

Trabajo practico 1

Ontología sobre Relojes

Introducción a la Inteligencia Artificial

Facundo Emmanuel Messulam y Franco Ignacio Vallejos Vigier

Mayo 2022

1 Introducción

Para este trabajo de ontologías decidimos tomar relojes, se tomaron una serie de características, se crearon instancias que ejemplifican las características y se generaron una serie de queries relevantes. Se modelan relojes para ayudar al usuario a entender las características y los tipos de relojes que sus clientes pueden desear.

La información para este trabajo fue generada con el conocimiento experto de los integrantes del grupo, y no se analizaron otras ontologías.

2 Desarrollo

2.1 Clases

Para poder modelar información sobre damos una serie de características que pertenecen a nuestro modelo:

- **Correa** la correa por su material
- **Cristal** el tipo de material de la cobertura transparente
- **Dial** la cara del reloj
- **Movimiento** el mecanismo de mantención del tiempo

Para representar *Cristal*, lo hacemos mediante su clase homóloga *Cristal* que va aportar información con respecto a cualidades intrínsecas de los distintos tipos de cristal. A esto último lo hacemos mediante Data Properties:

- **Grosor** En su representación es un Entero que hace referencia al grosor del cristal en mm
- **Color** Es una String en relacion al color de cristal

- **Antirrayones** Es un Bool que denota si el Cristal tiene o no dicha propiedad
- **Dureza** Es un Entero del 0 al 10 que hace referencia a la dureza del cristal en la escala de Mohs
- **Material Derivado** Es una String que representa su material madre en la fabricación de este

En si esta Clase Cristal tiene subclases hijas que son definidas mediante predicados en Equivalent To:

- **Plexiglas** Clase definida por Equivalencia, compuesta por todas aquellas instancias de Cristal que su Material Derivado es el Plastico, que posee una Dureza de 3 o 4 y es plausible a rayones
- **Sapphlex** Clase definida por Equivalencia, compuesta por todas aquellas instancias de Cristal que su Material Derivado es el Zafiro, que posee una Dureza de 8 y no es afectado por rayones
- **Hardlex** Clase definida por Equivalencia, compuesta por todas aquellas instancias de Cristal que su Material Derivado es el Silice, que posee una Dureza de 6 o 7 y no es afectado por rayones
- **Zafiro** Clase definida por Equivalencia, compuesta por todas aquellas instancias de Cristal que su Material Derivado es el Zafiro, que posee una Dureza de 9 y no es afectado por rayones
- **Cristal Mineral** Clase definida por Equivalencia, compuesta por todas aquellas instancias de Cristal que su Material Derivado es el Silice, que posee una Dureza de 5 y es plausible a rayones

Para representar Correa, lo hacemos mediante su clase homóloga Correa que va aportar información con respecto a cualidades intrínsecas de los distintos tipos de correas. A esto último lo hacemos mediante Data Properties:

- **Largo** En su representación es un Entero que hace referencia al largo de la correa en mm
- **Cierre** Es una String en relacion al broche que usa la correa para cerrar

En si esta Clase Correa tiene subclases hijas que cada una están bien definidas y aportan información extra:

- **Eslabones** Intenta representar a aquellas correas compuestas con eslabones metalicos. Para ello agrega 2 data properties que hacen referencia a la cantidad de eslabones que tienes y de qué material están hechos
- **Cuero/Piel** Busca representar las correas confeccionadas con Cuero o Piel de animales, en este caso se le agrega una data property que referencia al origen de ese producto animal

- **Caucho** Representa las correas plásticas, para ello en esta clase hija agregamos una Data property del tipo string que referencia a su color

Para representar el Movimiento o la mecánica del reloj, lo hacemos mediante su clase homóloga Movimiento que va aportar información con respecto a cualidades comunes a los distintos tipos de Movimiento. A esto último lo hacemos mediante Data Properties:

