

Práctica de Sistemas Basados en el Conocimiento

Josep Maria Olivé Fernández, Pol Monroig, Yaiza Cano

20 de abril del 2020

Índice

1	Introducción	4
2	Identificación	5
2.1	Descripción del Problema	5
2.2	Viabilidad de un SBC	5
2.3	Fuentes de conocimiento	6
2.4	Objetivos	6
3	Conceptualización	7
3.1	Elementos del dominio	7
3.2	Descomposición en subproblemas	8
3.2.1	Recopilación de la información del usuario	8
3.2.2	Análisis y validación de la información recogida	8
3.2.3	Elaboración de la solución	8
3.3	Ejemplos del conocimiento del experto	9
3.4	Flujo de razonamiento	9
3.4.1	Suposiciones consideradas	10
4	Formalización	11
4.1	Desarrollo de un ontología	11
4.1.1	Dominio y cobertura	11
4.1.2	Considerar la reutilización de ontologías existentes	12
4.1.3	Enumerar los términos importantes en la ontología	12
4.1.4	Definir las clases y su jerarquía	12
4.1.5	Creación de instancias	18
4.2	Método de resolución	19
4.2.1	Abstracción de datos	19
4.2.2	Asociación heurística	20
4.2.3	Refinamiento	20
5	Implementación	21
5.1	Construcción de la ontología	21
5.2	Módulos	21
5.2.1	Módulo Input	21
5.2.2	Módulo Abstract	21
5.2.3	Módulo Solver	22
5.2.4	Módulo Output	22
5.3	Prototipos	22
5.3.1	Inicial	22
5.3.2	Mejorado	22
5.3.3	Final	22
6	Prueba	24
6.1	Prueba #1	24
6.1.1	Resultado esperado	24
6.1.2	Resultado obtenido	25
6.2	Prueba #2	25

6.2.1	Resultado esperado	25
6.2.2	Resultado obtenido	25
6.3	Prueba #3	26
6.3.1	Resultado esperado	27
6.3.2	Resultado obtenido	27
7	Conclusiones	29

1 Introducción

El objetivo de esta práctica es enfrentarnos a un problema y solucionarlo por un sistema basado en el conocimiento (SBC). Para llevar ésto a cabo, necesitaremos de la perspectiva de un *ingeniero del conocimiento* así como de un *experto*. Ambos dos serán recreados por los integrantes del grupo e, indirectamente, por las fuentes de información en las que nos basaremos.

Además, el problema que se plantea necesita resolver tareas de análisis e identificación y construcción de una solución; y, para hacerlo de manera adecuada, seguiremos la metodología explicada en clase de teoría que se caracteriza por ser sencilla, estar basada en un ciclo de vida en cascada, en un prototipado rápido y en un diseño incremental. Todas estas tipologías las explicaremos, en profundidad, en sus correspondientes fases.

2 Identificación

En esta fase llevaremos a cabo, en primer lugar, la descripción del problema que se nos propone. Acto seguido determinaremos si el problema se puede abordar, o no, mediante las técnicas de un sistema basado en el conocimiento (SBC). A partir de aquí, buscaremos las fuentes de conocimientos necesarias para desarrollar el sistema y que determinarán los datos necesarios para la resolución del problema y los criterios que determinen la solución.

2.1 Descripción del Problema

La cadena de gimnasios *Coaching Potato* ha decidido aprovecharse de las personas que han adquirido recientemente la necesidad que incorporar hábitos saludables a sus vidas. Para hacerlo, cree que una muy buena opción es ofrecer un servicio de recomendación de programas de entrenamiento personalizados a sus futuros clientes.

Así pues, con el fin de obtener un sistema bastante completo y que se ajuste a sus expectativas, necesitaremos obtener cierta información del usuario en cuestión como, por ejemplo, sus condiciones físicas generales, su estado de salud y sus hábitos más frecuentes.

Dichos datos serán relevantes para computar su estado físico que, a su vez, nos permitirá ajustar la intensidad inicial de los ejercicios a los que se enfrentará o, incluso, qué ejercicios serán, o dejarán de ser, aptos.

Es importante considerar la opción de que los usuarios no tengan problemas de salud ni hábitos destacables. Si esto ocurriese, el sentido común y el conocimiento especializado sobre este dominio serán capaces de rellenar y completar la información necesaria.

2.2 Viabilidad de un SBC

A primera vista, el problema que se plantea se podría tratar como un problema de búsqueda (metodología que hemos estado practicando hasta ahora), ya que nos estamos desplazando por un espacio de ejercicios de gimnasio para construir una rutina formada por un conjunto de ellos, respetando las posibles restricciones y preferencias del usuario. Aún así, el espacio de búsqueda del que partimos es suficientemente amplio y las posibles soluciones suficientemente numerosas como para que no sea posible resolver este problema con los métodos de resolución que contempla dicha técnica.

Resolverlo de forma algorítmica, sin embargo, no genera ninguna duda su inviabilidad ya que no se pueden representar todas las decisiones de exploración necesarias para llegar a una solución válida en una única función.

En cambio, este problema sí es adecuado para el tipo de técnica que se pide (SBC) ya que, por suerte, vivimos en una época donde tenemos mucha facilidad de acceso, gracias a internet, a las fuentes de conocimiento del dominio necesarias para resolver esta tarea y que denominaremos como *conocimiento experto*. Además, podemos ver que el problema tiene un tamaño adecuado que

no constituye algo inabordable por su complejidad.

Así pues, decidida una vez la metodología a través de la cual resolveremos el problema, nos disponemos a realizar una primera descripción de éste:

2.3 Fuentes de conocimiento

Las fuentes de conocimiento son intrínsecas en un SBC. Sin ellas, el sistema no podría tomar las decisiones que le llevarían a una solución. Así pues, nuestras fuentes se basan en:

- Los usuarios. Ellos introducirán los objetivos con los que quieren trabajar, los diferentes problemas de salud que puedan tener y que les pueda ocasionar alguna desventaja, los diversos hábitos que practiquen con cierta frecuencia, etc.
- Entrenadores personales. Ahora que estamos en cuarentena, ha habido mucho personal especializado en deporte que ha compartido rutinas para hacer ejercicio desde casa, conocimiento el cual podemos incluir y adaptar, por que no, en nuestro sistema.
- Nuestra experiencia. Además de lo que podamos encontrar en internet, tendremos en cuenta nuestra experiencia personal en el gimnasio con diferentes rutinas y en diferentes épocas de nuestra vida.

