Laboratory Report in Artificial Intelligence
Knowledge-Based Systems with CLIPS

H. Bjørgan, L. Folkestad, M. Klev

Facultat d'Informàtica de Barcelona, Universitat Politècnica de Catalunya, 08034 Barcelona.

Abstract

The second practical exercise in Artificial Intelligence at FIB - UPC. The task was to schedule a detailed training program for elders, taking into account diagnoses, physical shape and events as falls through the use of a knowledge-based system made in CLIPS. Our solution succeeded in providing a training program for elders with specific diagnoses, varying fitness levels as well as falls. However, we did not take into account injuries, cases with multiple diseases nor some specific cases of some diseases. There are also some other minor issues, which we believe would be fairly easy to resolve.

1. Introduction

The aim of the training program will be to help the patients maintain a healthy lifestyle, to prevent the risk of developing illnesses as well as keeping certain diagnoses at bay. In order to create a taylor-made training program, it is necessary to take into account the different levels of fitness among the participants as well as any illnesses, diseases or recent events like falls. Moreover, as mental health plays a major role in our ability to maintain a good physical health, we deem it necessary to incorporate any hobbies or favourite activities into the training program.

1.1. Defining Knowledge-Based Systems

A KBS is defined as a computer program that reasons and uses a knowledge base to solve complex problems [1]. A *knowledge base* is the composition of an *ontology* and a set of individual *instances* of classes. The ontology is a formal explicit description of concepts in a domain of discourse, called *classes*, the properties of each concept describing various features and attributes of the concepts, called *slots*, and restrictions on slots, called *role restrictions* [2].

1.2. Viability of a Knowledge-Based System to the Domain

The domain of the problem is heavily based on expert knowledge; it combines knowledge of general exercise guidelines, illness diagnoses and health among elders in general. Restricted domains of expertise are well suited to being expressed as KBS's because one can define a specific set of rules of how to handle and structure information. Although the problem domain combines different domains of expertise, it can be narrowed down to a fairly restricted domain of expertise. Furthermore, one of the main advantages of a KBS is its ability to make decisions very quickly on large amount of information, as long as it is structured and represented correctly. This is very convenient when dealing with a large number of different diagnoses, exercise types and individual differences in shape. Moreover, the program can very easily be scaled up with more diagnoses and recommended exercises. There are however some difficulties of representing the domain in a KBS. People with several diagnoses, injuries and an medical history may be difficult to represent correctly. Although a KBS in general will work quite swiftly, the performance is in reality dependent on the computational power provided.

1.3. Sources of Knowledge

In order to create the knowledge base on which the KBS is based on, we have thoroughly read and taken notes of the provided sources, *Ejercicio físico para todos los mayores* [3], *Guia de ejercicios para pacientes con enfermedad cardiovascular* [4], *Ejercicio físico es salud* [5], *Guía de ejercicio física para mayores* [6] and *Guía para la Prescripción de Ejercicio Físico en Pacientes con Riesgo Cardiovascular* [7]. After the reading and note-taking process, we tried to extract and structure the most essential information. In essence, this was specific exercises for improving strength, balance and flexibility, cardio activities and exercise recommendations for specific diagnoses as well as general advice from events such as falls. The exercise recommendations for the different diagnoses consisted of specific advice on exercises and how these should be performed. They will be explained in greater detail later in this article.

2. Conceptualizing the Problem

2.1. The Concepts of the Domain

The concepts of the domain can be divided into the following categories:

2.1.1. Anatomy

The anatomy is the description of how the body is made up of different body parts. Knowing the anatomy of the body is important because certain exercises focus on improving strength or flexibility of specific body parts.

2.1.2. Illness

An illness together with a medical diagnosis describes what causes different symptoms and signs of a person. Different illnesses attack different parts of the body, and thus they may require different modes of exercise. Moreover, having knowledge about the illness is important to adjust the quantity and intensity of the sessions to keep the illness at bay.

2.1.3. Training program

A training program is a schedule that consists of sessions in a certain order. The training program thus describes the course of how the sessions are put together.

