ABSTRACT
This paper presents a health advice system using an ontology and inference rules. The ontology consists of the various concepts and their relationships about health and exercise. By combining the ontology and inference rules to derive health advices, recommended and/or non-recommended exercises are inferred and presented to the users.

KEY WORDS
Ontology, Inference, Rule, Health, Support system, OWL.

1 Introduction
Now, many countries are turning gray very quickly. Especially, Japan is rapidly super-aging. The ratio of the elderly will reach about 30% until 2020. On the other hand, 40% of the Japanese medical cost is used for the elderly. Moreover, the lifestyle related diseases are the biggest killer of Japanese. Therefore, a computer-supported health advice system is desired as one of the solutions for preventing lifestyle diseases, in Japan.

In this paper, we present a support system for the high level health advices that is one of the projects of Ministry of Internal Affairs and Communications in Japan. In this system, the health condition of a user is checked using a mobile device and a sensor device. And health advices are provided for the user based on his/her health condition. Especially, a recommended exercise is presented to the user as the health advice, according to the user’s goal and health condition. To express the concepts of health and exercise and the relationship between them, we use the idea of an ontology. By using the ontology, a machine can understand the semantics of the concepts and their relationships. For the expression of the ontology, RDF (Resource Description Framework) [1], RDF Schema (RDF Vocabulary Description Language) [2] and OWL (Web Ontology Language) [3] are used. These are the standard languages proposed by W3C (World Wide Web Consortium).

To provide health advices, we use inference. By using inference, a new relationship on the ontology is derived. We equip inference rules to derive a recommended exercise and a non-recommended exercise as the health advices.

For description of the ontology and implementation of an inference engine, we use some existing tools. To describe the ontology, we use Protégé [5] which is an ontology editor. To implement the inference engine, we combine Protégé, Jess [6], JessTab [7] and Racer [8]. Jess is an inference engine written in Java language. JessTab is a plug-in of Protégé that plays a role of a bridge between Protégé and Jess. Racer classifies OWL individuals described by Protégé based on the definition of OWL classes.

This paper is structured as follows. The support system is outlined in Section 2. In Section 3, we define the ontology about health and exercise. Section 4 gives the inference rules to provide health advices. Section 5 describes a prototype implementation of the inference engine and some examples. Finally, Section 6 concludes this paper.
RDF Schema and OWL. RDF offers a standard mechanism to describe metadata that can be understood by a machine. RDF provides a data model and a grammar to describe the predicate and the object in its Model and Syntax specification. The data model of RDF consists of three components: a resource, a property and a statement. A statement consists of a subject, a predicate and an object. RDF Schema offers a basic mechanism to define the category of these properties and general resources. A necessary class can be defined by using the basic classes, the basic properties, etc. prepared in the specification of RDF Schema, and its instances and subclasses can be derived. OWL is the language developed by W3C that fills the requirement of ontology sharing, evolution, interoperability and inconsistency detection requested from the Web ontology language.

To provide an advice to a user who has a goal accomplished by an exercise, we define six main concepts. They are shown in Figure 2 in which an oval represents a concept (called a class in OWL) and an arrow represents a relationship (called a property in OWL) between classes.

In this ontology, a user is associated with an exercise to accomplish a goal which the user has. It is assumed the goal conforms to the effect of the exercise. The user is also associated with an exercise which the user must not do due to the health condition of the user.

The class Person represents a set of the users of the support system. The class Exercise represents a set of the exercises which the users do for health. The class EffectOfExercise represents a set of the effects of exercises. The class Goal represents a set of the goals which the users have for health. The class HealthData represents a set of data which the users have. The class HealthAdvice represents a set of concrete health advices which are given to the users. These classes have subclasses which represent their details.

All classes are associated with properties. The property between Person and Exercise is doesExercise which shows that a user does an exercise. The properties hasRecommendedExercise and hasNon-recommendedExercise also associate Person with Exercise. These show that there exist a recommended and/or non-recommended exercise for the user depending on his/her health condition. The property hasEffectOfExercise shows that an exercise has its effect. The property hasHarmfulHealthData shows that an exercise may be harmful due to some health data. Goal and EffectOfExercise are equivalent classes. As mentioned above, we assume the effect of an exercise which a user does is the same as a goal which the user wants to accomplish.