2.4 Objetivos

Primero de todo, necesitamos establecer los objetivos que queremos que asuma nuestro SBC para elaborar un buen sistema de recomendación de programas de entrenamiento:

- Obtener información relevante del usuario (condiciones físicas generales, hábitos personales, problemas de salud, objetivos del entrenamiento, ...).
- Computar su nivel de forma física. Esta información nos será de gran ayuda para ajustar la intensidad de su programa.
- Evaluar todos los posibles ejercicios dependiendo de si se ajustan a los objetivos del usuario.
- Descartar aquellos ejercicios que no son compatibles con los problemas de salud del usuario.
- Personalizar las intensidades de los ejercicios restantes según la capacidad física del usuario.
- Presentar un buen programa de entrenamiento semanal, con una breve descripción de qué se va a trabajar cada día y cuánto tiempo va a durar la sesión además de información específica necesaria para realizar la rutina según la modalidad de los ejercicios.

3 Conceptualización

En esta fase pretenderemos obtener una visión del problema desde el punto de vista del experto que nos permita decidir el conocimiento que es necesario para resolver el sistema. Así pues, necesitaremos:

- Conocer cuales son los conceptos que ha de manejar el sistema y sus características.
- Obtener una descomposición del problema en subproblemas.
- Descubrir el flujo del razonamiento en la resolución del problema.

Además de estos puntos, añadiremos algunos ejemplos del conocimiento del experto de manera que podamos hacernos a la idea de cómo se resuelven los problemas típicos desde el punto de vista del experto.

3.1 Elementos del dominio

Los conceptos que componen nuestro dominio son los siguientes:

Características de un **Usuario**.

- su nombre.
- su edad.
- su altura.
- su peso.
- la presión sanguínea: diastólica y sistólica.
- sus latidos por minuto.
- su índice de masa muscular.
- su nivel de forma física.
- los hábitos que tiene.
- los problemas de salud que tiene.

Características de un **Habit**.

- nombre del hábito.
- nivel de actividad física necesaria para realizarlo.
- la frecuencia y la duración con la que se realiza.

Características de un **Problem**.

- nombre del problema.
- tipo del problema: físico o de dieta.

Características de un **Exercise**

- nombre del ejercicio.
- tiempo durante el que se debe practicar el ejercicio: mínima y máxima.
- partes del cuerpo con las que se realiza el ejercicio.
- problemas a los beneficia y perjudica realizar el ejercicio.
- ámbito del ejercicio: crecimiento muscular, perder peso, balance corporal.

Además de estos elementos contemplaremos también **PersonalExercice**, concepto que nos permitirá personalizar como trabajar cada *Exercice* según las limitaciones del estado físico del *User* en cuestión.

Como último elemento tendremos **Program**, concepto que nos permitirá moldear la recomendación final que se le proporcionará al *User*. Consiste en una serie de *PersonalExercice* combinados de cierta manera a lo largo de diferentes días.

3.2 Descomposición en subproblemas

Descompondremos el problema en subproblemas y realizaremos un análisis por refinamientos sucesivos hasta obtener una relación jerárquica más o menos clara de las diferentes fases de resolución y los operadores de razonamiento más elementales.

3.2.1 Recopilación de la información del usuario

Lo primero que realizará el experto será hacer una serie de preguntas al usuario además de una prueba de esfuerzo que le servirán para hacerse una idea sobre el nivel de su estado físico; además, será conocedor también de los objetivos que espera asumir el usuario con el entrenamiento y los problemas de salud que éste acarrea y que le pudieran impedir la realización de determinados ejercicios.

3.2.2 Análisis y validación de la información recogida

Una vez recogida toda la información, el experto la analizará y realizará una primera selección de ejercicios basados en los objetivos del usuario. A continuación, realizará una valoración para saber si dichos ejercicios forman parte de la solución. Para llevar esto a cabo deberá, en un primer momento, descartar todos aquellos ejercicios que sean incompatibles o perjudiciales para el usuario según sus problemas de salud y a los restantes les asignará una intensidad que se adecue a las condiciones físicas del usuario.

3.2.3 Elaboración de la solución

Los ejercicios ya personalizados serán todos aquellos con los que, combinados entre ellos (de manera que puedan crear un programa de mínimo 30 minutos) y asignados en diferentes días (mínimo 3, máximo 7), formaran parte de la recomendación que puede ofrecer nuestro SBC. Por último, queda evaluar qué tan buena es la solución; pues bien, se valorarán positivamente aquellas soluciones que prioricen realizar aquellos ejercicios que son beneficiosos para los problemas de salud del usuario.

3.3 Ejemplos del conocimiento del experto

Para realizar las diferentes tareas de esta fase, necesitaremos observar primero cómo se resuelven los problemas más típicos desde el punto de vista del experto. Así pues, los datos que introduce el usuario serán la parte objetiva de la información y con la que se podrá deducir su nivel de forma física y, además, tendremos en cuenta los problemas de salud que acarrea. Con estos datos, habrán ejercicios que sean idóneos y ejercicios que no pueda realizar. Por ejemplo:

- Si el usuario se siente *fatigado* durante la prueba de esfuerzo, no podrá realizar ejercicios con *intensidad mediana*.
- Si el usuario tiene *problemas de salud*, no podrá hacer ejercicios que perjudiquen dichos problemas, pero se priorizarán aquellos que ayuden en alguna medida a beneficiarlos.
- Si la mayoría de hábitos de el usuario son *sedentarios*, la intensidad con la que realizará los ejercicios será baja, las sesiones diarias tendrán una duración moderada y la cantidad de días en los que se distribuirán dichas sesiones serán pocos por semana.
- Si la mayoría de hábitos de el usuario son de *actividad media* o no tiene hábitos en absoluto, la intensidad con la que realizará los ejercicios será mediana, las sesiones diarias tendrán una duración normal y la cantidad de días en los que se distribuirán dichas sesiones serán bastantes días por semana.
- Si la mayoría de hábitos de el usuario son de *actividad alta*, la intensidad con la que realizará los ejercicios será elevada, las sesiones diarias tendrán una duración completa y la cantidad de días en los que se distribuirán dichas sesiones abarcarán toda la semana.

