RHS of the rule is inferred. Thanks to these

P. Perego et al. (Eds.): MobiHealth 2016, LNICST 192, pp. 278–286, 2017.
DOI: 10.1007/978-3-319-58877-3_36

capabilities of embedding and reasoning over domain-specific knowledge, rule-based systems have been profitably used in mobile health and wellness applications to suggest actions to perform with the aim of enhancing quality of care, improving adherence to therapies and supporting wellness and healthy lifestyles.

However, often, missing knowledge can arise and rule-based systems embedded in mobile applications are not able to suggest a specific indication to follow or action to perform. Indeed, no sufficient domain-specific elements are available to prove truth or falsity of the condition elements of their rules and, thus, to infer definite suggestions. In such cases, the user is directly asked for making a decision, even though he/she is not able to formulate it based on the available knowledge. To address this issue, rule-based systems should support a form of hypothetical reasoning, where, first, one or more assumptions are made on missing information, then they are processed to draw a set of outcomes, and, finally, these latter are evaluated with respect to their possible impacts in order to support the user in his/her final choice. To the best of our knowledge, none of the existing approaches has been designed by considering the specific requirements of mobile health and wellness applications and, thus, they may result inadequate or intractable on mobile devices due to their limited resources. Indeed, the generation and evaluation of one or more hypothesis could be computationally intensive to be performed directly on mobile devices, with the risk of exhausting their computing and memorization resources, if not properly optimized.

Starting from these considerations, this paper proposes a mobile hypothetical reasoning system able to evaluate set of hypotheses, with the assumption that they must be alternatives among them, directly on mobile devices. In particular, it offers facilities to: (i) build specific scenarios starting from different initial hypothesis formulated by the user; (ii) optimize them by eliminating common domain-specific elements and avoiding their processing more than once; (iii) efficiently evaluate a set of logic rules over the optimized scenarios on the mobile devices and infer the logical consequences by providing timely responses and limiting the consumption of their resources. A case study has been arranged in order to evaluate the system's effectiveness within a mobile wellness application for managing personal diets according to daily caloric needs.

In the following, Sect. 2 introduces background and related work. In Sect. 3, the mobile hypothetical reasoning system is presented, whereas the case of study is described in Sect. 4. Finally, Sect. 5 concludes the work.

2 Background and Related Work

Hypothetical reasoning is a form of reasoning that considers the possibility of uncertain or incomplete knowledge. A typical hypothetical reasoning system tries to explain an observation by dividing the knowledge base into many possible assumptions, in order to provide evidence against hypotheses by testing their logical consequences [1, 2]. Existing approaches to hypothetical reasoning are mainly designed as Truth Maintenance Systems (TMSs). In detail, given an observation and a set of assumptions, a TMS is able to perform: (i) the evaluation of the coexistence and contradictions of assumptions; (ii) the retraction of not viable assumptions; (iii) the preservation of existing dependencies among the valid assumptions and the inferred beliefs. A pioneer hypothetical reasoning system is presented in [3], which consists into a TMS able to evaluate different assumptions and work as a cache by storing all inferences (justifications) made on the basis of the defined assumptions.

An Assumption-based TMS is proposed in [4, 5], which is able to manipulate both justifications and assumptions, i.e. each belief is labeled with the set of assumptions under which it holds, besides the justifications that support it. Some recent works have been focused on the automatic generation of hypotheses when a piece of knowledge is missing, on the basis of preconfigured admitted assumptions. These approaches typically demand a lot of computational resources spent in combinatorial calculations for evaluating conflicts and overlays in computed scenarios [6, 7].

Summarizing, none of these existing approaches is specifically thought to be executed directly on mobile devices, and even if a desktop-oriented TMS could be adapted for being applied to mobile health and wellness scenarios, it might soon become intractable due to the great amount of information stored and maintained. Moreover, this adaption may represent a useless waste of computational and memorization resources on mobile devices, since the possible hypotheses to be considered in the scenarios of interest are mainly alternative among them, and, thus, the complex management of coexisting hypotheses is not required.

All these considerations represent the rationale for the proposed hypothetical reasoning system, which is diffusely described in the next section.

3 The Proposed Hypothetical Reasoning System

The main components of the proposed hypothetical reasoning system, organized according to a typical layered software architecture, are shown in Fig. 1.

