

Ontología para bases de datos orientadas a objetos y multimedia

Ontology for object-oriented and multimedia databases

Ana Muñoz^{1,2}, PhD, José Aguilar¹ PhD

1. Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID), Universidad de Los Andes

2. Departamento de Tecnología, Instituto Universitario Tecnológico de Ejido

anamunoz@ula.ve, aguilar@ula.ve

Recibido para revisión 04 de Abril de 2008, aceptado 25 de Agosto de 2009, versión final 09 de Septiembre de 2009

Resumen— Las bases de datos multimedia han tenido un gran desarrollo en los últimos años. El uso efectivo de metadatos multimedia requiere de bases de datos multimedia para administrar, almacenar, buscar y entregar metadatos. Estas bases de datos necesitan tener conocimientos acerca de la ubicación de los datos media y de los metadatos, y debe enlazar descripciones de uso con los requerimientos de recursos media. Todo esto implica una serie de manipulaciones que deben definirse a través de parámetros muy específicos que contemplen su semántica, es decir sus conceptos, operaciones y restricciones. En este artículo se muestra una ontología para representar los conceptos, operaciones y restricciones que tienen las bases de datos multimedia. Adicionalmente, como las bases de datos orientadas a objetos describen el comportamiento de las bases de datos multimedia, también se presenta su descripción ontológica. Finalmente, se validan dichas ontologías a través de un caso de estudio utilizando Protégé.

Palabras Clave— Bases de Datos Orientadas a Objetos, Bases de Datos Multimedia, Objetos Multimedia.

Abstract— Multimedia databases had a great development in recent years. Effective use of multimedia metadata requires multimedia databases to manage, store, search and deliver metadata. This databases needs to have knowledge about the location of the data and the metadata, and link descriptions for use with the media resource requirements. All this involves a series of manipulations that must be defined by very specific parameters that provide the semantics, i.e. its concepts, operations and restrictions. This article presents an ontology to represent the concepts, operations and constraints of multimedia databases. Additionally, like an object-oriented database describes the behavior of a multimedia database, we present its ontological description. Finally, they are validated in a study case using Protégé.

Keywords— Oriented Objects Databases, Multimedia databases, Multimedia object, Ontology.

I. INTRODUCCION

Las bases de datos orientadas a objetos se crearon para aplicaciones como el diseño y fabricación en ingeniería (CAD/CAM, CIM), los experimentos científicos, los sistemas de información geográfica, o los sistemas multimedia. Los requerimientos y las características de estas aplicaciones tienen una estructura de los objetos más compleja, las transacciones son de larga duración, se necesitan nuevos tipos de datos para almacenar imágenes y textos, y hace falta definir operaciones no estándar, específicas para cada aplicación. Las bases de datos multimedia las enfocamos en este artículo como un caso de las bases de datos orientadas a objetos en la que se utilizan los tipos de datos como imagen, video, audio y texto. La International Organization for Standardization and International Electrotechnical Commission (ISO-IEC) ha establecido estándares para los metadatos, proponiendo la descripción de esquemas para multimedia, de tal forma de permitir encontrar datos multimedia por contenido [16]. Un ejemplo de ello es el estándar Moving Picture Experts Group (MPEG-7).

El uso efectivo de metadatos multimedia requiere de una base de datos multimedia para administrar, almacenar, buscar y entregar metadatos. Un Sistema Administrador de Base de Datos Multimedia (SABDMM, siglas en inglés MMDDBMS) necesita soportar tipos de datos multimedia, además de proporcionar los servicios de las bases de datos tradicionales. El SABDMM integra texto, datos, video, imagen (estática y dinámica) y sonido. La naturaleza de los datos multimedia hace que sea importante soportar nuevas funciones, como administración de grandes volúmenes de datos multimedia, los cuales requieren una organización específica sobre un servidor de almacenamiento. Además, el SABDMM debe proporcionar los medios para la recuperación e indexación de la información de manera eficiente y soportar modelos conceptuales de datos multimedia [16]. En

este artículo se muestra un modelo ontológico para las bases de datos multimedia. En una primera parte se muestran algunos trabajos relacionados en el área, luego se muestran los componentes teóricos de las bases de datos multimedia, luego el modelo ontológico para las bases de datos multimedia, y finalmente, un ejemplo que verifica el comportamiento del modelo.