3.4 Flujo de razonamiento

El usuario, después de identificarse en el sistema, responderá una serie de preguntas relacionadas con su estado físico y su salud y se le pedirá que especifique qué objetivo(s) pretende asumir con este entrenamiento.

Una vez obtenida toda la información, se abstraerán el nivel de estado físico del usuario y se seleccionarán los ejercicios que cumplan con sus objetivos.

Para mejorar la calidad de la selección, se descartarán todos aquellos ejercicios que perjudiquen los problemas de salud del usuario y se priorizarán aquellos que los beneficien.

Obtenidas, por fin, todos los posibles ejercicios, se les asignará a cada uno de ellos una intensidad adecuada al nivel de estado físico del usuario y se repartirán a lo largo de los diferentes días que dura el programa (máximo 7 días).

3.4.1 Suposiciones consideradas

El sistema supondrá lo siguiente:

- El usuario no introducirá como respuesta a las preguntas formatos no contemplados entre los corchetes[].
- El usuario no introducirá una altura o peso de 0 metros o negativa.
- Que el usuario sabra diferenciar los habitos de cada tipo

4 Formalización

En esta fase pretendemos obtener una visión del problema desde el punto de vista del ingeniero del conocimiento, de manera que tendremos que trasladar todo el análisis previo que se ha realizado desde el punto de vista del experto, a una representación acorde para la implementación de un SBC.

4.1 Desarrollo de una ontología

En inteligencia artificial, una ontología será una descripción formal explícita de los conceptos de un dominio. Eso quiere decir que el desarrollo de una ontología requerirá definir las clases que forman el dominio, organizarla en una jerarquía taxonómica, definir las propiedades de cada clase e indicar las restricciones, asignar valores a las propiedades con las que crearemos instancias...

Al contrario de lo que pueda parecer, no existe un único método correcto con el que modelar un dominio, ni una ontología invariable. El desarrollo de ésta es un proceso iterativo y nosotros hemos decidido seguir los pasos de la metodología informal practicada en clase de teoría aplicada a nuestro problema.

4.1.1 Dominio y cobertura

Para saber que elementos representaran de forma adecuada nuestra ontología, nos hemos tenido que plantear que tipos de preguntas y respuestas deseamos obtener, esto se conoce como *preguntas de competencia*.

1. ¿Qué dominio abarcará la ontología?

Los elementos del dominio los hemos descrito desde el punto de vista del experto en la fase de *Conceptualización* y nos servirá como base para definir el dominio desde el punto de vista del ingeniero en el conocimiento. Principalmente, los conceptos a definir junto con sus características serán: *User*, *Habit*, *Problem*, *Exercise*, *PersonalExercise* y *Program*.

2. ¿Qué finalidad tiene la ontología?

La ontología representará los ejercicios que pueden realizar diversos usuarios para mejorar su forma física. La utilizaremos, principalmente, para ofrecer un sistema de recomendación de programas de entrenamiento que seguir semanalmente.

3. Preguntas de competencia

La ontología deberá responder principalmente a preguntas relacionadas con el nivel de estado físico del usuario abstraído de los datos que éste ha introducido inicialmente en el sistema.

En el enunciado de la práctica se nos exige realizar las siguientes preguntas de forma obligatoria:

- ¿Qué altura y peso tiene el usuario?
- ¿Qué hábitos personales tiene el usuario, con cuánta frecuencia los practica y durante cuánto tiempo?

- ¿Qué problemas tanto músculo-esqueléticos como de dieta tiene el usuario?

A parte de éstos, también se nos propone, de manera opcional, realizar una prueba de esfuerzo de la cual se extraen las siguientes preguntas:

- ¿Cuántas pulsaciones por minuto tiene el usuario?
- ¿Siente mareo o fatiga el usuario?

Por último, nosotros mismos hemos decidido incorporar, de manera opcional, la siguiente pregunta.

- ¿Qué objetivo(s) pretende asumir el usuario con el programa de entrenamiento?

4. ¿Quién usará y mantendrá la ontología?

Para sorpresa de nadie, la cadena de gimnasios *Coaching Potato* no existe, esta ontología la utilizaremos un grupo de alumnos de la asignatura de inteligencia artificial de la carrera de ingeniería informática. Así pues, nosotros mismos seremos quienes la mantengamos durante todo el periodo que dure la realización de esta práctica.

No obstante y sin embargo, ésta no está limitada exclusivamente a nuestro uso, una de las partes buenas de esta asignatura es que, en la vida real, podremos aplicar las cosas que aprendemos aquí, ergo, si, por ejemplo, el *Poliesportiu Campus Nord* quisiera implementar un sistema de recomendación de programas de entrenamiento a sus usuarios, podría reutilizar nuestra ontología y adaptarla a sus objetivos.

4.1.2 Considerar la reutilización de ontologías existentes

Para la realización de esta práctica, hemos decidido crear una ontología nosotros mismos en función de los que se nos decía en el enunciado de la práctica. No hemos reutilizado ninguna ontología que ya estuviera creada previamente.

4.1.3 Enumerar los términos importantes en la ontología

La lista de los términos importantes que podemos usar para referirnos a nuestro dominio están detalladas en apartado 1 de la fase de *Conceptualización*. Aún así, resumimos aquí que nuestro dominio hablará de usuarios, ejercicios, hábitos y problemas de salud principalmente.

4.1.4 Definir las clases y su jerarquía

La jerarquía con la hemos representados los conceptos es *de arriba a abajo*, es decir, definimos los conceptos más generales y vamos especializándolos.