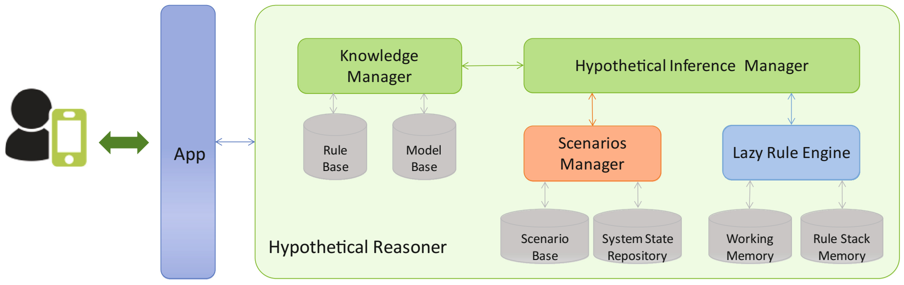


Fig. 1. The main components of the proposed reasoning system.

In detail, the **Knowledge Manager** (KM) and the **Hypothetical Inference Manager** (HIM) are the main interfaces between the reasoning system and other application components. The HIM manages the hypothetical reasoning cycle and it is in charge of invoking other components in order to ensure the correct flow of inference execution, the proper knowledge updating, and the notification of inference outcomes to external components.

The KM is responsible of handling knowledge repositories containing the domain knowledge and the procedural rule set encoding the behavior of the application. In this respect, the **Model Base** (MB) is the repository containing the terminological knowledge, which describes the specific domain in terms of classes and properties. It usually contains the significant assertional knowledge of the domain, i.e. collections of facts, encoded as individuals (instances of concepts) with the corresponding instances of properties. All the information stored in the MB is codified in the N-Triples serialization of OWL language [8], in the form of collections of subject, predicate and object elements. This solution ensures decidability over expressive power and offers a lightweight format for ontologies, which results easier to parse and process in mobile applications [9]. The **Rule Base** (RB) is the repository where production rules are stored. To this aim, a subset of the Jena rule language [10] is used, since it is suitable for being parsed on resource-limited settings, and its predicates include classes and properties encoded as N-Triples. The **Scenarios Manager** (SM) is in charge of defining and evaluating the hypotheses created by the user. In detail, given the ontology **O** describing domain-specific knowledge, and denoted with **R** the set of production rules built on top of the ontology **O**, the couple **K** = (**O**, **R**) can be referred as the knowledge base of the system. When a reasoning process is performed on **K**, the set **Th**(**K**) of logical consequences generated on the basis of **K** can be eventually used for updating the ontology **O** according to the executed rules. Given the set **C** of concepts defined in **O**, and the set **I** of possible instances of **C**, a hypothesis **h** built on the basis of **I** is an unordered list of *n* elements of **I**, where *n* is a positive integer, such that **h** ∪ **O** is semantically consistent.

Note that, given *n* hypotheses $h_1, \ldots, h_n$ defined by the user, they represent alternative actions that he/she wants to evaluate for determining the best one to choose according to the possible inference outcomes produced by the system. Starting from them, a set of *n hypothetical scenarios* $S(h_1), \ldots, S(h_n)$ are built on the basis of the domain knowledge and must be submitted to the HIM for evaluating their logical consequences. However, since the hypothetical scenarios can be determined as $\mathrm{S}(h_1) = (h_1 \cup O, R) \ldots S(h_n) = (h_n \cup O, R)$, they can often share common domain knowledge that will be evaluated repeatedly when more scenarios are submitted to the HIM. To face this issue, the SM is also in charge of modifying these scenarios, before submitting them to the HIM, with the goal of reducing the amount of information to be processed during their evaluation. In detail, all the knowledge characterizing the hypothetical scenarios are processed for determining the common elements, which can be defined as follows:

$$\mathrm{O}_{\mathbf{c}}[\mathrm{S}(\mathrm{h}_1), \ldots, \mathrm{S}(\mathrm{h}_n)] = \bigcap_{i=1}^{\mathrm{n}} [\mathrm{h}_\mathrm{i} \cup \mathrm{O}] = \mathrm{O} \cup \left[\bigcap_{i=1}^{\mathrm{n}} \mathrm{h}_\mathrm{i}\right]$$