2.1.4. Session

Every session is either a *strength session*, *cardio session*, *flexibility session* or a *balance session*. Each of these have specific properties.

The **strength session** consists of a set of exercises, The exercises should be done with a certain resistance or intensity and they should be done consecutively a certain number of times, called *repetitions*. One group of consecutive repetitions is called a *set*. Typically every exercise is done in one or more sets.

The **balance session** consists of a set of exercises that are done in sets of repetitions.

The **flexibility session**, like the strength session, consists of a set of exercises. Every exercise is done for a set amount of time, and the session itself has a duration.

The **cardio session** consists of a specific activity with a certain *duration* and *intensity*.

2.1.5. Exercise

An exercise is a specific movement or activity focused on a certain body part. Depending on its nature and intensity, exercises might aim to improve strength, balance or flexibility.

2.1.6. Physical shape

The physical shape of a patient determine how much exercise the patients body can bear in order to have the best progress and effect of the training. The recommendations on quantity and intensity is based on how much restitution, i.e. rest, the body needs to recover from a certain amount of training. We have decided to divide the different states of physical shape into low, medium and high.

2.2. Combining concepts to form a solution

In order to create a training program that is specialized for the patients, one must consider the different concepts not individually but rather in combination. Certain diagnoses may recommend a set amount of exercise, however this recommendation must be adjusted according to the patient's physical shape. Every person with the same diagnosis cannot follow the exact same training program and expect the desired results. Moreover, recent incidents as for example falls may require a patient to work more on balance regardless of diagnosis. Much of the expert knowledge of the domain lies in combining the different concepts correctly.

2.3. Examples of expert knowledge of the domain

- Patients with depression are advised to mainly do cardio exercises, and preferably in groups.
- Patients with recent falls need to focus more on balance than other patients.
- Patients with Rheumatoid arthritis are advised to exercise with higher intensities than e.g. patients with cystic fibrosis.
- Fragile elders are advised to start slowly with short sessions.
- Mental health affects the patients physical health, and therefore any hobbies or favourite activities should be incorporated to the training program.

2.4. The solution process

As mentioned in 1.3, we have collected information by reading the articles and taken notes. After deciding the most important concepts, the *classes*, we tried to decide what components they were built up by, the *attributes*, as well as how the different concepts were connected, the *relations*. Our concepts with attributes and restrictions are explained in further detail in 3.1.

When constructing a training program we had to decide how we would connect the exercise recommendations of the different illnesses. We decided to think of the recommendations as attributes of the specific illnesses.

To recommend a specific session, we had to create different sessions depending on the number of exercises. Moreover, the sessions had to be put together differently depending on the goal of the session. Improvement of strength and flexibility are examples of session goals.

We also had to consider how we would solve the problem of deciding how a session would differ depending on physical shape. We decided to use the upper and lower bounds of the different properties of a session specified for an illness to make a session adapted to the fitness level.

As the mental health is very important, we had to find a way of incorporating hobbies or favourite activities into the training program. We decided that the best way to do this was asking the patient of their hobbies, and make sure that the preferred activity was part of the training plan.

3. Formalization of the solution

3.1. The ontology

Our ontology contains 5 main classes; BodyPart, Exercise, Illness, Session and TrainingProgram. These classes describe all the different concepts in our formalization of the problem.

3.1.1. BodyPart

The BodyPart class represents a physical part of the body and only has one slot, which is a required single slot; BodyPartName. This class is created in order to classify training exercises by the part(s) of the body they affect. The instances of this class represent the actual body parts, and are the following: Abdominal, Ankle, Biceps, Calves, Glutes, Hamstring, Hips, Quadriceps, Shoulders, Triceps, Wrist.

3.1.2. Exercise

The Exercise class is an abstract class that represents a single physical exercise. Its only attribute is Activity, which is the name of the Exercise, and is of the type required single. Exercise has 5 subclasses, namely StrengthExercise, BalanceExercise, FlexibilityExercise, CardioExercise and GeneralExercise. As the names of these classes suggest, they categorize exercises by their type.

