Assignment 1 Report

Description

Nowadays, the car manufacturing is constantly changing. New cars with new technologies are introduced to the market. Thus, the selection of a good car could be difficult and time consuming. The customers need easily to be able to collect basic information about cars in general and their proper usage, i.e., which are the best cars for business and personal usage. Not only the customers, but the car manufacturing companies need to collect information about their car usage and use this information as feedback to adjust their future products.

Therefore, the Car ontology has been created to provide that information, by integrating the ontology into an application. The Car ontology provides basic information about different car brands from several countries around the world, for instance the type of fuel that specific cars could have. The most important thing about the ontology, despite all the basic characteristics that a car could have, is the purpose for which the different cars are used. Through the Car ontology, the customers and the companies will be able to receive this kind of information, affecting, so, not only the marketability but also the manufacturability of cars.

Competency Questions

The first and most important thing, before starting creating the ontology, is to determine the main score and domain of the ontology. To do that, one way to define the scope is a list of questions that the ontology should be able to answer. These questions called competency questions. Here are some questions that need to be answered through the Car ontology:

1. Which cars are Japanese?
2. Is Ford an American car?
3. How many cars are automatic and hybrid?
4. Are German cars suitable for private usage?
5. Find all Italian cars.
6. Which cars are mostly used for business?
7. What is the least popular car for private usage?
8. Show me all engine locations.
9. Are most of the French cars diesel?
10. Are KIA cars belong to front or rear wheel drive?

Description of the steps

Step 1

In step 1, the main classes and subclasses were created for the ontology. First of all, the class named Car with 4 subclasses (USA, Europe, Korea and Japan) was created. Then, the subclass Europe was divided into 3 subclasses named Germany, Italy and France. Finally, the last subclasses, which are the car brands

for each country, were added. For the America class, the subclasses Ford, Chevrolet, Tesla and Jeep were created. For the Germany class, the subclasses Mercedes, BMW, Audi, Volkswagen and Opel were added. The subclasses Alfa Romeo, Fiat and Renault, Peugeot, Citroen divided the classes Italy and France, respectively. Afterwards, the car brands Hyundai and Kia were introduced as subclasses for the class Korea and finally the class Japan split up into the Japanese car brands (Toyota, Nissan, Honda, Mazda, Suzuki, Mitsubishi).

Step 2

After the completion of the main Car class, in which are contained several car brands for different countries, the classes for the car description were created. More specifically, the classes Aspiration, DriveWheels, EngineLocation, TransmissionType and FuelType were constructed. The classes, in turn, were divided into different subclasses. The Aspiration class was divided into the Natural, Super and Turbo subclasses. The DriveWheels class was divided into the 4wd, Rwd and Fwd subclasses. The 4wd, Rwd and Fwd indicate the four-wheel drive, the rear-wheel drive and the front-wheel drive, respectively. The EngineLocation class was split into the Front and Rear subclasses. The TransmissionType class was partitioned into the Automatic and Manual subclasses. Finally, the FuelType class was separated into the Petrol, Diesel, Electric and Hybrid subclasses.

Step 3

The last step for the classes' creation, is the creation of the class CarUsage. This class indicates the purpose for which the different car brands are used for. The class CarUsage has 2 subclasses, the Private subclass and the Business subclass.

Step 4

Upon the completion of the classes' creation, the object properties were added to the ontology. In total, 6 object properties were created, hasAspiration, hasDrivewheels, hasEnginelocation, hasFueltype, hasTransmissiontype and isUsedfor. The object property hasAspiration has domain the Car class and range the Aspiration class. The object property hasDrivewheels has domain the Car class and range the DriveWheels class. The object property hasEnginelocation has domain the Car class and range the EngineLocation class. The object property hasFueltype has domain the Car class and range the FuelType class. The object property hasTransmissiontype has domain the Car class and range the TransmissionType class. The object property isUsedfor has domain the Car class and range the CarUsage class.

Step 5

The last step for the formulation of the Car ontology, is the creation of the data properties. 4 data properties named name, year, horsepower and engine_cylinders were introduced to the ontology. The data property name has domain the Car class and range string. The data property year has domain the Car class and range int. The data property horsepower has domain the Car class and range int. The data property engine_cylinders has domain the Car class and range int.

Enumeration and exemplification of the OWL features

Now that the initial RDF(S) ontology was created, several OWL features were used to extend the ontology. The OWL constructs, which were used, are:

1. disjointness
2. logical connectors
3. property constraints in values
4. cardinalities

Regarding the disjointness, the subclasses USA, Europe, Korea and Japan are disjoint because there is no car brand that can be both American, European, Korean and Japanese. The disjointness was, also, used for the countries France, Germany and Italy and for each and every car brand. An additional example that the disjointness was used, is for the EngineLocation class and the 2 subclass Front and Rear. There is no car that has its engine front and rear. The same thing applies for the subclasses Fwd (front-wheel drive) and Rwd (rear-wheel drive).

Regarding the logical connectors, the owl: unionOf (= logical OR) was used in the classes Aspiration, DriveWheels, EngineLocation, TransmissionType, FuelType. For instance, for the class Aspiration the logical OR was used as Natural OR Super OR Turbo because we want every individual for the class Aspiration will be part of the subclasses Natural or Super or Turbo. The same logic was applied to the rest of the classes, which were mentioned before.

Regarding the value constraints, the owl: allValuesFrom (= only) was used. For instance, the class Car hasAspiration only Aspiration. Another example is the class Car hasEnginelocation only EngineLocation. Thus, wherever there is a hasAspiration or a hasEnginelocation property, all the values must derive from the classes Aspiration or EngineLocation, respectively.

Finally, regarding the cardinalities, one example is that Car isUsedfor min 1 CarUsage. This means that each car can be used for at least one usage. An additional example is Car hasTransmissiontype max 2 TransmissionType. This indicates that a car can have max 2 types of transmission. Automatic, manual or both. Finally, a further example could be Car hasEnginelocation exactly one EngineLocation. For this cardinality, a car has exactly one engine which locates front or rear.