In this way, a hypothetical "base scenario" $Sc(h_1, \ldots, h_n) = [O_c(h_1, \ldots, h_n), R]$ is automatically determined, representing the common knowledge elements shared among the *n* hypotheses $h_1, \ldots, h_n$ defined by the user. Such a knowledge will produce the same consequences in all the hypotheses and, thus, should be processed only once for reducing the computational and memorization resources involved during the evaluation of the hypothetical scenarios. In fact, the distinctive knowledge drawn by the scenarios can be determined by re-using the results of the **Th**[$Sc(h_1, \dots, h_n)$]:

$$\mathbf{Th}\left[S(h_1)\right] = \mathbf{Th}\left[Sc(h_1, \dots, h_n)\right] \cup \mathbf{Th}\left[S'(h_1)\right] \dots$$
$$\mathbf{Th}\left[S(h_n)\right] = \mathbf{Th}\left[Sc(h_1, \dots, h_n)\right] \cup \mathbf{Th}\left[S'(h_n)\right]$$

where the sets $S'(h_1), \dots, S'(h_n)$ are composed as follows:

$$S'\left(h_1\right) = [h_1 \bigcup O - O_c\left(h_1, \dots, h_n\right), R] \dots$$
$$S'\left(h_n\right) = [h_n \bigcup O - O_c\left(h_1, \dots, h_n\right), R]$$

Finally, the $n + 1$ scenarios $[Sc(h_1, \dots, h_n), S'(h_1), \dots, S'(h_n)]$ are stored in the repository named **Scenario Base** (SB) and, successively, submitted to the HIM for their evaluation. In this respect, the HIM repeatedly configures and invokes the **Lazy Rule Engine** (LRE), which is based on a lazy pattern matching algorithm, proposed by the authors in [9] and specifically designed and implemented as a light-weight solution suitable for resource-limited mobile devices. By exploiting this algorithm, the LRE is able to provide timely responses without maintaining complex memory structures for processing all the rules contained in the RB. In detail, the LRE evaluates all the rules stored in the RB with respect to the domain knowledge elements characterizing each hypothetical scenario, which are maintained and updated into the **Working Memory** (WM). The current state of this rule evaluation process is maintained into the **Rule Stack Memory** (RSM), in order to enable the possibility of pausing and resuming the search for rule instances that have to be still assessed and eventually executed. At each reasoning cycle, the content of both WM and RSM represents the system's state and is stored into the **System State Repository** (SSR). In particular, when the common scenario $Sc\left(h_1, \dots, h_n\right)$ is evaluated, the LRE generates the logical consequences drawn from it, i.e. Output($Sc(h_1, \dots, h_n)$), and stores it into the SSR. Successively, when the other n scenarios $[S'(h_1), \dots, S'(h_n)]$ are considered, Output($Sc(h_1, \dots, h_n)$) is resumed from SSR, without being recalculated, and is used to incrementally generate the Output ($S'(h_1)$), …, Output($S'(h_n)$).

Finally, at the end of the scenarios' evaluation, the outcomes generated are returned to the application and presented to the user, with the goal of enabling the user to select the single hypothesis h_{best} to make all its outcomes persistent in the MB.

Summarizing, the global reasoning scheme is the following:

```
Load $Sc(h_1,..,h_n)$;
Reason;
Output($Sc(h_1,..,h_n)$) = SystemState;
For $i=1..n$ do
    SystemState = Output($Sc(h_1,..,h_n)$);
    SystemState = SystemState + $S'(h_i)$;
    Reason;
    Output($S'(h_i)$) = SystemState;
End;
Input($h_{best}$)
SystemState = Output(S'($h_{best}$))
```

4 Case of Study

As a proof of concept, the presented reasoning system has been implemented for mobile devices equipped with the Android platform and embedded within a mobile application for monitoring and managing the personal diet according to daily caloric needs. In detail, the application supports users to monitor the food portions consumed during a meal, over a week of observation and alert him/her when potential abnormal situations are detected (e.g. inadequate or excessive consumption of aliments, with respect to the caloric need of the user). For each aliment, a set of diet recommendations has been formulated describing the right portion of food for a meal, for a day, and for a week. Right portions are distilled in terms of minimum and maximum quantities that are recommended and, respectively, forbidden to consume for a given period.

The domain knowledge describing the health status of the user and stored in the MB has been formalized by means of the ontology model outlined in Fig. 2.