StrengthExercise has one additional attribute, BodyPartsTrained, which is a required multifield of instances of BodyPart, with cardinality (1:5). This class represents exercises that are focused on training strength. StrengthExercise is identified by its name (Activity), and the body parts that are affected are listed in the field BodyPartsTrained. In our ontology, we have included 11 instances of StrengthExercise, which are all based on exercises represented in the articles that form the information source of this project. These exercises include KneeFlexion, which trains the hamstring muscles, and ArmLift, which trains the triceps muscles.

BalanceExercise has two additional attributes, SourceExercise and AdditionsForBalance. None of these are required, but SourceExercise has the cardinality (0:1), as where AdditionsForBalance has the cardinality (0:5). The atribute SourceExercise is an instance of StrengthExercise. The reason for this is that many of the balance exercises described in the source articles are simply modifications of a strength exercise. By including this attribute, we can make a logic relation between a balance exercise and its corresponding strength exercise. This leads us to the explanation of the other attribute, AdditionsForBalance. This field consists of Strings, describing the modifications that can be added to the original strength exercise in order to make it a balance exercise. Examples of additions can be performing the exercise with closed eyes, or without holding onto something. By making both of these attributes non-required, it is possible to create independent instances only identified by their name. We have included eight instances of BalanceExercise in our ontology, such as AlternatinglyStandingOnOneLeg, which is a independent exercise only identified by its name, and BalanceHipExtension, which is a modification of the StrengthExercise HipExtension.

FlexibilityExercise has the additional attribute BodyPartsTrained, which consists of one or several instances of BodyPart. It is a required multifield with cardinality (1:5). This field serves the same purpose as it does in StrengthExercise, which is to describe which parts of the body are affected by the exercise. The instances of FlexibilityExercise are identified by their name (Activity). In our ontology we have included ten instances, which again are all based on exercises presented in the source articles. These instances include ShoulderRotation, which stretches the shoulder muscles, and TricepsStretchWithTowel, which stretches the triceps muscles by stretching a towel behind your back.

CardioExercise has no additional attributes, but is simply identified by its name (Activity). The instances of CardioExercise are exercises that train stamina. We have included the following 11 instances of CardioExercise in our ontology; Aerobic dance, Biking, Rowing, Running Skating, Static bike, Walking and Walking stairs.

GeneralExercise has, as CardioExercise, no additional attributes. The reason that we have divided these two classes, is that the instances of GeneralExercise are slower activities, often seen as leisure activities. The only two instances of this class included in our ontology are Golf and Hiking.

3.1.3. Illness

The Illness class represents the different types of illnesses that we have gathered from the source articles. It has the following attributes:

- Name, which is a required single String, simply telling the name of the illness.
- RecommendedCardioDuration, which is of the type required multiple integer, with cardinality (1:2). This attribute describes the recommended duration of a cardio session for a patient with this disease. If it has one value, this is the recommendation, where as if it has two values, these represent the upper and lower bound of the recommended duration.
- RecommendedStrengthIntensity, which is of the type required multiple integer with cardinality (1:2). This class represents an intensity recommendation from the Borg-scale, for a patient with this disease. It has lower bound 6 and upper bound 20, corresponding with levels in the Borg-scale.
- RecommendedCardioIntensity, which is also of the type required multiple integer with cardinality (1:2). This class represents an intensity recommendation from the Borg-scale, for a patient with this disease. It has the same logic as RecommendedCardioDuration with regards to allowing lower and upper bounds.

