

Informe Practica 1 – Sistemas de recuperación de información WEB

Ontología: Computador

Autores:

Santiago Salazar Ramírez

Deyner López Pineda

Miguel Henao

Esteban Rojas

Sergio Zambrano

Introducción

Esta ontología tiene el objetivo de explicar las distinciones entre un computador que se denomina “gamer” y el común denominador llamado de uso común. En principio se define el computador y lo qué es gracias a lo que lo compone, es decir, gracias a la parte de lo que es un computador. Esto es así para primero definir un computador desde su forma general a su forma particular, para poder atacar de una forma efectiva el tema de la definición. Luego de definir el computador por partes se lleva a algunos ejemplos de lo que podría ser un computador según su categoría (gamer y oficina), para luego definir cada categoría y ver el cómo se comporta la ontología al definir un computador de una forma diferente a la otra.

Existen algunas limitaciones en esta ontología, como lo es la dificultad del lenguaje para explicar un fenómeno que de por sí es ambiguo, dado que la palabra computadora gamer se refiere a una recolección de objetos que tienden a una delgada línea dependiendo de la definición otorgada por el individuo, entonces es una limitación para una representación más limpia y deseable del lenguaje

Diseño y construcción de la ontología

Alcance:

La definición y construcción de lo que determina y es un computador, se usaría para determinar el tipo de computadora. El uso de esta ontología sería adecuado para los vendedores de computadoras que quieren determinar el tipo de computadora que tienen en su almacén. Una computadora gamer se define si posee una gráfica dedicada o si posee un CPU de 8 núcleos, mientras que la computadora de oficina se define por su GPU integrada, bajo rendimiento de los componentes y solo para uso de programas ofimáticos.

Reúso:

La razón por la cual se acopla otra ontología es para la utilización de otros recursos para la ontología creada, se contempla la opción de ahorrar tiempo al utilizar una ontología ya fabricada. La ontología también es de computador.

Términos:

- Computador: Máquina electrónica que, mediante determinados programas, permite almacenar y tratar información, y resolver problemas de diversa índole.
- Oficina: Departamento donde trabajan los empleados públicos o particulares.
- Videojuego: Juego electrónico que se visualiza en una pantalla.
- Tarjeta madre: es la placa principal en la estructura interna del computador donde se encuentran los circuitos electrónicos, el procesador, las memorias y las conexiones principales.
- Procesador: Componente electrónico donde se realizan los procesos lógicos.
- Disco duro: Elemento de almacenamiento de datos en forma magnética u óptica, constituido por una lámina delgada con forma circular.

- Tarjeta gráfica: Tiene a su cargo el procesamiento de los datos que provienen del procesador principal (CPU o UCP) para convertirlos en información que se pueda representar en dispositivos tales como los monitores y los televisores.
- Portátil: Un computador portátil o laptop es un equipo personal que puede ser transportado fácilmente.

Clases:

- Computador (Se divide por sus tipos y diseños):
 - Computador bajo rendimiento (No necesariamente es antiguo, pero no es lo suficiente para ser usado para videojuegos de alto rendimiento)
 - Computador gamer (Se refiere a un computador especializado para las necesidades modernas que requieren los videojuegos)
 - Computador moderno (Después del 2000)
 - Computador oficina (Satisface las necesidades básicas que debe de tener un computador de oficina)
- Gabinete (La carcasa del computador)
 - Portátil
 - Torre
- Parte Computador
 - Board (Tarjeta madre)
 - ATX
 - EATX
 - Micro-ATX
 - Mini-ITX
 - Tipo_Refrigeracion (Ventilador de la CPU)
 - Aire
 - Líquido
 - Nucleo_CPU
 - Dual Core
 - Octa Core
 - Quad Core
 - Single Core
 - Disco duro
 - HDD
 - SSD
 - GPU (Tarjeta gráfica)
 - Dedicada
 - Integrada
 - RAM
 - Dos GB
 - Cuatro GB
 - Ocho GB
 - Dieciséis GB
 - Treinta Y Dos GB

Propiedades

Object Properties

- es_Componente_De (Se refiere a si un objeto es componente de otro objeto, inverso de tiene_Componente)
 - es_Gabinete_De (Se refiere a si un objeto es un gabinete de otro objeto)
 - es_Parte_De (Se refiere a si un objeto es parte de otro objeto)