- **Calibre de Maquina** En su representación es un Entero que hace referencia al tamaño del mecanismo en mm
- **Origen** Es una String en relacion a su pais de origen.
- **Fabricante** Es una String en relacion a quien es su fabricante. Cabe aclarar que un reloj puede ser de una marca X pero su maquina puede ser de un fabricante Y

En si esta Clase Movimiento tiene subclases hijas que cada una están bien definidas y aportan información extra:

- **Automatico** Representa a aquellas maquinas precisas que puramente se mueven por cuerda o por el constante movimiento de su rueda de rotor. Tiene como informacion extra la cantidad de joyas que tiene y el tamaño de su volante
- **Quartz** Representa a las maquinas que usan pilas como principal fuente de movimiento. Como data property de esta clase hija es la duracion de su pila en horas
- **Mixto** Esta clase hija de Mecanismo tiene la particularidad de ser hija de Quartz y Automatico. Son aquellos relojes que utilizan pilas pero a su vez pueden tener un complemento Mecanico.

Para representar el *Dial*, lo hacemos mediante su clase homóloga Dial que va aportar información con respecto a cualidades intrínsecas de los distintos tipos de correas. A esto último lo hacemos mediante Data Properties:

- **Forma** En su representación es una String que busca capturar la forma que tiene el mismo. Ej Rectangular
- **Tamaño** Es un entero que captura el tamaño del mismo en mm
- **Calendario** Es un Bool que denota si el dial tiene o no calendario
- **Tamaño de Numeros** Es un entero que busca capturar la medida o el tamaño que van a tener los numero en el dial independientemente si es digital o analogico

En si esta Clase Dial tiene subclases hijas que cada una están bien definidas y aportan información extra:

- **Digital** Representa a los diales digitales. Y como data property distintiva ponemos un booleano que indica si este dial digital tiene o no una luz de pantalla
- **Analogico** Captura a aquellos diales que tienen agujas. En la misma subclase tenemos informacion extra sobre la cantidad de agujas que este dial usa y si usa numeros Romanos o Indoarabigos para denotar la hora como un booleano

Esta clase tiene como subclases hijas a Reloj Deportivo, Reloj Formal y Reloj Lujo. Que al ser clases definidas por equivalencia vamos a poder dejar al razonador inferir sobre las instancias de reloj con todas sus object properties correctamente instanciadas y poder gracias a las inferencias clasificarlas.

Las clases de *Reloj* tienen predicados específicos:

- **Deportivo**
(tieneCorrea some Caucho)
and (tieneDial some Digital)
and (tieneMovimiento some Cuarzo)
- **Formal**
(tieneCorrea some Eslabones)
or (tieneCorrea some Cuero/Piel)
- **Lujo**
Formal
and (tieneCorrea some Cuero/Piel)
and (tieneCristal some Zafiro)

3 Object properties

Estas caracterizan al reloj por sus elementos, y cada una puede tener una cantidad de instancias, al rededor de cinco.

Las propiedades, y las instancias correspondientes:

- **tieneCorrea** CorreaAcero, CorreaCocodrilo, CorreaDeportiva, CorreaOro, CorreaVacuna
- **tieneCristal** CristalHard, CristalMineral, CristalNoClasificado, CristalPlex
- **tieneDial** CristalZafiro, DialCasio, DialCronografo, DialFechador, DialFormal, DialSimple
- **tieneMovimiento** 27Jewels, PilaCitizen, PilaMiyota, PilaSeiko, Tissot-Mixto

3.1 Instancias

3.1.1 Instancias las características

Estas instancias no tienen mayor complejidad que generar elementos validos de cada clase, se crearon diferentes versiones de cada elemento y se busco distribuir homogéneamente las instancias creadas en las subclases.