- RecommendedExerciseType, which is of the type required multiple with cardinality (1:5). It consists of instances of the Session class, which will be described in 3.1.4. This field represents the recommended types of exercises for a person with this disease.
- RecommendedNrOfCardioSessions, which is of the type required multiple integer with cardinality (1:2). This attribute represents the recommended number of cardio sessions per week for a person with this disease. It has the minimum value 0, and maximum value 7, corresponding with the number of days in a week.
- RecommendedNrOfStrengthSessions, which is of the type required multiple integer with cardinality (1:2). This attribute represents the recommended number of strength sessions per week for a person with this disease. It has minimum value 0 and maximal value 7, corresponding with the number of days in a week.
- RecommendedNrOfExercises, which is of the type required multiple integer with cardinality (1:2). This attribute represents the recommended number of exercises in a strength session for a patient with this disease. Its minimum value is 0, and its maximum value is 10, as this is the highest number of recommended exercises per strength session observed in the source articles.
- RecommendedNrOfSets, which is of the type required multiple integer with cardinality (1:2). This attribute represents the recommended number of sets for an exercise in a strength session for a person with this disease. It has minimum value 0 and maximum value 5.
- RecommendedNrOfRepetitions, which is of the type required multiple integer with cardinality (1:2). This attribute represents the recommended number of repetitions of a strength exercise with a person with this disease. It has minimum value 0 and maximum value 15, as this is the highest number of recommended repetitions per set found in the source articles.

The Illness class has 12 instances in our ontology, each representing a disease found in the source articles. The instances/illnesses are: Cancer, Cardiac ischemia, Chronic obstructive pulmonary disease, Cystic fibrosis, Depression, Diabetes mellitus type 2, Fragile elders, Healthy, Hypertension arterial, Obesity and overweight, Osteoporosis and Rheumathoid arthrisis. Notice that we have included the instance Healthy; this is because it makes us able to construct a training program for a healthy person, using general health advice for healthy elders. The data in the attributes of these instances has been chosen based on the information given in the source articles.

3.1.4. Session

The Session class represents a collection of instances of the Exercise class, together forming a training session. The class attributes are Name and Exercises. The Name is simply a required simple String, representing the name of the session. The Exercises attribute is a required multiple of cardinality (1:10), containing instances of Exercises. The reason for the chosen cardinality, is that a session both can consist of only one exercise, for example running, or up to ten exercises, for example a collection of lower body strength exercises.

In our ontology, there are 47 instances of Session. Some of these are simple sessions like Swimming or Biking, but the biggest share of these instances are typical strength, balance and flexibility sessions, consisting of 6 to 10 exercises. The instances that are of this type, have a number at the end of its name. This number corresponds to the number of exercises in the session. Examples are Flexibility6, StrengthBalanceLower10 and StrengthUpper8. The reason that we have done this, is to easily be able to identify a session that is adequate for the patient. For example, if the patients recommended number of exercises for a strenght session is 7 (given by the attribute RecommendedNrOfExercises in the class Illness), we will recommend to the patient sessions ending with the number 7. Furthermore the title of the session implies what kind of exercises it is made up of; for example, the StrengthBalanceLower sessions contain exercises focusing on strength and balance in the lower part of the body.

3.1.5. TrainingProgram

The class TrainingProgram can be considered as the final product that we give back to the user of the application. It has 7 attributes, each representing a day of the week; Monday through Friday. Each day can consist of 0 to 2 instances of Session. This means that a day of the week can be a rest day, have one session or a maximum of two sessions.

3.2. The problem-solving methodology

In our application, we have chosen a problem solving methodology that is built around gathering a lot of information in the ontology. This way, we can obtain a lot of information from a low number of questions. For example, we can identify all the information we need about the quantitative details of the training program, such as the number of exercises per week, the number of repetitions per set and the intensity of the cardio sessions, only by asking for the disease of the patient and its physical shape.

3.3. Implementation of the solution

In this section it will be explained how the functions and rules we have implemented in CLIPS, as well as the ontology, together form the application.

This application is based on asking questions to a person in order to develop a training program. For asking questions we have used the method `ask-question (?question`

$?allowed-values). This method takes in as input the question and its allowed values, and prints the question until a legal value has been typed in by the user. For checking whether a yes or no-question has been answered with yes or no, we have used the function "ask-question" which returns TRUE if the answer was yes and else returns FALSE.