Así pues, la ontología completa que hemos realizado es la siguiente:

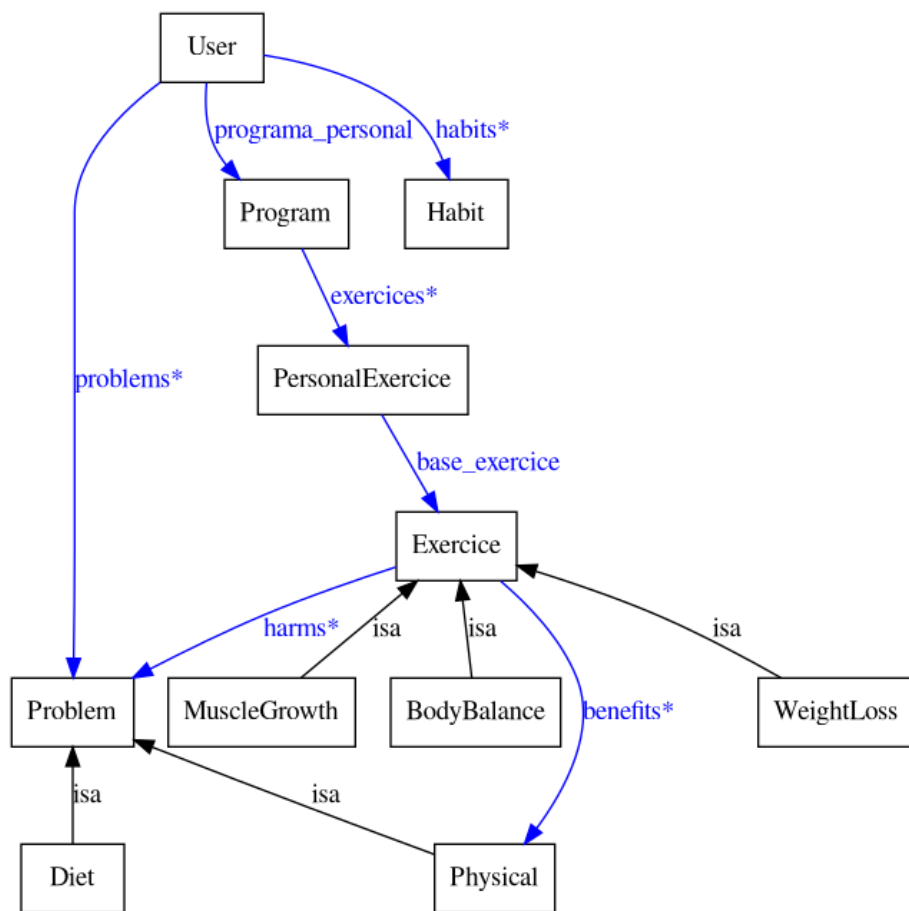


Figure 1: Ontología completa en Protege

4.1.4.1 User

User		
fatigue	Boolean	
min_blood_pressure	Float	
max_blood_pressure	Float	
bpm	Float	
age	Integer	
height	Float	
instance_name	String	
bmi	Float	
activity	Symbol	low
		medium
		high
weight	Float	
habits	Instance*	Habit
problems	Instance*	Problem
programa_personal	Instance	Program

Figure 2: Clase User

El *usuario* es uno de los conceptos principales de nuestra ontología ya que será él quien nos proporcione toda la información necesaria para poder definir su estado físico y generar una recomendación idónea; se caracteriza por:

- fatigue: si el usuario se ha sentido fatigado durante la prueba de esfuerzo.
- min_blood_pressure: presión sanguínea diastólica del usuario después de la prueba de esfuerzo. Necesaria para saber si el usuario tiene hipertensión.
- max_blood_pressure: presión sanguínea sistólica del usuario después de la prueba de esfuerzo. Necesaria para saber si el usuario tiene hipertensión.
- bpm: latidos por minuto del usuario después de la prueba de esfuerzo.
- age: edad del usuario.
- height: altura del usuario.
- instance_name: nombre del usuario.
- bmi: índice de masa corporal del usuario. Necesario para saber si tiene sobrepeso u obesidad.
- activity: nivel de forma física del usuario.
- weight: peso del usuario.
- habits: hábitos que tiene el usuario.
- problems: problemas de salud que tiene el usuario.
- programa_personal: rutina de ejercicios personalizada según las características del usuario que genera como output nuestro sistema.

4.1.4.2 Habit

Habit		
activity	Symbol	low
		medium
		high
frequency		Integer
instance_name		String
duration		Float

Figure 3: Clase Habit

El elemento *hábito*, representa todas aquellas actividades que realiza el usuario de forma relativamente frecuente y que sirven para definir su nivel de estado físico. Se caracteriza por:

- activity: nivel de actividad física que requiere la actividad para ser realizada.
- frequency: frecuencia con la que el usuario realiza el hábito.
- instance_name: nombre del hábito.
- duration: tiempo que invierte el usuario al realizar el hábito.

4.1.4.3 Problem

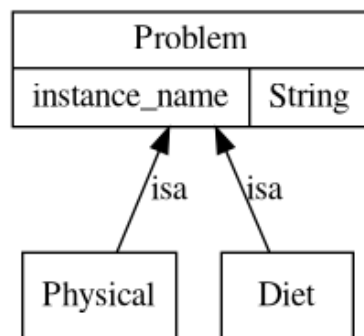


Figure 4: Clase Problem

El elemento *problema*, representa todos aquellos problemas de salud que pueden impedir o dificultar la realización de algunos ejercicios por parte del usuario. Solo se almacena su nombre.

En este concepto podemos observar una jerarquización debido a que podemos diferenciar entre diferentes tipos de problemas.

- Physical.
- Diet.