- tiene_Coincidencia (Se refiere a si un computador es similar a otro)
 - tiene_Ancestro (Se refiere a si existe un computador de la misma línea generacional anterior a él)
- tiene_Componente (Se refiere a si un computador tiene cierto componente)
 - tiene_Gabinete (Se refiere a si un computador tiene cierto gabinete)
 - tiene_Parte (Se refiere a si un computador tiene cierta parte)
- tiene_Generación (Se refiere a si un computador tiene cierto tipo de generación)
 - tiene_Segunda_Generación (Se emplea para distinguir entre el ancestro directo del computador y el de segunda generación)
 - tiene_Tercera_Generación (Esta vez es para la distinción entre segunda y tercera generación)
 - tiene_Hijo_Generacion (En vez de referirse a un padre se refiere a su hijo, es inverse Functional porque un hijo sólo puede tener un padre)

Property Chain:

- tiene_Segunda_Generacion o tiene_Ancestro
- tiene_Ancestro o tiene_Ancestro

Data Properties

- tiene_Precio (Se usa para determinar el precio de cierto computador)

Restricciones

- Computador: Debe de tener un gabinete y un precio.
 - Computador antiguo: Debe de tener sólo un procesador y 2 gb de ram
 - Computador de bajo rendimiento: Debe de no ser un computador gamer, tiene máximo 1 disco duro, tiene procesador de 2 nucleos y tiene 2 o 4 gb de ram
 - Computador gamer: Debe de tener una gráfica dedicada o ser octacore
 - Computador moderno: No debe de ser un computador antiguo, tener 2, 4 u 8 nucleos y tener 8 o 16 o 32 gb de ram.
 - Computador oficina: Debe de tener una gráfica integrada, tener 2 o 4 nucleos en el procesador pero no 8 y tener 4 u 8 gb de ram. Restricciones de propiedades de objetos:
- es_Componente_De es inverso a tiene_Componente.
- tiene_Coincidencia es simetrico.
- tiene_Ancestro es transitivo
- tiene_Gabinete es funcional
- tiene_Gabinete: Dominio computador y rango gabinete
- tiene_Parte: Dominio computador y rango Parte_Computador

Restricciones de propiedades de datos:

- tiene_Precio tiene dominio en computador y rango en int.

Instancias

- Asus Tuf Gaming: Computador gamer, con 8 gb de ram con precio de 3750000
- ChromeBook 113180: Computador de bajo rendimiento con 2 gb de ram
- Computador gamer con 8 gb de ram
- Computador HP 250: Computador moderno con 8 gb de ram
- iMac G3: Es un computador de torre, tiene refrigeracion de aire, disco duro HDD, grafica integrada, una CPU single core y 2 gb de ram Su ancestro es la Macintosh 128K
- iMac G4: Es un computador de torre, tiene refrigeracion de aire, disco duro HDD, grafica integrada, una CPU single core y 2 gb de ram Tiene ancestro iMac G3, tiene hijo generacion iMac G5.
- iMac G5: Es un computador de torre, tiene refrigeracion de aire, disco duro HDD, grafica integrada, una CPU single core y 2 gb de ram. El ancestro es iMac G4.
- Lenovo IdeaCenter: Es un computador de torre, refrigeracion de aire, CPU dual core, tarjeta madre EATX, disco duro HDD, gpu integrada y 4 gb de ram.

- Lenovo Ideapad 3: Es un computador de torre, refrigeracion de aire, tarjeta madre ATX, disco duro HDD, GPU integrada, CPU Octa core y 8 gb de ram.
- Macintosh 128K: Es computadora de torre, refrigeracion de aire, disco duro HDD, GPU integrada, CPU single core y 2 gb de ram.
- Victus Dh007: Es computadora portátil, refrigeracion de aire, gpu dedicada, tarjeta de madre EATX, cpu octa core, disco duro SSD y 8 gb de ram.
- Victus dlh008: Es computadora portátil, refrigeracion de aire, gpu dedicada, tarjeta de madre ATX, cpu quad core, disco duro SSD y 16 gb de ram.

