Problema 2 de 4 - SID 2223q2

Decisiones que hemos tomado para implementar la ontología y el behaviour.

Lo primero que hemos tenido en cuenta para desarrollar la ontología es el dominio y la utilidad que queremos cubrir con ella. A partir de la primera creación de clases, fuimos añadiendo subclases, atributos o propiedades de forma iterativa. En nuestro caso, consideramos que para cubrir el dominio y las necesidades de la práctica basta con representar:

1) Las entidades **(Entity)**, que tendrán como subclases los agentes, los gólems y el pozo. Esta no fue la primera representación que escogimos, ya que al inicio el pozo no formaba parte de esta clase entidades. Escogimos finalmente añadirlo ya que deja "rastro" (**hasTraceFrom**) de su localización en el mapa de igual manera que lo hacen los agentes y podría interpretarse como un agente estático, así que nos pareció interesante agruparlos en el mismo conjunto. Destacar que estas subclases de entidades son disjuntas entre sí (igual que el resto de subclases), por lo que jamás podrá haber un agente que por ejemplo también sea gólem, o nuestra ontología inferirá que hay una inconsistencia.

2) Otra de las representaciones que escogimos en nuestra ontología son los nodos **(Node)**. Los nodos son realmente importantes en nuestra ontología ya que representarán la topología del mapa. A parte de las relaciones que representan las conexiones entre nodos, escogimos hacer dos subclases para identificar un nodo que en el que está situado un agente almacenador, y un nodo en el que se puede descargar. La primera de las subclases la creímos necesaria para justamente poder identificar cuando un agente recolector puede descargar en un nodo (esto será siempre que haya un agente almacenador en el nodo contiguo, propiedad **nextTo**).

3) Otra clase representada es la clase referente a los recursos **(Resource)**. Esta clase es la que nos representa los diferentes recursos del mapa. Tiene como subclases los dos tipos de recursos.

Tanto los recursos como las entidades se encuentran en un nodo, por tanto tenemos la relación isAt para representar esto. La propiedad de **isAt** para el pozo nunca se añade a la ontología ya que para ello deberíamos visitarlo y nuestro agente moriría. Hemos planteado algunas opciones para esta problemática pero para este segundo problema no están implementadas.

Estas clases, como hemos comentado, son las que hemos considerado básicas para la representación del "mundo" de la práctica y la utilidad y respuestas que queremos obtener de ella, como podría ser por ejemplo, si un nodo X es un nodo en el cual el agente Y puede descargar.

Comentar también que hemos pensado en añadir a nuestra ontología las clases: Plan, Message y Goal. Sin embargo, hemos considerado que para esta entrega no eran necesarias pero que creemos serán añadidas para las futuras entregar en las que nuestro agente se convierta en un agente BDI e interaccione con más agentes.

Para la parte de implementación del behaviour, se plantea un agente non-Dedale que explore el mapa hasta que haya cerrado todos los nodos. Básicamente, al no estar trabajando con agentes BDI's, se plantea un action que haga esto mismo. A medida que el agente explora el mapa, va añadiendo la información relevante para la ontología, por ejemplo: si hay rastro de viento (caso especial en el que el agente cierra el nodo para no correr peligro en caer al pozo), si hay recursos, etc. Una vez acabada la exploración, la ontología queda totalmente actualizada. Cabe destacar también que, según pedido para este segundo problema, el agente carga la ontología al inicio de la ejecución.
A parte de este behaviour también hemos desarrollado behaviours para añadir un individuo a la ontología, una propiedad, para eliminar una propiedad, para cargar la ontología y para liberarla.

Reparto del trabajo
Este segundo problema lo hemos desarrollado conjuntamente.