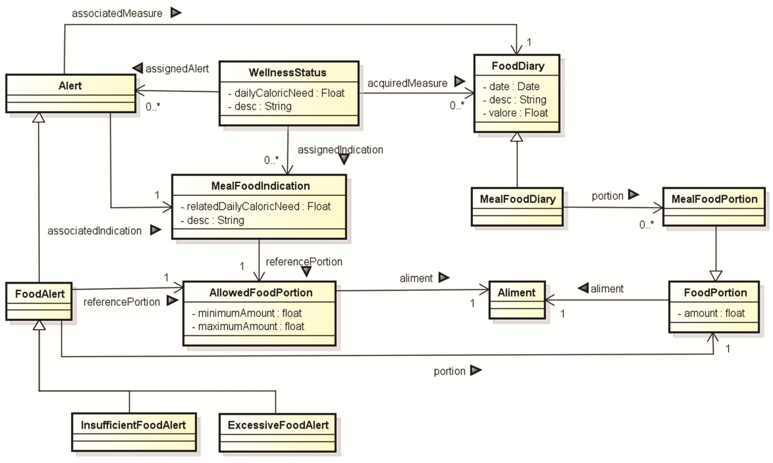
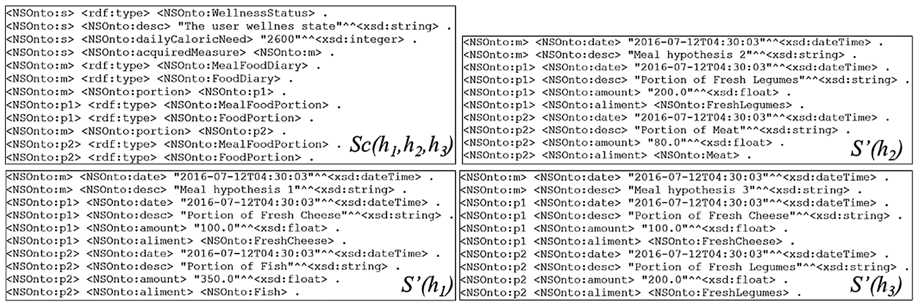


Fig. 2. The ontology model arranged for the case of study considered.

On top of this ontology model, which is diffusely described in [11], a collection of logic rules has been formulated and inserted into the RB of the system. Such rules are aimed at: (i) evaluating the user's daily caloric needs in order to determine the most pertaining diet recommendations; (ii) comparing the selected recommendations with the health status of the user for detecting abnormal food portions; (iii) generating alerts in accordance with the abnormal consumptions detected.

The case of study here considered for this application foresees three possible hypotheses of meals with respect to a personal diet, that are formulated by the user. The hypothesis h_1 consists into a wellness state *s* composed of 2600 calories as daily caloric needs, and a meal food diary *m* containing a portion *p1* of 100 g of fresh cheese, and a portion *p2* of 350 g of fish. The hypothesis h_2 consists into a wellness state *s* composed of 2600 calories as daily caloric needs, and a meal food diary *m* containing a portion *p1* of 200 g of fresh legumes, and a portion *p2* of 80 g of meat. Finally, the hypothesis h_3 consists into a wellness state *s* composed of 2600 calories as daily caloric needs, and a meal food diary *m* containing a portion *p1* of 100 g of fresh cheese, and a portion *p2* of 200 g of fresh legumes.