II. ANTECEDENTES

Los primeros SABDMM se basaban en el sistema operativo para almacenar y consultar archivos. Estos eran sistemas ad-hoc que servían principalmente como repositorios. A mediados de los 90 surge la primera generación de los SABDMM comerciales. Estos sistemas permitían manejar diferentes clases de datos y proporcionaban mecanismos de consulta, recuperación, inserción y actualización de datos.

En una segunda generación surgen los sistemas que manejan contenido multimedia, proporcionando tipos de datos complejos de diferentes clases de medios. La orientación a objeto proporciona facilidades para definir nuevos tipos de datos y operadores apropiados para esta nueva clase de datos media como video, imagen y audio. Los SABDMM ven los datos como Objetos-Relacionales Extendidos. Las soluciones más avanzadas al respecto las tienen Oracle, IBM DB2 e IBM Informix, que extienden la administración del SABDOR (Sistema Administrador de Base de Datos Objeto-Relacional) a imágenes, video, audio y objetos espaciales.

MIRROR (Multimedia Information Retrieval Reducing Information Overload) [2] es un SABDMM de investigación, desarrollado para comprender la clase de administración de datos que se requiere en el contexto de librerías digitales multimedia. DISIMA [23], es un sistema de bases de datos para procesamiento de imágenes distribuidas y que permite la consulta de imágenes basada en contenido.

La tercera generación incluye los proyectos que están dirigidos principalmente a la necesidad de aplicaciones para contenido de enriquecimiento semántico. La mayoría de ellos se basan en los nuevos estándares MPEG-7 y MPEG-21 [16]. MPEG-7 es el estándar ISO/IEC 15938 de descripción multimedia. Es un estándar de metadato multimedia basado en XML que propone la descripción de elementos para el ciclo de procesamiento multimedia desde la captura, análisis/filtrado, hasta su uso. MPEG-21 es el estándar ISO/IEC 21000 que define una estructura multimedia abierta que permite el uso transparente de los recursos entre redes y dispositivos utilizados por diferentes comunidades. Un proyecto basado en este estándar es MARS (Multimedia Analysis and Retrieval System) (www.mars.ics.uci.edu), que permite la recuperación de información multimedia y soporta objetos de información multimedia basado en su contenido semántico. Otro proyecto es el MPEG-7 Multimedia Data Cartridge (MDC) [16].

III. BASES TEÓRICAS

A. Bases de Datos Orientadas a Objetos

Las bases de datos orientadas a objetos se crearon para tratar de satisfacer las necesidades de aplicaciones como el diseño y fabricación en ingeniería (CAD/CAM, CIM), los experimentos científicos, los sistemas de información geográfica, o los sistemas multimedia. Los requerimientos y las características de estas aplicaciones tienen una estructura de los objetos más compleja, las transacciones son de larga duración, se necesitan nuevos tipos de datos para almacenar imágenes y textos, y hace falta definir operaciones no estándar, específicas para cada aplicación. La orientación a objetos ofrece flexibilidad para manejar algunos de estos requisitos, y no está limitada por los tipos de datos y los lenguajes de consulta de los sistemas de bases de datos tradicionales [6].

Las BDOO manejan el paradigma de la Programación Orientada a Objeto (POO), cuyos conceptos de base son los siguientes: Clases; Herencia simple, múltiple y repetida; Polimorfismo y Objetos complejos. A continuación se describen estos conceptos:

Objeto: es una instancia auto contenida de una entidad de interés del mundo real. Los objetos tienen un tipo de identificador único. Los tipos de objetos se descomponen en atómicos, colecciones y tipos estructurados. Los objetos pueden ser transitorios o persistentes. Los objetos transitorios existen mientras vive el programa de aplicación que los ha creado. Los objetos persistentes son aquellos que se almacenan en la base de datos [5].

