

Descripción de la ontología

La ontología se compone de tres partes diferenciadas:

- *Entity*
- *Amount*
- *Pollutant*

El primer elemento trata de englobar los actores que tomarán algún rol activo en el sistema. El segundo elemento se utiliza para representar cualquier tipo de cantidad y, en concreto, la cantidad de un contaminante (*pollutant*). En general las cantidades tienen unidad, pero las estamos obviando y lo añadimos en la descripción del mismo (en concreto los volúmenes se miden en metros cúbicos y los contaminantes nos referiremos por su concentración en gramos por metro cúbico).

Finalmente tenemos a *Pollutant*, que lista los diferentes indicadores de contaminantes que consideraremos en nuestro sistema. Por ahora solo tendremos en cuenta los sólidos en suspensión (que aparece como un individuo de esa clase).

También se han definido algunas propiedades de objeto y de datos:

- **ObjectProperties:**
 - o **hasPollutant:** para relacionar un objeto con un contaminante (por ejemplo, masas de agua o la producción de una industria).
 - o **pollutantGenerated:** para relacionar una industria con una cantidad de contaminante
 - o **pollutantTreated:** para relacionar una planta de tratamiento con la cantidad de contaminante que puede eliminar
- **DataProperties:**
 - o **hasValue:** nos permite relacionar un elemento con un valor numérico
 - **storageAvailability:** un caso concreto que relaciona un elemento con la capacidad de almacenamiento que tiene de agua (en metros cúbicos). Para nuestro caso específico esos elementos que pueden almacenar se restringen a plantas de tratamiento
 - o **maximumProduction:** nos permite relacionar una industria con la cantidad de toneladas máximas que puede producir en un tick
 - o **wastePerProduction:** nos permite relacionar una industria con el volumen de agua que genera por tonelada producida (es decir metros cúbicos por tonelada)

Por último, hemos definido algunas instancias para ejemplificar:

- **SuspendedSolids:** Representa el indicador de sólidos en suspensión en una masa de agua
- **FoodSuspendedSolidsGeneration:** Representa el concepto “cantidad de sólidos en suspensión que genera una industria del sector alimentario”. En este caso esta instancia se relaciona con **SuspendedSolids** mediante **hasPollutant** y con el valor 50.0 mediante **hasValue**. Es decir, una industria

alimentario genera unas masas de agua residuales cuya concentración de sólidos en suspensión es siempre de 50 gr/m^3 . El volumen de esa masa dependerá de las toneladas que produzca y se calcula por la relación `wastePerProduction` que la industria concreta tenga.

- Tenemos luego una serie de industrias alimentarias: Por ejemplo, `Morrich`. Todas tienen una relación con `FoodSuspendedSolidsGeneration` mediante la propiedad `pollutantGenerated`. También tienen definidos su `maximumProduction` y `wastePerProduction` mediante las correspondientes `dataProperties`.
- Por último, tenemos la instancia de `PlantaBesos`, que representa la planta de tratamiento que existe en el río Besós. Esta instancia tiene una capacidad de tratamiento de sólidos en suspensión definida por la propiedad `pollutantTreated`, que lo relaciona con `PlantaBesosSuspendedSolidsTreated`.
- `PlantaBesosSuspendedSolidsTreated` representa la capacidad de limpiar un contaminante de la `PlantaBesos`. En concreto sabemos que trata el contaminante `SuspendedSolids` por la relación `pollutantTreated`, que apunta a `SuspendedSolids`. También sabemos que tiene la capacidad de eliminar el 20% de los sólidos en suspensión (en este caso el valor es un porcentaje que es la cantidad que eliminará del total que le venga, independientemente de la concentración o volumen).

Ejercicios

1. Formad grupos de 3 o 4 alumnos.
2. Abrid la ontología en Protégé y examinadla.
3. Cread un proyecto Java nuevo. incluid las librerías necesarias para utilizar JENA y JADE en el proyecto.
4. Cargad la ontología mediante JENA. Añadid nuevas industrias alimentarias a la ontología. `Hacendoso` y `SanManuel`. Hacedlo desde JENA.
5. Leed las instancias existentes de Industria y planta de tratamiento. Específicamente, que cantidad de contaminantes emiten las industrias y que cantidad son capaces de tratar las plantas de tratamiento.
6. Implementad el código necesario en la clase `WWTPJENA` dentro de vuestro proyecto.
7. Generad instancias de agentes JADE Industria a partir de los datos consultados mediante SPARQL. La clase `AgenteIndustria` recibe como parámetro la cantidad de contaminantes emitidos, según la Ontología. Añadid más parámetros si lo consideráis necesario
8. Generad instancias de agentes JADE Planta de tratamiento a partir de los datos consultados mediante SPARQL. La clase `AgentePlantaTratamiento` recibe como parámetro la cantidad de contaminantes que puede tratar, según la Ontología. Añadid más parámetros si lo consideráis necesario
9. Involucrad a los agentes en un protocolo CFP (Call for proposals) mediante los recursos disponibles en JADE. La instancia de `AgentePlantaTratamiento` contacta con las instancias de `AgenteIndustria` para recibir propuestas con cantidades de contaminantes. La instancia de `AgentePlantaTratamiento` decide que propuestas puede aceptar y cuáles no, y envía las respuestas correspondientes.

10. Debéis entregar los siguientes ficheros:

- a. Ontologia.owl: Fichero con la ontología mejorada, según consideréis necesario
- b. WWTPJENA.java: Como se define en puntos anteriores.
- c. AgenteIndustria.java: Como se define en puntos anteriores.
- d. AgentePlantaTratamiento.java: Como se define en puntos anteriores.
- e. Nombres.txt: Fichero con los nombres y DNIs de los integrantes del grupo
- f. Es necesario que todos los integrantes del grupo realicen la entrega. Todos los ficheros entregados deben ser idénticos dentro de un grupo.
- g. No olvidéis añadir comentarios al código justificando las decisiones de diseño de Software que hayáis tomando.