In OWL, an instance of a class is called an individual. The individuals of Person are concrete users of the system. Exercise has “jogging,” “muscular Workout,” “aerobics,” etc. as individuals. EffectOfExercise has “strengtheningMuscles,” “losingWeight,” “improvingCardiopulmonaryFunction,” etc. as individuals. The individuals of Goal are “toStrengthenMuscles,” “toLoseWeight,” “toImproveCardiopulmonaryFunction,” etc. HealthData has data of the users (e.g. age, height, weight, BMI, pulse). HealthAdvice has concrete advices as individuals (e.g. “Go jogging

2 System overview

We have been developing a support system to provide the health advices. An outline of this system is shown in Figure 1.

A wearable sensor device measures the vital data of a user. A mobile device collects the measured vital data and displays these data in real time. The collected vital data with ID and the measurement date in the mobile device is sent to the database. In addition to the vital data, the user inputs the record of exercise and meal. Then the record is sent to the database. When the user requests the data such as the measured vital data, a height, a weight, etc., the user agent gets the requested data from the database, and sends to the mobile device. The mobile device anytime displays these data in the database. This data is used by the ontology agent for inference to derive health advices. When the user requests the health advices, the user agent gets data of the user from the database, and sends it to the ontology agent. The ontology agent derives the health advices based on the data of the user, the ontology and the inference rules. The mobile device can display the health advices. Also, the ontology agent executes simple diagnosis. For example, the ontology agent calculates the BMI from a height and a weight which are kept in the database, and diagnoses that the user is fat or not. In this system, the user can easily get the health advices depending on the health condition no matter when or where by introducing the ontology, the inference rules and the agents.

In this system, user personal data such as vital data is transferred over the network. Therefore, sending and receiving of the transferred data should be high-security. For secure communication, data is encrypted and network traffic is monitored. Therefore, we restrict any unauthorized access and find malevolent users.

3 Ontology about health and exercise

In our system, we use an ontology as a basis to derive health advices. To describe the ontology, we use RDF, RDF Schema and OWL. RDF offers a standard mechanism to describe metadata that can be understood by a machine.
to lose weight").

This ontology explicitly express the relationships between health and exercise.

4 Inference for advice

Based on the ontology explained in the preceding Section, we provide health advices for users. In order to derive health advices, we use inference. By using inference, a new relationship on the ontology is derived.

We present two basic IF-THEN type of inference rules. One is for the recommendation of an appropriate exercise based on the user’s goal, and the other is for the non-recommendation of a harmful exercise based on the user’s health condition.

The basic rule to derive a recommended exercise is as follows.

Basic rule 1

\[
\begin{align*}
& \text{hasGoal}(?p, ?g) \land \text{Person}(?p) \land \text{Goal}(?g) \land \text{owl:sameAs}(?g, ?e) \land \text{EffectOfExercise}(?x, ?e) \land \text{Exercise}(?x) \\
& \implies \text{hasRecommendedExercise}(?p, ?x) & (1) \\
& \text{hasAdvice}(?p, "?x is recommended for ?p.") & (2)
\end{align*}
\]

This syntax conforms to the human readable syntax of SWRL [4]. The left side of \(\implies\) is an antecedent, and the right side is a consequence. An antecedent and a consequence are represented by the conjunction of atoms. Only if all atoms in the antecedent are satisfied, all atoms in the consequence are derived. A word with a question mark is a variable (e.g. ?p, ?g). "\cdots" means an individual or a data literal. A negation cannot be represented in SWRL. But in this rule, we represent a negation by "\sim". If an atom \(A\) is not evaluated as true, \(\sim A\) is evaluated as true (negation as failure).

This basic rule means the following. If a user has a goal (1), an exercise has its effect (3), the goal and the effect are the same (2), and the user does not have the exercise as a non-recommended exercise (4), then the exercise is recommended for the user (5) and the user gets a health advice (6).

The basic rule to derive a non-recommended exercise is the following.