Clase: Es un patrón o plantilla en la que se basan objetos que son similares. Cuando un programa crea un objeto de una clase, proporciona datos para sus variables, y el objeto puede entonces utilizar los métodos que se han escrito para la clase. Todos los objetos creados a partir de la misma clase comparten los mismos procedimientos para sus métodos, también tienen los mismos tipos para sus datos, pero los valores pueden diferir. Una clase también es un tipo de datos.

Herencia. En ocasiones se necesita trabajar con clases que son similares pero no idénticas. Para ello es muy útil una de las características del paradigma orientado a objetos: la herencia. Una clase puede tener varias subclases que representan ocurrencias más específicas de la superclase. Por ejemplo, podemos tener la clase (superclase) Animal con sus atributos (nombre común, nombre científico, fecha de nacimiento y género), y las subclases Mamífero, Reptil y Pez, cada una con unos atributos específicos (Mamífero: peso, altura del hombro, raza y color; Reptil: longitud actual y longitud máxima; Pez: color). Por el hecho de ser subclases de Animal, heredan sus atributos. La relación que mantienen las subclases con la superclase es del tipo "es un": un mamífero es un animal, un reptil es un animal y un pez es un animal. Cuando una clase hereda de más de una superclase se tiene herencia múltiple, y

es repetida cuando hereda una superclase más de una vez, es un caso especial de la herencia múltiple.

Polimorfismo. En general, las subclases heredan los métodos de sus superclases y los utilizan como si fueran suyos. Sin embargo, en algunas ocasiones no es posible escribir un método genérico que pueda ser usado por todas las subclases. Por ejemplo, la clase ObjetoGeométrico posee un método área que debería tener distinta implementación para sus subclases Círculo, Rectángulo y Triángulo. La superclase contendría un prototipo para el método que calcula el área, indicando sólo su interface pública. Cada subclase redefine el método, añadiendo las instrucciones necesarias para calcular su área. A esto último es lo que se le llama polimorfismo.

Métodos: especifican el conjunto de operaciones de una clase. Hay varios tipos de métodos que son comunes a la mayoría de las clases Constructores (es un método que tiene el mismo nombre que la clase. Se ejecuta cuando se crea un objeto de una clase); Destruidores (es un método que se utiliza para destruir un objeto); Accesoros (es un método que devuelve el valor de un atributo privado de otro objeto); Mutadores (es un método que almacena un nuevo valor en un atributo). Además, cada clase tendría otros métodos dependiendo del comportamiento específico que deba poseer.

Interfaces. Algunos lenguajes orientados a objetos no soportan la herencia múltiple. En lugar de eso permiten que una clase se derive de una sola clase, pero permiten que la clase implemente múltiples interfaces. Una interface es una especificación para una clase sin instrucciones en los métodos. Cada clase que implemente la interface proporcionaría las instrucciones para cada método de la misma. Una interface puede contener atributos y métodos, o sólo métodos.

El Modelo de Datos Orientado a Objeto

ODMG (Object Database Management Group) es un grupo de representantes de la industria de bases de datos que tienen como objetivo definir estándares para los SGBDOO. El modelo de objetos ODMG permite que tanto los diseños, como las implementaciones, sean portables entre los sistemas que lo soportan. La última versión del estándar, ODMG 3.0 propone los siguientes componentes principales de la arquitectura ODMG para un SGBDOO:

- Modelo de objetos.
- Lenguaje de definición de objetos (ODL, Object Definition Language).
- Lenguaje de consulta de objetos (OQL, Object Query Language).
- Conexión con los lenguajes C++, Smalltalk y Java (al menos).

El modelo de objetos ODMG permite que tanto los diseños, como las implementaciones, sean portables entre los sistemas que lo soportan. El modelo de datos ODMG dispone de las siguientes primitivas de modelado [5]:

- Los objetos y los literales. Un objeto es una instancia auto contenida de una entidad de interés del mundo real. Los objetos tienen algún tipo de identificador único. Un literal es un valor específico, como "Amparo" o 36. Los literales no tienen identificadores. Un literal no tiene que ser necesariamente un solo valor, puede ser una estructura o un conjunto de valores relacionados que se guardan bajo un solo nombre.