When the program is started, the first question asked to the user is Do you have any diseases?(yes/no). If the answer is yes, the question Which of the following diseases do you have? is prompted, followed by a numbered list of the diseases included in our system, as well as an option for None of the above. After this question is answered, the following facts are asserted: (disease-state disease "instance of disease") and (recommended-activities "the slot recommended activities from disease"). If the answer to the first question is no, the next question is not prompted, but the same facts are asserted with the instance Healthy of Disease and the corresponding recommended activities. The functionality of this first section of questions is made in the rules defrule determine-disease-state and defrule determine -recommended-activities, as well as in the function deffunction choose-illness (?val). The determine-disease-state rule asks the questions and the rule determine-recommended-activities finds the activities in the instance of the determined disease. The choose-illness function chooses the correct illness based on the input from the user.

The next question that is asked is Have you had any falls recently?(yes/no). If the answer is yes, the fact (fall-state fall yes) is asserted, and if the answer is no, the fact (fall-state fall no) is asserted. This logic is implemented by the rule defrule determine-fall-state, which asks the questions and asserts the fact accordingly.

The next question that is asked is How would you determine your physical shape?(low/mid/high). Depending on the answer, facts about physical shape, number of weekly cardio sessions, number of weekly strength sessions, duration of cardio sessions, number of reps, number of sets, and intensity of strength and cardio sessions are asserted. The first fact is asserted with the rule determine-physical-shape. The numbers in the other facts mentioned are determined in the following way; if the patient has a low physical shape, the minimum value is selected. If the patient has a medium physical shape, the average of the minimum and maximum value is selected, rounded down to the nearest integer. If the patient has a high physical shape, the maximum value is selected. For example, the number of sets for a patient with cancer with a medium physical shape, will be set to 10, as the numbers in the attribute Recommended-NrOfRepetitions for this illness is 8 and 12. The logic of this section is made in a number of rules and functions called determine-.... For example, the number of recommended repetitions is determined by the rule defrule determine-reps, which checks if disease and physical shape is asserted, and the function deffunction determine-reps, which calculates the number of reps according to physical shape and disease. Moreover, the functions deffunction roundDownAverage(?firstNr ?secondNr), deffuction min-value (?list) and deffunction max-value(?list) are used to find the rounded down average of two numbers and determine which of two number is smallest or biggest.

Next, the user is asked Do you practice, or would you like to practice, any of the following leisure activities?, with option of answering Golf, Hiking, Swimming, Cycling or None of the above. Depending on the answer, the fact (preferred-leisure-activity activity "the session corresponding to the answer) is asserted. This is done with the rule defrule determine-preferred-leisure-activity, which both asks the question and asserts the fact. This is the final question, and when the preferred-leisure-activity fact has been asserted, the fact (info-procurement finished) is asserted. This is done within the rule determine-preferred-leisure-activity, after the preferred-leisure-activity fact has been asserted.

The last process of the application is to construct the weekly training program based on the facts that have been asserted. In the beginning of the execution of the program, an empty instance of TrainingProgram is created. The logic behind filling this training program is as follows:

We have predetermined which days the sessions are placed on. For example, if the patient is to have 2 weekly cardio sessions and 2 weekly strength sessions, the cardio sessions will be on Tuesday and Sunday, and the strength sessions on Monday and Friday.

Moreover, we have defined that if a person has a preferred leisure activity, this activity will be set to Sunday, so that it can be done a day when most people do not work. For example, golf is most often practiced during the weekends. This rule is enforced through the rule (defrule put-leisure-activity-on-sunday, which checks if there is a preferred-leisure-activity and places it on Sunday.

When this is done, the days that are determined for cardio sessions are filled with the session from the fact (recommended-activities "..."). This is done with the rule defrule construct-cardio-trainingprogram, which fills the cardio session days with the recommended activities. It uses the function deffunction put-session-on-day (?session ?day) to put a session on the desired day. After this, if there are still "cardio days" left without an activity, it uses the function deffunction fill-in-walking-on-empty-cardio-session() to fill in the session Walking on these days. This has been done so that the training programs do not end up being to difficult to conduct; walking is an activity that almost anyone can do, and it has numerous positive impacts on the general health of a patient.