4.1.4.4 Exercise

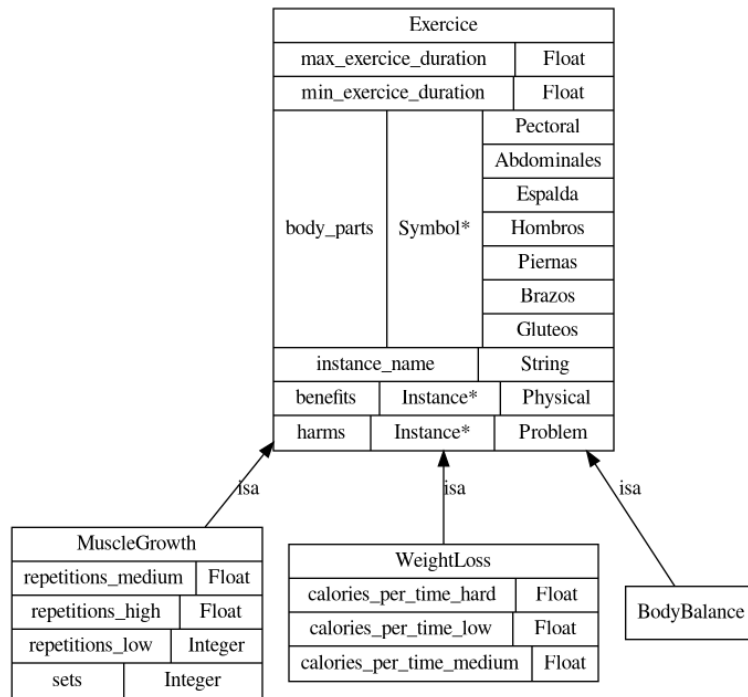


Figure 5: Clase Exercise

El elemento *ejercicio*, representa todos aquellos ejercicios que se pueden realizar en el gimnasio *Coaching Potato* y con los que se le podría generar un programa de entrenamiento al usuario.

- max_exercise_duration: duración máxima que tendrá el ejercicio.
- min_exercise_duration: duración mínima que tendrá el ejercicio.
- body_parts: partes del cuerpo que se ejercitan con el ejercicio.
- instance_name: nombre del ejercicio.
- benefits: problemas de salud que se ven beneficiados con el ejercicio.
- harms: problemas de salud que se ven perjudicados con el ejercicio.

En este concepto podemos observar, también, una jerarquización debido a que diferenciamos entre 3 tipos diferentes de ejercicios.

- MuscleGrowth: ejercicio enfocado al desarrollo muscular. Se almacenan la cantidad de repeticiones según diferentes intensidades y la cantidad de conjuntos.
- WeightLoss: ejercicio enfocado a quemar grasas. Se almacenan la cantidad de calorías previstas que se quemarán según diferentes intensidades.
- BodyBalance: ejercicio enfocado a ganar equilibrio y flexibilidad.

4.1.4.5 PersonalExercise

PersonalExercise		
exercice_type	Symbol	bodybalance
		musclegrowth
		weightloss
bonus		Boolean
base_exercice	Instance	Exercise
day		Integer*
difficulty	Symbol	low
		normal
		hard
duration		Float

Figure 6: Clase PersonalExercise

El elemento *ejercicio personal*, representa la adaptación de los ejercicios descritos anteriormente a las capacidades del usuario y se caracteriza por:

- exercise_type: ámbito del ejercicio seleccionado.
- bonus: el ejercicio beneficia algún problema de salud que tenga el usuario.
- base_exercice: ejercicio seleccionado para personalizar según las características del usuario.
- day: día(s) de la semana que el usuario deberá realizar el ejercicio.
- difficulty: intensidad del ejercicio según el nivel de forma física del usuario.
- duration: tiempo total que transcurre entre el principio y el fin del ejercicio según el nivel de forma física del usuario.

4.1.4.6 Program

Program		
exercices	Instance*	PersonalExercise

Figure 7: Clase Program

El elemento *programa*, será la lista del programa de entrenamiento final. En ella aparecerán los ejercicios personalizados anteriores repartidos en varios días de manera que el usuario entienda, de forma fácil, que tiene que hacer durante la semana y que va a trabajar cada día.

- *exercices*: programa personalizado de ejercicios según las características del usuario. Es la solución que presenta nuestro sistema.

4.1.5 Creación de instancias

En nuestra ontología hemos creado instancias para los ejercicios y los problemas.

4.1.5.1 Exercice

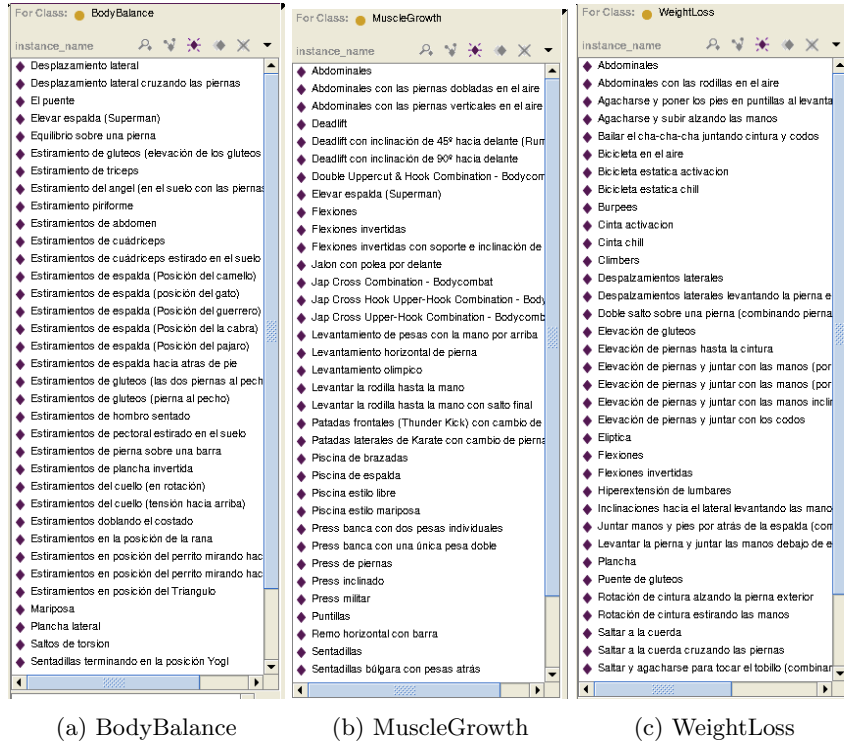


Figure 8: Instancias de Exercices en Protege.