- Los objetos y los literales se categorizan en tipos. Cada tipo tiene un dominio específico compartido por todos los objetos y literales de ese tipo. Los tipos también pueden tener comportamientos. Cuando un tipo tiene comportamientos, todos los objetos de ese tipo comparten los mismos comportamientos. En el sentido práctico, un tipo puede ser una clase de la que se crea un objeto, una interface o un tipo de datos para un literal (por ejemplo, integer). Un objeto se puede pensar como una instancia de un tipo.

- Lo que un objeto sabe hacer son sus operaciones. Cada operación puede requerir datos de entrada (parámetros de entrada) y puede devolver algún valor de un tipo conocido.

- Los objetos tienen propiedades, que incluyen sus atributos y las relaciones que tienen con otros objetos. El estado actual de un objeto viene dado por los valores actuales de sus propiedades. Un atributo se define del tipo de un objeto. Un atributo no tiene identificador, pero toma como valor un literal o el identificador de un objeto. Las relaciones se definen entre tipo, y son relaciones binarias con cardinalidad 1:1, 1:n y n:m.

- La definición de una base de datos está contenida en un esquema que se ha creado mediante el lenguaje de definición de objetos ODL (Object Definition Language).

Las bases de datos orientadas a objetos soportan dos tipos de herencia: la relación "es un" y la relación "extiende". La relación "es un", que también se conoce como generalización-especialización, crea una jerarquía donde las subclases son tipos específicos de las superclases. Con la relación "extiende", una clase expande su superclase en lugar de estrecharla en un tipo más específico [5].

Una clase puede tener una o más claves. Una clave es un identificador único. Cuando una clave está formada por una sola propiedad, es una clave simple; si está formada por varias propiedades, es una clave compuesta.

B. Bases de Datos multimedia

En las Bases de Datos Multimedia se pueden distinguir dos tipos: las referenciales y las descriptivas. Las bases de datos referenciales son bancos de datos sobre material como películas, series de televisión o música. En la mayoría de los casos la información que se almacena hace referencia a cuestiones descriptivas (autor, título, duración, productor, etc.), o a cuestiones técnicas (formato, duración, etc.). Las bases de datos multimedia descriptivas, además de los datos técnicos y generales, aportan información específica sobre el contenido,

indicando, por ejemplo, dónde se sitúan los cambios de plano en una película o la transcripción de un diálogo determinado.

En un sistema de base de datos multimedia, de acuerdo a como se realice el tratamiento de los datos (de manera integrada o no con el sistema de base de datos), se presentan diferentes arquitecturas [24]:

Arquitectura de Acoplamiento separado (loose coupling): Esta arquitectura está conformada por 3 módulos: el sistema manipulador de los metadatos (lo hace un sistema manejador de Bases de Datos), el administrador de archivos multimedia, y el módulo integrador de ambos [24].

Arquitectura de acoplamiento compacto (tight coupling): En este caso, el administrador de bases de datos maneja los archivos multimedia y los metadatos. Así, en esta arquitectura todas las funcionalidades del sistema administrador de bases de datos se aplican sobre la base de datos multimedia [24].

Arquitecturas de Hipermedia: funciona básicamente como un hipertexto. El inicio del enlace o su origen está denotado por un "link marke", y el final del enlace o su destino es un "link anchor".

Los tipos y funciones de los metadatos multimedia pueden clasificarse de acuerdo a su uso [16] en:

Metadatos para Texto. El texto es el objeto media más simple, en términos de requerimientos de almacenamiento, representación, identificación del contenido de información, etc. Los siguientes tipos de metadatos se usan para describir objetos texto:

Representación del texto: describe el formato, técnicas de codificación y compresión utilizadas para almacenar los datos.

Histórico del documento: describe la historia de creación del documento. Este metadato es descriptivo.

Localización del documento: este metadato independiente del contenido, describe la localización en donde se encuentra el documento (estación de trabajo o PC dentro de una red).

Metadatos para el Habla. Se refiere al lenguaje hablado, y no es un tipo de dato independiente, sino que es considerado como parte del audio. Los metadatos generados para el habla pueden ser:

Identificación de palabras habladas. Este metadato consiste del tiempo inicial y final del discurso.

Identificación del Hablante (speaker). Los metadatos generados como parte del reconocimiento del hablante se consideran descriptivos del contenido.