Basic rule 2

\[
\begin{align*}
& \text{hasHealthData}(?p, ?d) \land \text{Person}(?p) \land \text{HealthData}(?d) \land \text{hasHarmfulHealthData}(?x, ?d) \land \text{Exercise}(?x) \\
& \implies \text{hasNon-recommendedExercise}(?p, ?x) & (I) \\
& \text{hasAdvice}(?p, "?x is not recommended for ?p due to ?d.") & (II)
\end{align*}
\]

If a user has a health data (I) and an exercise has the health data as a harmful health data (II), then the exercise is not recommended for the user (III) and the user gets a health advice (IV).

We demonstrate an application example of the Basic rule 1. First, we define the following facts.

Facts

\[
\begin{align*}
& \text{Person("Bob")} & (a) \\
& \text{Exercise("muscularWorkout")} & (b) \\
& \text{Goal("toStrengthenMuscles")} & (c) \\
& \text{EffectOfExercise("strengtheningMuscles")} & (d) \\
& \text{hasGoal("Bob", "toStrengthenMuscles")} & (e) \\
& \text{hasEffectOfExercise("muscularWorkout", "strengtheningMuscles")} & (f) \\
& \text{owl:sameAs("strengtheningMuscles", "toStrengthenMuscles")} & (g)
\end{align*}
\]

They are the rules which have only the consequence. Therefore, they mean facts. In this case, the ontology agent can infer a health advice from the Basic rule 1 and the facts. We show the forward reasoning. In the atoms (1), we set ?p="Bob" and ?g="toStrengthenMuscles." Then these atoms are satisfied from the facts (a), (b) and (c). Setting ?e="strengtheningMuscles," the atoms (2) are satisfied from the facts (d) and (g). Setting ?x="muscularWorkout," the atoms (3) are satisfied from the facts (b) and (f). In this case, the atom hasNon-recommendedExercise("Bob", "toStrengthenMuscles") is false.
atom (III) hasNon-recommendedExercise("Bob", "muscularWorkout") is derived. That is, "Bob" gets the health advice "muscularWorkout is recommended for Bob."

As the second example, we show an application of the Basic rule 2 in addition to the Basic rule 1. Here, we add the following facts.

**Added facts**

\[
\begin{align*}
\text{hasHealthData}("\text{heartDisease}") & \quad \text{(h)} \\
\text{hasHealthData}("\text{Bob}", "\text{heartDisease}") & \quad \text{(i)} \\
\text{hasHarmfulHealthData}("\text{muscularWorkout}", "\text{heartDisease}") & \quad \text{(j)}
\end{align*}
\]

In the application of the Basic rule 2, the atoms (I) are satisfied from the facts (a), (h) and (i), and the atoms (II) are satisfied from the facts (b) and (j). Therefore, the consequence of the Basic rule 2 is derived. The atom (III) hasNon-recommendedExercise("Bob", "muscularWorkout") is derived. That is, "muscularWorkout" is not recommended for "Bob". The consequence is not derived and any advice is not provided.

Next, we show the backward reasoning. We set hasRecommendedExercise("Bob", ?x) (the atom (5)) and hasAdvice("Bob", "?x is recommended for Bob") (the atom (6)). Applying the above goals, the atoms (I) are satisfied under the condition of ?p="Bob" and ?g="toStrengthenMuscles." Applying the facts (d) and (g), the atoms (2) are satisfied from ?e="strengtheningMuscles." Applying the facts (b) and (f), the atoms (3) are satisfied from ?x="muscularWorkout." In this case, the atom (4) is not satisfied because of hasNon-recommendedExercise("Bob", "muscularWorkout"). Therefore, the consequence is not derived and any advice is not provided.

Our inference rules are described based on the ontology, and derive the exercise which has the effect accomplishing the goal of the user. Moreover, our inference rules introduce negation in order to prohibit harmful exercise, because some exercise may be harmful depending on user’s health condition. In this way, we provide the health advices for the user based on the goal and the health condition.

5 Implementation

We are now implementing a prototype of our support system. This section describes an implementation of an inference engine to derive health advices based on the ontology, as part of the whole system.