3.1.2 Instancias de Reloj

Estos fueron creados para evaluar el razonador en las diferentes subclases de Reloj:

- **RelojDePetru** Es *Deportivo*, esto se debe a que tiene correa de caucho, dial digital y movimiento de cuarzo.
- **RelojDeLucho** Es simplemente un *Reloj*, esto se debe a que tiene correa de caucho y movimiento automático, entonces no entra en las otras categorías.
- **RelojDeValle** Es *Formal*, esto se debe a que tiene correa de eslabones.
- **RelojDeAlfredo** Es de *Lujo*, esto se debe a que tiene correa de cuero y es formal.
- **RelojDeMessu** Es *Formal*, a pesar de no tenga ningún movimiento conocido, esto confirma fuertemente la asunción de mundo abierto.
- **RelojDeYoyi** No tiene mayor significancia.

4 DL Query

4.1 Consulta 1

Todos los relojes que tienen correa de cuero o piel.

```
Reloj and (tieneCorrea some Cuero/Piel)
```

Resultado:

```
RelojDeAlfredo
```

4.2 Consulta 2

Todos los relojes que tienen movimiento fabricado por Rolex.

```
Reloj and (tieneMovimiento some (fabricanteMovimiento value "Rolex"))
```

Resultado:

```
RelojDeAlfredo
```

```
RelojDeLucho
```

```
RelojDeYoyi
```

4.3 Consulta 3

Todas las instancias de relojes que tienen movimiento fabricado por Citizen y su marca es Citizen.

```
Reloj
  and (tieneMovimiento some (fabricanteMovimiento value "Citizen"))
  and (Marca value "Citizen")
```

Resultado:

RelojDeValle

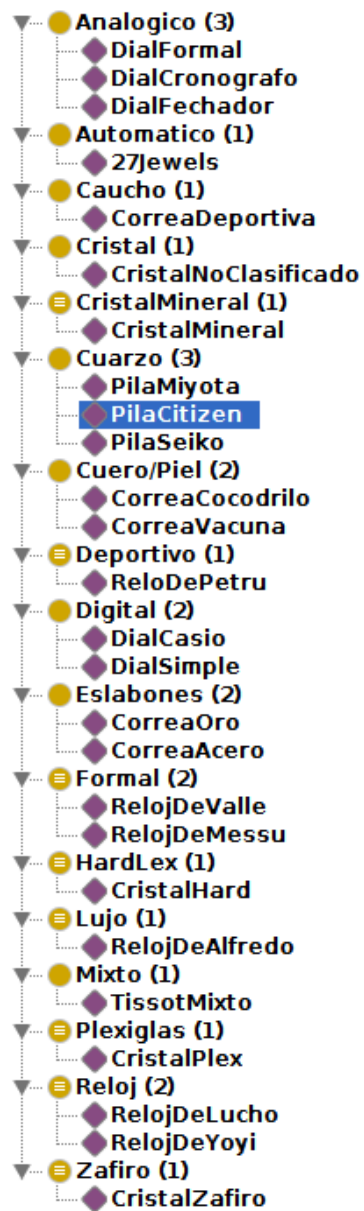
5 Conclusiones

Habiendo creado esta ontología podemos ver la utilidad que provee a personas que no están familiarizadas con un tema específico en detalle pero necesitan disponer de la información. Pero también nos encontramos con dos límites:

- **Limitaciones de la asunción de mundo abierto** no se pueden hacer queries sobre la no existencia de instancias de clase/object properties/data properties, porque la asunción de mundo abierto genera que si una instancia no está dada o la propiedad no está dada, se asume que se desconoce y nunca que no existe.
- **Limitaciones del sistema de queries** No se pueden hacer queries por las limitaciones inherentes al lenguaje usado en OWL, por ejemplo: “todas los relojes producidos por el mismo fabricante de su máquina” no es expresable.

A pesar de que esta ontología es muy limitada, la capacidad de expresión está clara, se puede decidir sobre los componentes y las propiedades de cada reloj hasta un punto muy generalista, pudiendo agregar tantos como sean necesarios para el caso de uso particular de cada usuario y esto aumenta la capacidad de razonamiento del usuario enormemente.

6 Esquema de instancias por relaciones



7 Grafo de la ontología