Identificación de la información prosódica que puede ser utilizada para llamar la atención de una frase o una oración, o alterar el significado de la palabra. Los metadatos que describen la información prosódica son dependientes del contenido.

Metadatos para Imágenes. Los metadatos de imágenes dependen del tipo de imágenes a ser analizadas y la aplicación

que se utilizará para analizar los datos. Algunos de estos tipos de imágenes son:

Metadatos para Imágenes Satelitales.

Metadatos para imágenes de diseño arquitectónico.

Metadatos para Imágenes faciales.

Metadatos para Video.

C. Ontología

Una definición de Ontología hecha en términos de base de datos, es la que ofrece Weigand [25]: "Una ontología es una base de datos que describe los conceptos del mundo de un dominio específico, algunas de sus propiedades y cómo estos conceptos se relacionan entre sí". Desde el punto de vista de la Inteligencia Artificial, las ontologías permiten definir un vocabulario con lógica [10]. El conocimiento en las ontologías se formaliza a través de seis componentes[12]:

- Las **clases** o **conceptos** en la ontología se organizan en taxonomías.

- Los **atributos** representan la estructura interna de los conceptos. De acuerdo a su origen se clasifican en específicos y heredados. Los específicos son los propios del concepto al que pertenecen, y los heredados vienen dados por las relaciones taxonómicas en las que el concepto desempeña el rol de hijo, y por tanto, hereda los atributos del padre. Los atributos se caracterizan por el dominio en el cual pueden tomar valor.

- Las **relaciones** representan la interacción y enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Ejemplos son: "subclase de", "parte-de", "conectado-a".

- Las **funciones** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo pueden aparecer funciones como asignar-fecha.

- Los **axiomas** son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Los axiomas permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

- Las **instancias**, son las ocurrencias en el mundo real de los conceptos. En una instancia, todos los atributos del concepto tienen asignado un valor concreto [12].

IV. MODELO ONTOLÓGICO PARA BASES DE DATOS ORIENTADAS A OBJETOS Y BASES DE DATOS MULTIMEDIA

A. Ontología de Bases de Datos Orientadas a Objetos

Las bases de datos orientadas a objetos están basadas en varios conceptos fundamentales [7]: cada entidad del mundo real es modelada como un objeto. Cada objeto está asociado

con un identificador único y tiene un conjunto de atributos y métodos; a su vez, el valor de un atributo puede ser un objeto o un conjunto de objetos. El conjunto de atributos de un objeto representa la estructura, y el conjunto de métodos el comportamiento. El estado de un objeto es accesado ó modificado a través del envío de mensajes al objeto para invocar los métodos correspondientes. Los objetos que comparten la misma estructura y comportamiento se agrupan en clases. Una clase representa una plantilla para un conjunto de objetos similares. Cada objeto es una instancia de una clase. Una clase puede ser definida como una especialización de una o más clases. Una clase definida como una especialización se le llama subclase y hereda atributos y métodos de sus superclases.

A.1 Conceptos de Bases de Datos Orientadas a Objetos

Objeto: es cualquier cosa real ó abstracta acerca de la cual se almacenan datos y los métodos que controlan dichos datos. Un objeto está compuesto por el identificador de objetos, el constructor del tipo de objeto y el valor del objeto.

Identificador de Objetos: Es un concepto o valor que identifica al objeto de manera única y es inmutable. Se le genera al objeto en el momento de ser creado.

Tipo de Objeto: Es un conjunto de objetos que tienen un mismo comportamiento. Los tipos de objeto se definen a través de los constructores de tipo de objeto: átomos, tuplas, conjuntos, listas y arreglos.

Los objetos tienen un Dominio que contienen los valores atómicos básicos disponibles como son: enteros, reales, cadena de caracteres, booleanos, fechas, y cualquier otro tipo de datos que maneja la BDOO directamente. Además, puede tener valores no atómicos, que se refieren a un conjunto de objetos del mismo tipo como son tuplas, listas y arreglos.

Clase: Es la agrupación de objetos parecidos. Especifica una estructura de datos y los métodos operativos permitidos que se aplican a cada uno de sus objetos [5].