In the same manner, the days that are determined for strength sessions are filled. There are only two options

for the number of strength sessions adviced to a patient; 2 or 3. The only exception from this rule is if a patient suffers from depression; then no strength sessions are recommended (this is based on the information found in the source articles). If a patient has had a fall, and it has 2 strength sessions, then the first strength session is set to StrengthBalanceLower, and the second one is set to StrengthBalanceUpper. If the person has 3 recommended strength sessions, than an additional session of Flexibility is added. The same rule goes for a person who has not had a fall, but with the following sessions; StrengthLower and StrengthUpper, with an additional Flexibility session if the patient has 3 recommended strength sessions and not two. The chosen sessions contain the number of exercises corresponding to the number of exercised in the fact `(number-of-sessions strengthSessions "...")`. The logic for this part of the program is in the rule `defrule construct-strength-trainingprogram` which again uses the function `deffunction construct-strength-trainingprogram (?fall ?strength-sessions ?number-of-exercises)`. It uses the function `deffunction find-session (?session-name ?reps)` to find the sessions with the right number of exercises.

Lastly, the training program is printed. This is used with rule `defrule print-training-program` which simply prints out the weekly program, as well the recommendations with regards to cardio and strength sessions in addition to the warm-up and cool-down routine. This window can be seen in 4.

4. Results

We have decided to create test cases that show a variety of profiles, with different physical shape, hobbies, and illnesses.

4.1. Test case 1

The test subject has the following characteristics:

- Illness: Healthy
- Physical shape: High
- Falls: No falls
- Hobby: Swimming

Expected output:

- Cardio duration: 60 min
- Number of cardio sessions: 3
- Number of strength sessions: 2
- Cardio intensity: 14
- Strength intensity: 14
- Number of exercises: 8
- Number of sets: 3
- Number of repetitions: 10
- Activities: Swimming and two of {biking, running, rowing}

Actual output:

```
*** Your Weekly Training Program ***

     Monday:     StrengthLower8
     Tuesday:    Biking
     Wednesday:  Rest
     Thursday:   Running
     Friday:     StrengthUpper8
     Saturday:   Rest
     Sunday:     Swimming
--------------------------------------


     Info about your warm-up and cool-down

     Activities: Walking and stretching
     Duration: 15 minutes
     Set duration: 30 seconds
     Number of sets: 4
     Intensity: 10 on the Borg scale

--------------------------------------


     Info about your cardio-sessions

     Duration: 60 minutes
     Intensity: 14 on the Borg scale

--------------------------------------


     Info about your strength-sessions

     Number of repetitions: 10
     Number of sets: 3
     Intensity: 14 on the Borg scale

**************************************
```

4.2. Test case 2

The test subject has the following characteristics:

- Illness: Cancer
- Physical shape: Mid
- Falls: Yes
- Hobby: Golf

Expected output:

- Cardio duration: 25 min
- Number of cardio sessions: 4
- Number of strength sessions: 2
- Cardio intensity: 14
- Strength intensity: 13
- Number of exercises: 9
- Number of sets: 2
- Number of repetitions: 10
- Activities: Golf and three of {rowing, running, swimming}

Actual output:

```
*** Your Weekly Training Program ***

     Monday:     StrengthBalanceLower9
     Tuesday:    Rowing
     Wednesday:  Rest
     Thursday:   Running
     Friday:     StrengthBalanceUpper9
     Saturday:   Swimming
     Sunday:     Golf

---------------------------------------

     Info about your warm-up and cool-down

     Activities: Walking and stretching
     Duration: 10 minutes
     Set duration: 20 seconds
     Number of sets: 4
     Intensity: 10 on the Borg scale

---------------------------------------

     Info about your cardio-sessions

     Duration: 25 minutes
     Intensity: 14 on the Borg scale

---------------------------------------

     Info about your strength-sessions

     Number of repetitions: 10
     Number of sets: 2
     Intensity: 13 on the Borg scale

***************************************
```

4.3. Test case 3

The test subject has the following characteristics:

- Illness: Obesity and overweight
- Physical shape: Low
- Falls: No falls
- Hobby: None

Expected output:

- Cardio duration: 20 min
- Number of cardio sessions: 2
- Number of strength sessions: 2
- Cardio intensity: 13
- Strength intensity: 13
- Number of exercises: 8
- Number of sets: 1
- Number of repetitions: 8
- Activities: Two of {static bike, running}

Actual output:

```
*** Your Weekly Training Program ***

     Monday:     StrengthLower8
     Tuesday:    Static bike
     Wednesday:  Rest
     Thursday:   Rest
     Friday:     StrengthUpper8
     Saturday:   Rest
     Sunday:     Running
---------------------------------------

     Info about your warm-up and cool-down

     Activities: Walking and stretching
     Duration: 5 minutes
     Set duration: 10 seconds
     Number of sets: 4
     Intensity: 10 on the Borg scale
---------------------------------------

     Info about your cardio-sessions

     Duration: 20 minutes
     Intensity: 13 on the Borg scale

---------------------------------------

     Info about your strength-sessions

     Number of repetitions: 8
     Number of sets: 1
     Intensity: 13 on the Borg scale

***************************************
```

4.4. Test case 4

The test subject has the following characteristics:

- Illness: Depressions
- Physical shape: Low
- Falls: Yes
- Hobby: None

Expected output:

- Cardio duration: 30 min
- Number of cardio sessions: 3
- Cardio intensity: 10
- Activities: Three of {walking, running, static bike, biking, golf, skating, aerobic dance}

Actual output:

```
*** Your Weekly Training Program ***

     Monday:     Rest
     Tuesday:    Walking
     Wednesday:  Rest
```

```
     Thursday:   Running
     Friday:     Rest
     Saturday:   Rest
     Sunday:     Static bike
----------------------------------------

     Info about your warm-up and cool-down

     Activities: Walking and stretching
     Duration: 5 minutes
     Set duration: 10 seconds
     Number of sets: 4
     Intensity: 10 on the Borg scale

----------------------------------------

     Info about your cardio-sessions

     Duration: 30 minutes
     Intensity: 10 on the Borg scale

************************************
```

4.5. Test case 5

The test subject has the following characteristics:

- Illness: Fragile
- Physical shape: High
- Falls: Yes
- Hobby: None

Expected output:

- Cardio duration: 30 min
- Number of cardio sessions: 3
- Number of strength sessions: 3
- Cardio intensity: 12
- Strength intensity: 16
- Number of exercises: 8
- Number of sets: 3
- Number of repetitions: 12
- Activities: Three of {static bike, walking}

Actual output:

```
*** Your Weekly Training Program ***

     Monday:     StrengthBalanceLower8
     Tuesday:    Static bike
     Wednesday:  Flexibility8
     Thursday:   Walking
     Friday:     StrengthBalanceUpper8
     Saturday:   Rest
     Sunday:     Walking
----------------------------------------

     Info about your warm-up and cool-down

     Activities: Walking and stretching
     Duration: 15 minutes
     Set duration: 30 seconds
     Number of sets: 4
     Intensity: 10 on the Borg scale

----------------------------------------

     Info about your cardio-sessions

     Duration: 30 minutes
     Intensity: 12 on the Borg scale

----------------------------------------

     Info about your strength-sessions

     Number of repetitions: 12
     Number of sets: 3
     Intensity: 16 on the Borg scale

************************************
```

4.6. Test case 6

The test subject has the following characteristics:

- Illness: Rheumatoid arthritis
- Physical shape: High
- Falls: None
- Hobby: Cycling

Expected output:

- Cardio duration: 60
- Number of cardio sessions: 5
- Number of strength sessions: 3
- Cardio intensity: 17
- Strength intensity: 17
- Number of exercises: 10
- Number of sets: 3
- Number of repetitions: 12
- Activities: Five of {running, static bike, swimming, walking}

Actual output:

```
*** Your Weekly Training Program ***

     Monday:     StrengthLower10
     Tuesday:    Running
     Wednesday:  Flexibility10
     Thursday:   Swimming
     Friday:     StrengthUpper10
     Saturday:   Walking
     Sunday:     Biking
----------------------------------------

     Info about your warm-up and cool-down
```

```
    Activities: Walking and stretching
    Duration: 15 minutes
    Set duration: 30 seconds
    Number of sets: 4
    Intensity: 10 on the Borg scale

----------------------------------------

    Info about your cardio-sessions

    Duration: 60 minutes
    Intensity: 17 on the Borg scale

----------------------------------------

    Info about your strength-sessions

    Number of repetitions: 12
    Number of sets: 3
    Intensity: 17 on the Borg scale

**************************************
```

4.7. Test case 7

The test subject has the following characteristics:

- Illness: None of the listed

Expected output:

> "Please go to your doctor. We refrain from advising a training program on diseases we have no knowledge on."