These hypotheses are submitted to the HIM that interacts with the SM to build three hypothetical scenarios $S(h_1)$, $S(h_2)$, $S(h_3)$ on the top of them by exploiting the ontology model above mentioned. Successively, the SM determines the common knowledge $O_c[S(h_1), S(h_2), S(h_3)]$ among them by calculating the intersection among the scenarios $S(h_1)$, $S(h_2)$, $S(h_3)$ previously computed. In particular, it is worth noting that these hypotheses differ for the type of aliment consumed, and for the number of grams consumed, while the knowledge about the caloric need, and the structure of the objects containing the food portions, is mainly unchanged among them. In detail, with respect to a set of 22 total assertions formulated for encoding the hypotheses h_1, h_2, h_3, a set of 12 assertions has resulted to be shared among them, as reported in Fig. 3.



```
<NSOnto:s> <rdf:type> <NSOnto:WellnessStatus> .
<NSOnto:s> <NSOnto:desc> "The user wellnes state"^^<xsd:string> .
<NSOnto:s> <NSOnto:dailyCaloricNeed> "2600"^^<xsd:integer> .
<NSOnto:s> <NSOnto:acquiredMeasure> <NSOnto:m> .
<NSOnto:m> <rdf:type> <NSOnto:MealFoodDiary> .
<NSOnto:m> <rdf:type> <NSOnto:FoodDiary> .
<NSOnto:m> <NSOnto:portion> <NSOnto:p1> .
<NSOnto:p1> <rdf:type> <NSOnto:MealFoodPortion> .
<NSOnto:p1> <rdf:type> <NSOnto:FoodPortion> .
<NSOnto:m> <NSOnto:portion> <NSOnto:p2> .
<NSOnto:p2> <rdf:type> <NSOnto:MealFoodPortion> .
<NSOnto:p2> <rdf:type> <NSOnto:FoodPortion> .
```
$Sc(h_1,h_2,h_3)$

```
<NSOnto:m> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:m> <NSOnto:desc> "Meal hypothesis 2"^^<xsd:string> .
<NSOnto:p1> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:p1> <NSOnto:desc> "Portion of Fresh Legumes"^^<xsd:string> .
<NSOnto:p1> <NSOnto:amount> "200.0"^^<xsd:float> .
<NSOnto:p1> <NSOnto:aliment> <NSOnto:FreshLegumes> .
<NSOnto:p2> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:p2> <NSOnto:desc> "Portion of Meat"^^<xsd:string> .
<NSOnto:p2> <NSOnto:amount> "80.0"^^<xsd:float> .
<NSOnto:p2> <NSOnto:aliment> <NSOnto:Meat> .
```
$S'(h_2)$

```
<NSOnto:m> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:m> <NSOnto:desc> "Meal hypothesis 1"^^<xsd:string> .
<NSOnto:p1> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:p1> <NSOnto:desc> "Portion of Fresh Cheese"^^<xsd:string> .
<NSOnto:p1> <NSOnto:amount> "100.0"^^<xsd:float> .
<NSOnto:p1> <NSOnto:aliment> <NSOnto:FreshCheese> .
<NSOnto:p2> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:p2> <NSOnto:desc> "Portion of Fish"^^<xsd:string> .
<NSOnto:p2> <NSOnto:amount> "350.0"^^<xsd:float> .
<NSOnto:p2> <NSOnto:aliment> <NSOnto:Fish> .
```
$S'(h_1)$

```
<NSOnto:m> <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:m> <NSOnto:desc> "Meal hypothesis 3"^^<xsd:string> .
<NSOnto:p1 <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:p1 <NSOnto:desc> "Portion of Fresh Cheese"^^<xsd:string> .
<NSOnto:p1 <NSOnto:amount> "100.0"^^<xsd:float> .
<NSOnto:p1 <NSOnto:aliment> <NSOnto:FreshCheese> .
<NSOnto:p2 <NSOnto:date> "2016-07-12T04:30:03"^^<xsd:dateTime> .
<NSOnto:p2 <NSOnto:desc> "Portion of Fresh Legumes"^^<xsd:string> .
<NSOnto:p2 <NSOnto:amount> "200.0"^^<xsd:float> .
<NSOnto:p2 <NSOnto:aliment> <NSOnto:FreshLegumes> .
```
$S'(h_3)$

Fig. 3. The scenarios for evaluating the consequences of the meal hypotheses h_1, h_2, h_3.

Thus, denoted with N_c the average number of condition elements in the rules, and denoted with $card(S(h_i))$ the cardinality of the ontology model associated to the scenario $S(h_i)$, the numbers $KB[S(h_1), S(h_2), S(h_3)]$ and $KB[S'(h_1), S'(h_2), S'(h_3)]$ of comparisons required for evaluating the hypotheses h_1, h_2, h_3, before and after the optimization performed by the SM, can be approximately estimated as follows:

$$KB\left[S(h_1),\ldots,S(h_3)\right] = n * card(O) * \left[card(S(h_1)) + \ldots + card(S(h_n))\right]$$
$$* card(R) * N_c = 95 * 10^6$$
$$KB\left[S'(h_1),\ldots,S'(h_3)\right] = \left[card(O) + card(O_c) + card(S'(h_1)) + \ldots + card(S'(h_n))\right]$$
$$* card(R) * N_c = 490 * 10^3$$
where
$$n = 3, card(O) = 2000, card(S(h_i)) = 22, card(O_c) = 12, card(S'(h_i)) = 10, card(R) = 20, N_c = 12$$