Jerarquía de clases: definición de clases a partir de clases predefinidos. Cuando las clases son parecidas se agrupan en subclases.

Herencia: tanto los atributos, como las operaciones de las clases, son heredados por las subclases.

En la figura 1 se muestra la taxonomía de los conceptos para bases de datos orientadas a objetos.

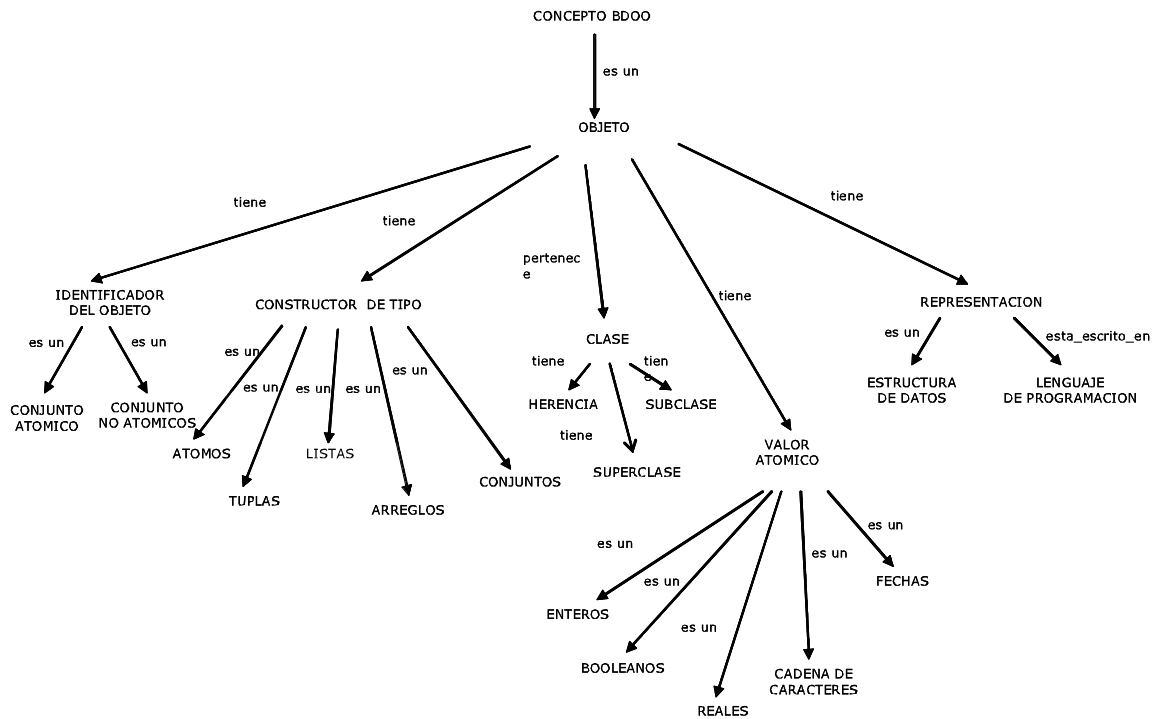


Figura 1. Taxonomía de Conceptos para Bases de Datos Orientadas a Objetos.

Las bases de datos orientadas a objetos poseen los siguientes atributos que describen sus características:

BASE_DE_DATO_ORIENTADA_A_OBJETOS (Id_BDOO, Nombre_BDOO, Dirección, Usuario, Esquema_BDOO, Lenguaje_de_Consulta, SABD)

Donde:

Id_BDOO: identificador de la BDOO, este identificador es único

Nombre_BDOO, donde se coloca el nombre de la base de datos

Dirección: Ubicación de la base de datos

Usuario: Propietarios de la BDOO
 Esquema de la BDOO: nombre de los esquemas que conforman la base de datos orientada a objetos.
 Lenguaje de Consulta: Lenguaje de Consulta que utiliza el Sistema Manejador de Base de Datos. Por ejemplo: SQL

SABD: Sistema Administrador de Base de Datos. Por ejemplo: MySQL, Informix, etc.
 Relaciones: es_un, tiene, pertenece
 Los axiomas de los conceptos para las Bases de Datos Orientadas a Objetos se muestran en la tabla 1.