Limitaciones de la ontología

- Se requiere una mayor cantidad de diseños de computadora si se quiere expandir el diseño de la ontología, por ahora sigue siendo un pequeño mock up de lo que podría ser una explicación de una tienda de computadores.
- Algunas computadoras antiguas no pueden ser definidas bajo los términos asociados en esta ontología, entonces se tendrían dos opciones: extender la ontología para que las partes refieran al objeto o definir el objeto bajo otros conceptos diferentes al sistema de partes preestablecido.
- Hay que recordar que los dispositivos ARM también pueden ser considerados como computadores, esta ontología no tiene en cuenta los celulares o relojes inteligentes, lo cual debe de ser considerado para ser más consistente.
- Se requiere más partes para expandir el diseño de la ontología, ya que se tuvo una apreciación muy general del rendimiento y calidad de cada una de las partes que define un computador

Casos de estudio

- Una empresa de distribución de computadores tiene la tarea de categorizar sus computadoras según los componentes de su dispositivo, tienen el pequeño problema de que algunos clientes tienen el arrebato de decir que las etiquetas para sus computadoras están mal, es decir, que sus computadoras realmente no son gamer o no son de oficina. Los dueños de esta empresa están cansados de la falta de definición para lo que asigne el tipo de computadora, entonces buscan una ontología para poder definir cada computadora.
- Una empresa fabricante de computadoras quiere reconocer el tipo de computadora que está produciendo, para ver si puede cobrar más según el tipo de computadora. Esta empresa quiere verse como una empresa competitiva y seria, quieren seguir las métricas de calidad para que cada proceso sea de calidad. Esperan emplear esta ontología para que su categorización tenga sentido.

Consultas

1.

tiene_Ancestro value iMac_G3

Query (class expression)

tiene_Ancestro value iMac_G3

Execute

Add to ontology

Query results

Subclasses (1 of 1)

owl:Nothing

Instances (2 of 2)

iMac_G4

iMac_G5

Se debe a que directamente iMac_G4 es el siguiente a iMac_G3, como también iMac_G5 es siguiente a iMac_G4, como es transitivo se toma que sus ancestros son ellos dos.

2

tiene_Coincidencia value iMac_G3

Query (class expression)

tiene_Coincidencia value iMac_G3

Execute

Add to ontology

Query results

Subclasses (1 of 1)

owl:Nothing

Instances (3 of 3)

Macintosh_128K

iMac_G4

iMac_G5

¿Por qué se relaciona con toda la línea de Mac antigua? Porque es una relación simétrica y al mismo tiempo es padre de tiene_Ancestro, entonces al final tendrá coincidencia con todo computador que siga su línea de herencia.

3

tiene_Segunda_Generacion value iMac_G3

DL query:

Query (class expression)

tiene_Segunda_Generacion value iMac_G3

Execute Add to ontology

Query results

Subclasses (1 of 1)

owl:Nothing

?

Instances (1 of 1)

iMac_G5

?

Hay que recordar una de las propiedades principales de la segunda generación: es siguiente al ancestro primero, es decir, es anterior al computador que es anterior a él.

4

DL query:

Query (class expression)

tiene_Tercera_Generacion value Macintosh_128K

Execute Add to ontology

Query results

Subclasses (1 of 1)

owl:Nothing

?

Instances (1 of 1)

iMac_G5

?

Es similar a la segunda generación, sólo que el anterior se hace desde segunda generación

5

Query (class expression)

Computador_Antiguo

Execute

Add to ontology

Query results

Subclasses (1 of 1)

owl:Nothing

?

Instances (4 of 4)

Macintosh_128K

?

iMac_G3

?

iMac_G4

?

iMac_G5

?

Recordemos lo que necesita un computador para ser un computador antiguo: Debe de ser computador y tener sólo un procesador. Esas instancias tienen sólo un procesador, entonces cumplen con la normativa de ser computadores antiguos

Conclusiones

La ontología de computador puede llegar a ser útil dentro del contexto empresarial, ya que puede ayudar en la competitividad en una empresa permitiendo la planeación y modelación de un computador, incluso se podría extender los usos de esta ontología para la elicitación de requisitos de un computador para conocer las necesidades de los componentes de los computadores que requiere una empresa. Por ahora existen limitaciones y problemas con la ontología, pero en un futuro esto podría ser resuelto si se mejora el modelamiento de la ontología con una ampliación de los conceptos de las clases y una mejor interpretación del lenguaje natural, ya que este puede llegar a ser muy ambiguo, ello implica que el esfuerzo en reconocer necesidades es un pilar para la creación de ontologías.