4.1.5.2 Problem

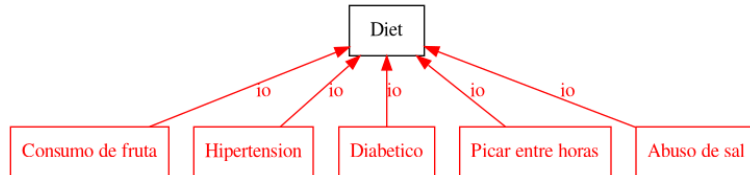


Figure 9: Clase Diet

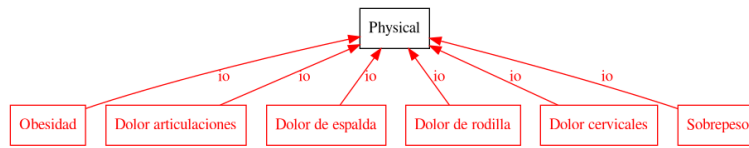


Figure 10: Clase Physical

4.2 Método de resolución

Existen ciertas técnicas generales de resolución que se pueden aplicar a diferentes tipos de dominios y tareas. De entre ellas, en clase de teoría se nos destacan dos:

- Clasificación Heurística
- Resolución Constructiva

La principal diferencia entre ambas es que la segunda, en contraste con la primera, puede atacar el problema mediante métodos no guiados por el conocimiento y que las soluciones no se pueden enumerar, sino que se han de construir.

Así pues, con dicha comparativa y entendiendo que la primera técnica requiere de realizar inferencias y transformaciones sobre el problema para poder asociarlo con la descripción de las clases, hemos decidido que ésta se adapta a nuestra situación.

La clasificación heurística se encuentra dividida en tres etapas y necesita del esquema de razonamiento del experto para realizar las inferencias que acabamos de mencionar.

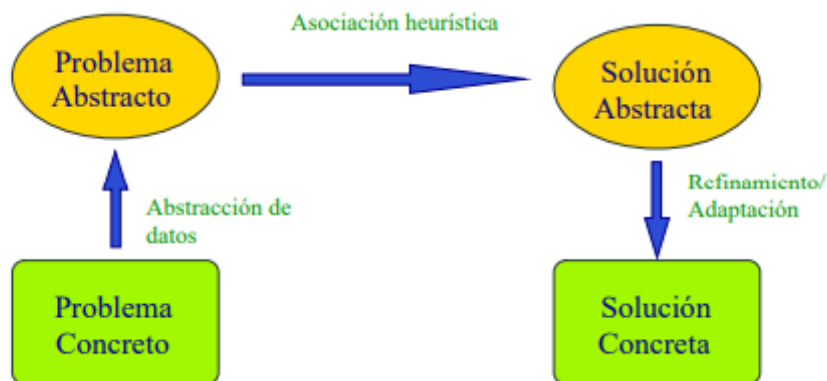


Figure 11: Clasificación Heurística

4.2.1 Abstracción de datos

La información con la que contaremos son las características que definen al usuario, ya sea de manera explícita con las respuestas a las preguntas iniciales como por ejemplo si tiene algún problema de salud, o de manera implícita realizando alguna que otra abstracción como por ejemplo su nivel de estado físico.

Si echamos un vistazo a nuestro código CLIPS, esta etapa está incluida en el módulo *input_module*.

4.2.2 Asociación heurística

Una vez situados en el *Problema Abstracto*, debemos hacer la inferencia de las características que tienen los ejercicios que forman parte del conjunto de la solución. Por ejemplo, descartaremos todos aquellos ejercicios que no sean compatibles con los problemas de salud del usuario.

Si echamos un vistazo a nuestro código CLIPS, esta etapa está incluida en el módulo *abstract_module*.

4.2.3 Refinamiento

La última fase consistirá en 2 partes:

En primer lugar les asignaremos una intensidad acorde al nivel de estado físico del usuario a los ejercicios restantes.

Por último, evaluaremos dichos ejercicios y priorizaremos recomendar aquellos que son beneficiosos de alguna manera para los problemas del usuario.

Si echamos un vistazo a nuestro código CLIPS, esta etapa está incluida en los módulos *output_module* y *solver_module*.

5 Implementación

En esta fase implementaremos la ontología que servirá como base para la resolución del problema. Decidiremos, también, como encajar la descomposición de subproblemas de las tareas involucradas en el sistema con las metodologías de resolución de problemas genéricas que habremos identificado en la fase anterior.

5.1 Construcción de la ontología

La representación final de la ontología la hemos generado con *Protege*. El código en CLIPS lo hemos dividido en 4 módulos intentando imitar la metodología de la *clasificación heurística* que detallaremos en el siguiente apartado con sus correspondientes reglas.

En cuanto a la instanciación de las clases, desde *Protege* solo tenemos de *Exercice* y *Problem* (con sus respectivas subclases). Mientras que la única de la clase *User* se crea de manera dinámica durante la ejecución del programa, así como las de la clase *PersonalExercice* y las de *Habit*.

5.2 Módulos

A continuación explicaremos como hemos dividido la ontología en diferentes módulos y las reglas más relevantes se ejecutan en cada uno.

5.2.1 Módulo Input

Este módulo es el que se encarga de recibir todos los datos que introduce el usuario a través del cuestionario inicial. Esta información concreta del problema se almacena en la instancia del usuario sin embargo no se procesa aquí.

Las reglas más importantes son: *input_questions*, *problems_questions* y *habits_questions* en las cuales se le pregunta al usuario información general suya, problemas físicos o dietéticos que pueda tener y sus hábitos más frecuentes, respectivamente.

5.2.2 Módulo Abstract

En este segundo modulo, obtenemos información deducible a través del usuario (i.e. tiene sobrepeso, obesidad, hipertensión). Después obtenemos el nivel de actividad que tiene según sus hábito, su edad y el resultado de la prueba física. Este nivel de actividad es muy importante ya que es lo que determinara la dificultad de los ejercicios al igual que la duración del programa. En este modulo también elegimos los ejercicios que puede que haga el usuario según los objetivos que ha escogido en el modulo Input. Por último eliminamos ejercicios que por algún problema no puede hacerlos, ya que le resultaría dañino, por otra parte, bonificamos los ejercicios que benefician algún problema que tenga el usuario. Sin embargo, solo podemos bonificar problemas físicos ya que los problemas dietéticos (e.g. Diabetes) no podrán mejorar según un ejercicio.