Tabla 1. Axiomas que conforman los conceptos para las bases de datos orientadas a objetos

| Sentencia | LPO |
|--|--|
| Un concepto de base de datos orientada a objetos es un objeto | $\forall x \text{ ConceptoBDOO}(x) \Rightarrow \text{es_un}(x, \text{Objeto})$ |
| Un objeto puede ser un objeto multimedia | $\forall x \text{ Objeto}(x) \Rightarrow \text{es_un}(x, \text{ObjetoMM})$ |
| Un objeto tiene identificador de objeto, constructor de tipo, y valor atómico | $\forall x \text{ Objeto}(x) \Rightarrow \text{tiene}(x, \text{IdentificadorObjeto}) \wedge \text{tiene}(x, \text{ConstructordeTipo}) \wedge \text{Tiene}(x, \text{Valor atómico})$ |
| Un objeto pertenece a una clase | $\forall x \text{ Objeto}(x) \Rightarrow \text{Pertenece a}(x, \text{Clase})$ |
| Una clase tiene herencia, superclases y subclases | $\forall x \text{ Clase}(x) \Rightarrow \text{tiene}(x, \text{Herencia}) \wedge \text{tiene}(x, \text{Subclases}) \wedge \text{Tiene}(x, \text{Superclases})$ |
| Un identificador de objeto puede ser un conjunto atómico o un conjunto no atómico | $\forall x \text{ IdentificadordeObjeto}(x) \Rightarrow \text{es_un}(x, \text{ConjuntoAtómico}) \vee \text{es_Un}(x, \text{ConjuntoNoAtómico})$ |
| Un constructor de tipo puede ser un átomo o una tupla o una lista o un arreglo o un conjunto | $\forall x \text{ ConstructordeTipo}(x) \Rightarrow \text{es_un}(x, \text{Atomo}) \vee \text{es_un}(x, \text{Tupla}) \vee \text{es_Un}(x, \text{Lista}) \vee \text{es_Un}(x, \text{Conjunto}) \vee \text{es_Un}(x, \text{Arreglo})$ |
| Un valor atómico es un entero, booleano, real, cadena de caracteres, fecha | $\forall x \text{ ValorAtómico}(x) \Rightarrow \text{es_un}(x, \text{Entero}) \vee \text{es_un}(x, \text{Booleano}) \vee \text{es_Un}(x, \text{Real}) \vee \text{es_Un}(x, \text{Cadena}) \vee \text{es_Un}(x, \text{Fecha})$ |
| Una representación es una estructura de datos | $\forall x \text{ Representación}(x) \Rightarrow \text{es_un}(x, \text{EstructuraDatos})$ |
| Una representación está escrita en un lenguaje de programación | $\forall x \text{ Representación}(x) \Rightarrow \text{esta_escrita_en_un}(x, \text{Lenguaje de Programacion})$ |

A.2 Operaciones de Bases de Datos Orientadas a Objetos

Los objetos son manipulados a través de los métodos que consisten de dos partes: un primer componente, llamado signatura o interfaz de la operación, que especifica el nombre del método, los nombres y clases de los argumentos, y los resultados si existen; y un segundo componente, que especifica la implementación de la operación, que es un código escrito en un lenguaje de programación [3].

Las operaciones se invocan pasándole un mensaje a un objeto, que incluye el nombre de la operación y los parámetros. Luego el objeto ejecutará el método para esa operación. Se denomina Solicitud a la invocación de una operación específica, con uno ó más objetos como parámetros. Por otro lado, los objetos tienen una representación que viene dada por una estructura de datos y un lenguaje de implantación. En la figura 2 se muestran las operaciones para las bases de datos orientadas a objetos.