As a consequence, the proposed approach enables to drastically reduce the amount of comparisons that are required for comparing rules with the ontology assertions, since all the shared information are computed only once, and properly re-used during the evaluation of each hypothetical scenario. Successively, after the elaboration performed by the SM, the actual scenarios are submitted to the HIM, which repeatedly configures and invokes the LRE for collecting the inference outcomes drawn on the basis of them. In particular, two of the submitted hypotheses generate an excessive food alert, whereas the last one generates no alarm. Thus, the application reports this result to the user, who is enabled to choose the third hypothesis as actual meal to consume.

5 Conclusions

This paper proposes a mobile hypothetical reasoning system able to evaluate set of hypotheses, made on missing information and with the assumption that they must be alternatives among them, infer their outcomes and assess the possible impacts in order to support the user in choosing the best one. Differently from existing solutions, the proposed system is not aimed at automatically generating hypotheses to be validated with respect to specific scenarios and detecting eventual conflicts and overlays. Indeed, the novelty of this system mainly consists into its facilities to: (i) build specific scenarios starting from different initial hypothesis formulated by the user; (ii) optimize them by eliminating common domain-specific elements and avoiding their processing more than once; (iii) efficiently evaluate a set of logic rules over the optimized scenarios directly on the mobile devices and infer the logical consequences by providing timely responses and limiting the consumption of their resources.

A case study has been arranged in order to evaluate the system's effectiveness within a mobile wellness application for managing personal diets according to daily caloric needs, showing its capability of efficiently evaluating scenarios determined on the basis of the hypotheses formulated from the user, as well as reducing the amount of data to process singularly, with respect to the number of comparisons required.

Next step of the research activities will be, on the one hand, to define a more formal and mathematical background to describe the proposed hypothetical reasoning scheme and, on the other hand, to minutely evaluate the performance of the proposed system in terms of computation and memorization resources required when executed in several scenarios and subjected to different load situations.

Acknowledgment. This work has been partially supported by the Italian project "eHealthnet" funded by the Italian Ministry of Education, University, and Research.

References

1. Poole, D., Goebel, R., Aleliunas, R.: Theorist: a logical reasoning system for defaults and diagnosis. In: Cercone, N., McCalla, G. (eds.) Knowledge Frontier: Essays in the Representation of Knowledge, pp. 331–352, Springer, New York (1987)
2. Poole, D.: A logical framework for default reasoning. Artif. Intell. **36**(1), 27–47 (1988)
3. Doyle, J.: A truth maintenance system. Artif. Intell. **12**(3), 231–272 (1979)
4. de Kleer, J.: An assumption-based TMS. Artif. Intell. **28**(2), 127–162 (1986)
5. de Kleer, J.: Extending the ATMS. Artif. Intell. **28**(2), 163–196 (1986)
6. Giannikis, G.K., Daskalopulu, A.: Assumption-based reasoning in dynamic normative agent systems. Web Intell. Agent Syst. **8**(4), 343–362 (2010)
7. Tahara, I.: Computing scenario from knowledge with preferentially ordered hypotheses. Syst. Comput. Jpn. **35**(4), 19–26 (2004)
8. Patel-Schneider, P., Hayes, P., Horrocks, I., et al.: OWL web ontology language semantics and abstract syntax. W3C Recommendation, 10 (2004). http://www.w3.org/TR/owl-semantics/
9. Minutolo, A., Esposito, E., De Pietro, G.: Design and validation of a light-weight reasoning system to support remote health monitoring applications. Eng. Appl. Artif. Intell. **41**, 232–248 (2015)
10. Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track, New York, USA, pp. 74–83 (2004)
11. Minutolo, A., Esposito, M., Pietro, G.: An ontology-based approach for representing medical recommendations in mHealth applications. In: Chen, Y.-W., Tanaka, S., Howlett, Robert J., Jain, Lakhmi C. (eds.) Innovation in Medicine and Healthcare 2016. SIST, vol. 60, pp. 171–182. Springer, Cham (2016). doi:10.1007/978-3-319-39687-3_17