5.2.3 Módulo Solver

Lo primero que hace este modulo es elegir la cantidad de días que tendrá el programa según la actividad del usuario. Este modulo es mucho más pequeño ya que solo se encarga de asignarle días específicos a los días, según dos listas distintas, la de los ejercicios que benefician al usuario y la de los ejercicios que neutrales para este (no le perjudican) se le van asignando dándole prioridad a los que benefician a este. Sin embargo, intenta hacer una selección variada, evitando repeticiones de ejercicios el mismo día. También da una preferencia a los ejercicios que aun no tienen un día asignado.

5.2.4 Módulo Output

Este módulo se encarga de imprimir la solución por pantalla al usuario, sin embargo también hace función de refinamiento; en la que según algunos atributos concretos del problema, tales como las partes del cuerpo de cada ejercicio, se genera una solución concreta.

5.3 Prototipos

Para resolver nuestro problema, hemos procedido con una metodología basada en prototipado rápido y desarrollo incremental; por ello, debemos construir un prototipo inicial a partir de una restricción de las capacidades del sistema.

5.3.1 Inicial

El prototipo inicial es capaz de calcular el índice de masa corporal e identificar problemas de obesidad o sobrepeso, es capaz de definir la actividad física del usuario según sus hábitos y ser consecuente en la dificultad de los ejercicios propuestos a pesar de que la distribución y elección de los ejercicios es aleatoria. No hemos registrado aun los diferentes problemas que pueda tener el usuario. El prototipo es capaz de generar una rutina de 7 días en la cual el tiempo total de los ejercicios diarios superan los 30 minutos.

5.3.2 Mejorado

Este prototipo intermedio es capaz de, además de realizar todo lo que hacía el inicial, descartar ejercicios que perjudicaran los problemas de salud del usuario, la elección de los ejercicios que montan la rutina deja de ser aleatoria para ser elegida por el usuario según sus objetivos y se ha ajustado los días en los que se reparten las rutinas de ejercicios al nivel de estado físico del usuario.

5.3.3 Final

Este prototipo definido hemos priorizado los ejercicios que beneficiaban los problemas de salud del usuario, hemos hecho que pequeños factores que no habíamos utilizado hasta entonces como la presión sanguínea o la sensación de fatiga influyeran en el nivel de estado físico del usuario. Además, hemos añadido toda la parte de comunicación con el usuario sobre qué datos necesita introducir, se le hace conocedor de la intensidad de ejercicios que va a poder realizar, cuales de dichos ejercicios se están descartando y porqué y cuales de dichos ejercicios

se están priorizando y porqué.

Finalmente, al mostrar el programa al usuario, se le detallará el tiempo total de la rutina diaria, que ámbitos se trabajarán y datos específicos de cada tipología de ejercicio como por ejemplo la cantidad esperada de calorías que se perderán con ejercicios *weightloss* o la cantidad de conjuntos y repeticiones a realizar con ejercicios *musclegrowth*.

6 Prueba

En esta fase nos centraremos en explicar los diferentes juegos de pruebas que hemos utilizado para validar nuestro sistema y, si alguno de ellos nos han obligado a cambiar el diseño. La validación de un SBC, por norma general, es más compleja que en otros sistemas software dado que estamos implementando resoluciones de problemas de naturaleza heurística, por lo que es muy difícil determinar la completitud y correctitud del sistema.

6.1 Prueba #1

En esta prueba queremos comprobar que si el sistema es capaz de identificar correctamente si el usuario tiene sobrepeso, o hipertensión. Para ello, vamos a dar valores que se encuentren ligeramente por encima de los valores límite.

En el caso de que $bmi = peso / (altura * altura)$ sea superior o igual a 25, el usuario tendrá sobrepeso, y en el caso que este por encima de 30, se considerará que este tiene obesidad.

En el caso de la hipertensión, si la presión sanguínea mínima es superior a 90, o la máxima a 114, también se considera que el usuario tiene hipertensión.

Hacemos diversas pruebas, con diversos grupos de valores para este caso:

Grupo 1:

- 1.61m de altura.
- 70 kg de peso.
- Presión sanguínea máxima 115.
- Presión sanguínea mínima 89.

Grupo 2:

- 1.61m de altura.
- 90 kg de peso.
- Presión sanguínea máxima 110.
- Presión sanguínea mínima 97.

6.1.1 Resultado esperado

Para llevar a cabo este juego de pruebas, vamos a introducir datos prácticamente vacíos en cuanto a objetivos y hábitos, mientras que introduciremos los valores comentados anteriormente, esperando que las alertas de *hipertensión* y *sobrepeso* u *obesidad* en su caso, salten.

6.1.2 Resultado obtenido

Efectivamente, vemos que nuestro programa nos indica lo siguiente como salida del primer grupo:

- "El usuari té hipertensió"
- "El usuari té sobrepes"

Mientras que ésta es la salida del segundo grupo:

- "El usuari té hipertensió"
- "El usuari té obesitat"

6.2 Prueba #2

En esta prueba queremos ver si nuestro programa es capaz de controlar bien los casos en los que nuestro usuario tiene fatiga. Para ello, vamos a introducir datos en la entrada que podrían resultar un tanto contradictorios a primera vista pero para los que hemos dotado a nuestro sistema de reglas para reconocer y actuar en consecuencia.

La entrada consistirá en indicar que el usuario tiene fatiga al realizar la prueba de esfuerzo, pero también vamos a comentar que hace múltiples actividades de alto rendimiento a lo largo de la semana. Este último punto por separado debería hacer que, en condiciones normales, se recomendara al usuario actividades de intensidad alta pero que al remarcar que éste sufre de fatiga, se debería limitar la intensidad de las actividades a una intensidad media.

6.2.1 Resultado esperado

Esperamos pues que nuestro programa actúe en consecuencia con la fatiga y limite la actividad del usuario en el gimnasio a una intensidad media por el bien de su salud ya que por mucho que éste haya asegurado que la mayoría de actividades que realiza son de intensidad alta, la prueba de esfuerzo dice justo lo contrario. Además de planificar un horario de como mucho 5 días de ejercicios.