Actual output:

```
Please go to your doctor. We refrain from
advising a training program on diseases we
have no knowledge on.
```

5. Discussion

For the most part, the results obtained are as expected. The program prints out a complete training program which describes every day of the week. The information of how to carry out the sessions is provided together with the description of sessions proposed.

Nonetheless, there are some flaws in the program that needs improvement. As we can see in 4.6, the program does not schedule the desired number of sessions when the number surpass seven. Although a person with Rheumatoid arthritis with a high physical shape is advised to do eight sessions a week, only seven are proposed in the program. To solve this, we have to handle having more than one session a day.

Another aspect of our solution that can be improved, is the fact that a flexibility session is defined as a strength session. This can be seen in 4.5 and 4.6. This has the consequence that the number of repetitions and sets increases and decreases in the same way as it does for the strength sessions. However, when the flexibility exercises initially were implemented in the ontology, we included slots for these numbers, as we thought that they should not be affected by the disease of the patient. Therefore, it will be relatively easy to change the implementation of the application so that the flexibility exercises become independent of the strength exercises.

As one can observe from the test cases, the cardio sessions selected from the recommended activity types, are chosen in the order that they are listed in our ontology. In order to increase the variability of the application, the cardio sessions can be randomly picked from the list of recommended exercises.

A final shortcoming of our program would be that we do not provide a progress plan to the patient. This could be a general advice provided to all patients, or a more specific one calculated depending on the patients physical shape.

6. Conclusion

To conclude, it can be said that the application serves its intended purpose. The program takes into account the diseases, falls, physical shape and hobbies of the user, and uses this information to choose exercises, number of repetitions and sets, duration and intensity of sessions that fits the user. However, as mentioned in 5, it does have a few shortages, but these are mostly small deficiencies that can easily be resolved.

[1] *Knowledge-based systems* (2018) Available at: https://en.wikipedia.org/wiki/Knowledge-based_systems (Downloaded: December 4, 2018).

[2] Noy, N. and McGuinness, D. *Ontology Development 101: A Guide to Creating Your First Ontology.* (2001) Available at: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf (Downloaded: December 4, 2018).

[3] Carpio, M et al. *Ejercicio físico para todos los mayores* (Downloaded: November 14, 2018)

[4] Hornero, M et al. (2014) *Guia de ejercicios para pacientes con enfermedad cardiovascular* (Downloaded: November 14, 2018)

[5] Izquierdo, M. et al. *Ejercicio físico es salud* (Downloaded: November 14, 2018)

[6] Gregorio, P et al. *Guía de ejercicio físico para mayores* (2012) (Downloaded: November 14, 2018)

[7] Alemán, J. et al. *Guía para la Prescripción de Ejercicio Físico en Pacientes con Riesgo Cardiovascular* (Downloaded: November 14, 2018)

[8] Giarrantano, Joseph C. *CLIPS User's Guide* (2002) Available at: http://www.lsi.upc.edu/~bejar/ia/material/laboratorio/clips/usrguide.pdf (Downloaded: November 14, 2018).

[9] *FAQ de CLIPS* (2011) Available at: http://www.lsi.upc.edu/~bejar/ia/material/laboratorio/clips/FAQ-CLIPS.pdf (Downloaded: November 14, 2018).

[10] *CLIPS Reference Manual Volume I* (2003) Available at: http://www.lsi.upc.edu/~bejar/ia/material/laboratorio/clips/bpg.pdf (Downloaded: November 14, 2018).