We use Protégé to describe the ontology explained in Section 3. Protégé is an ontology editor which enables to define classes, properties and individuals easily. These defined contents are used for inference.

The structure of the inference engine is shown in Figure 3. To implement the inference engine, we combine some existing tools such as Protégé, Jess, JessTab and Racer as in a related research [9], and implement the user interface using Java language. Jess is an inference engine for forward and backward reasoning. It is written in Java language. JessTab is a plug-in of Protégé that connects Protégé and Jess. It can convert OWL classes to Jess processable formats. Racer can classify OWL individuals and supplement properties based on the definition of OWL classes.

These tools are all written in Java language and their APIs are opened. We refer them and develop a new user interface using Java. Therefore, we can use these tool functions from the interface. For example, we can input the ontology, facts and inference rules, run the inference and browse the result of the inference. We also develop new functions such as display of the inference process, input of data needed at the inference process, etc.

Below, we demonstrate execution examples using the inference rules and facts explained in Section 4. In this example, we use the forward reasoning.

Figure 4 shows the definition of the ontology about health and exercise explained in Section 3. The left side in this Figure shows the class hierarchy, where the six classes explained in Section 3 are shown. The right side shows the properties which the selected class has. In this case, the six properties of the class Person are shown. We can also define the individuals of each class using Protégé. Figure 5 shows the definition of the individuals of the class Exercise.

Figure 6 shows the user interface of the system. The user selects one of the goals listed at the upper side in the window. Then clicking the “get health advice” button, the user can get a health advice as the result of inference based on the health data given in advance, the ontology and the inference rules.

We show an inference example using the ontology (Figures 4 and 5) and two basic rules. First, the ontology and individuals are input to the working memory of Jess. The individuals defined in Figure 5 are converted into the facts of Jess. And the basic rules are converted into Jess format, and are set into Jess. The Basic rules 1 and 2 are written as follows in Jess format.

**Basic rule 1**

```jess
(defrule BasicRule1
 (and
```


Basic rule 2

(defrule BasicRule2
  (and
   (object (is-a Person) (OBJECT ?p)
     (hasGoal ?g))
   (object (is-a Goal) (OBJECT ?g)
     (owl:sameAs ?e))
   (object (is-a EffectOfExercise) (OBJECT ?e))
   (object (is-a Exercise) (OBJECT ?x)
     (hasEffectOfExercise ?e))
   (not (object (is-a Person) (OBJECT ?p)
     (hasNon-recommendedExercise ?x))))
  =>
  (and
   (slot-set (call ?p getName) hasRecommendedExercise (call ?x getName))
   (printout t (call ?x getName) "is recommended for" (call ?p getName) "." crlf)
   (halt))
)

When a user “Bob” selects the goal “to strengthen muscles,” the result of inference is shown as in Figure 7. The Basic rule 1 is applied, and the user gets the health advice “Muscular workout is recommended for Bob.”

Next, we add the following facts using Jess commands.

Added facts

(make-instance heartDisease of...
We create the “heartDisease” as an individual of the class HealthData and set it to the value of the properties hasHealthData which “Bob” has and hasHarmfulHealthData which “muscularWorkout” has. When the user selects the goal “to strengthen muscles,” the inference result is shown in Figure 8. The Basic rule 1 is not applied, but the Basic rule 2 is applied and the user gets the health advice “Muscular workout is not recommended for Bob due to heart disease.”

By using this inference engine, the user gets the appropriate health advices based on the goal and the health condition.

6 Conclusion

We have been developing a health advice system. To provide an advice to the user who has a goal accomplished by an exercise, we use an ontology about health and exercise. We equip two basic rules to recommend an appropriate exercise based on the goal and not to recommend an exercise based on the health condition. The inference engine is implemented using some existing tools. An appropriate exercise as the health advice can be derived.

Our future work includes a more detailed definition of the ontology, an equipment of inference rules to derive various advices and revision and improvement of the whole system explained in Section 2.

Acknowledgements

This work is financially supported by the Strategic Information and Communications R and D Promotion Programme by the Ministry of Internal Affairs and Communication, Japan.

References