6.2.2 Resultado obtenido

Efectivamente, vemos que nuestro programa, con las condiciones especificadas, prioriza la fatiga y nos da un programa de ejercicios para 5 días en vez de 7. Podemos confirmar que la intensidad es media gracias a este fragmento de la salida.

- "El nivel de actividad es medio"

6.3 Prueba #3

En esta prueba nos centramos en comprobar que nuestro programa es capaz de generar un horario correcto a pesar de tener un usuario con muchos problemas tanto físicos, como alimenticios o de hipertensión. Para ello vamos a probar tres ejecuciones diferentes del programa con los siguientes grupos de datos de entrada y esperando que como mínimo tengamos rutinas de 30 minutos cada día para un mínimo de 3 días distintos por ejecución.

Grupo 1:

- 1.61m de altura.
- 70 kg de peso.
- Presión sanguínea máxima 115.
- Presión sanguínea mínima 89.
- Responder que si a todos los problemas.
- Agregar hábitos sedentarios y fatiga.
- Agregar como objetivo Balance.

Grupo 2:

- 1.61m de altura.
- 90 kg de peso.
- Presión sanguínea máxima 110.
- Presión sanguínea mínima 97.
- Responder que si a todos los problemas.
- Agregar hábitos sedentarios y fatiga.
- Agregar como objetivo MuscleGrowth.

Grupo 3:

- 1.61m de altura.
- 90 kg de peso.
- Presión sanguínea máxima 110.
- Presión sanguínea mínima 97.
- Responder que si a todos los problemas.
- Agregar hábitos sedentarios y fatiga.
- Agregar como objetivo WeightLoss.

6.3.1 Resultado esperado

El resultado esperado en todos los casos es que la intensidad del programa sea baja, y que solo se nos generen horarios de 3 días en todos los casos, pero siempre con un mínimo de 30 minutos de ejercicios por día y sin ninguna repetición de ejercicio dentro del mismo día.

Además, esperamos que nuestro programa priorice los ejercicios que van a beneficiar la condición del usuario, es decir, que si tiene dolor de espalda, esperamos que le recomiende ejercicios favorables a dicho dolor evitando los dañinos.

6.3.2 Resultado obtenido

Efectivamente, vemos que los resultados son los esperados, aquí los datos importantes de la salida para cada caso:

Grupo 1:

- "Segons l'activitat del usuari s'han programat els exercicis amb una intensitat baixa"
- "El usuari té hipertensió"
- "El usuari té sobrepes"
- "Se han encontrado un total de 20 ejercicios con recomendación alta"
- "Se han encontrado un total de 5 ejercicios con recomendación neutral"
- Se han eliminado los ejercicios dañinos para la condición del usuario, y se han priorizado los beneficiosos.
- Todos los días tienen un mínimo de 30 minutos de ejercicios, sin repetir ejercicios
- Horario de 3 días

Grupo 2:

- "Segons l'activitat del usuari s'han programat els exercicis amb una intensitat baixa"
- "El usuari té hipertensió"
- "El usuari té obesitat"
- "Se han encontrado un total de 5 ejercicios con recomendación alta"
- "Se han encontrado un total de 7 ejercicios con recomendación neutral"
- Se han eliminado los ejercicios dañinos para la condición del usuario, y se han priorizado los beneficiosos.
- Todos los días tienen un mínimo de 30 minutos de ejercicios, sin repetir ejercicios.
- Horario de 3 días

Grupo 3:

- "Segons l'activitat del usuari s'han programat els exercicis amb una intensitat baixa"
- "El usuari té hipertensió"
- "El usuari té obesitat"
- "Se han encontrado un total de 3 ejercicios con recomendación alta"
- "Se han encontrado un total de 17 ejercicios con recomendación neutral"
- Se han eliminado los ejercicios dañinos para la condición del usuario, y se han priorizado los beneficiosos.
- Todos los días tienen un mínimo de 30 minutos de ejercicios, sin repetir ejercicios.
- Horario de 3 días

7 Conclusiones

Durante esta práctica hemos tenido sentimientos encontrados con el lenguaje de programación CLIPS. Al principio se nos hacia muy semejante a Prolog pero conforme hemos ido avanzando en su desarrollo hemos ido comprobando que a medida que el tamaño del código aumentaba, también lo hacia su ambigüedad.

Hemos tenido serios problemas, por ejemplo, con como se ejecutaban las reglas ya que éstas, una vez agendadas, no las podíamos desagendar y nos hemos encontrado en la situación que una condición x dada se cumpliera y se agendara la regla correspondiente, se disparara un regla (anterior a ésta) que modificara la condición x y que la regla correspondiente se acabara disparando de igual forma.

Nos ha tomado gran cantidad de tiempo conseguir este resultado ya que cada cosa que añadíamos, error que surgía, regla que no se disparaba o procedimiento que no funcionaba como creíamos que lo habíamos programado.

Aunque es cierto que CLIPS nos ha ayudado mucho a visualizar como funciona un SBC internamente debido a su flexibilidad de poder separar el código de forma muy independiente.

Bibliografia

- [1] Appendix G - CLIPS Error Messages. http://galahad.plg.inf.uc3m.es/~iat/CLIPS_REFERENCIA/Appendix_G_-_CLIPS_Error_Messages.html.
- [2] Apunts de representació del coneixement. <https://www.cs.upc.edu/~jvazquez/teaching/iag/apuntes/ApuntesIA-G-ReprK.pdf>.
- [3] Clasificación del sobrepeso y la obesidad. <https://meiga.info/Escalas/Obesidad.pdf>.
- [4] CLIPS. <http://www.clipsrules.net>.
- [5] Clips User's Guide. <https://www.cs.upc.edu/~bejar/ia/material/laboratorio/clips/usrguide.pdf>.
- [6] Ejercicios equilibrio y coordinación. <https://atopedegym.com/ejercicios-para-equilibrio-coordinacion/>.
- [7] FAQ de CLIPS. <https://www.cs.upc.edu/~bejar/ia/material/laboratorio/clips/FAQ-CLIPS.pdf>.
- [8] Graphviz plugin. <https://www.graphviz.org>.
- [9] Protegé 3.5. https://protege.stanford.edu/download/protege/3.5/installanywhere/Web_Installers/.