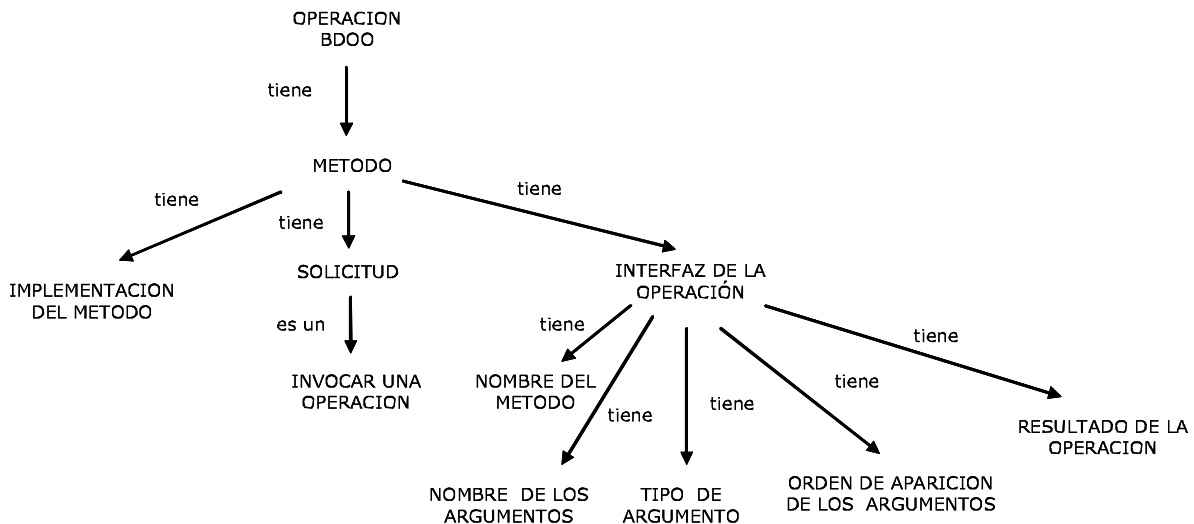


Figura 2. Taxonomía de las Operaciones para Bases de Datos orientadas a Objetos.

Tabla 2. Axiomas que conforman las operaciones para las bases de datos orientadas a objeto

| Sentencia | LPO |
|--|---|
| Una operación de bases de datos orientadas tiene métodos y tiene representación | $\forall x \text{ OperacionBDOO}(x) \Rightarrow \text{tiene}(x, \text{Metodos}) \wedge \text{tiene}(x, \text{Representación})$ |
| Un método tiene implementación del método, tiene solicitudes y tiene interfaz de la operación | $\forall x \text{ Metodo}(x) \Rightarrow \text{Tiene}(x, \text{ImplementaciondelMetodo}) \wedge \text{Tiene}(x, \text{Solicitud}) \wedge \text{Tiene}(x, \text{InterfazdeOperacion})$ |
| Una solicitud es una invocación de una operación | $\forall x \text{ Solicitud}(x) \Rightarrow \text{es_un}(x, \text{InvocaciónOperación})$ |
| Una interfaz de operación tiene nombre del método, tiene nombre de los argumentos y tiene resultados de la operación | $\forall x \text{ InterfazdeOperación}(x) \Rightarrow \text{tiene}(x, \text{NombredeMetodo}) \wedge \text{tiene}(x, \text{NombredeArgumento}) \wedge \text{tiene}(x, \text{TipodeArgumento}) \wedge \text{tiene}(x, \text{OrdendeApariciondelosArgumentos}) \wedge \text{tiene}(x, \text{ResultadosdeOperacion})$ |

A.3 Restricciones de Bases de Datos Orientadas a Objetos

Las bases de datos orientadas a objeto poseen las mismas restricciones que las bases de datos relacionales, con la diferencia de que en lugar de referirse a tuplas y relaciones se refieren al objeto. Cada tipo de objeto tiene sus restricciones de integridad programadas en los métodos que crean, eliminan y actualizan los objetos.

Las restricciones de Integridad Referencial de Objetos vienen dadas por:

- Restricción de dominio, que son las restricciones sobre los valores que pueden tomar las variables que conforman un objeto.

- Restricción de integridad de objeto, establece que ningún valor del identificador de objeto puede ser nulo.

- Asociaciones permitidas, especifican el tipo de asociaciones entre objetos.

Las restricciones de integridad semánticas son aquellas que se establecen en las consultas.

Las operaciones de actualización (insertar, modificar y eliminar) no deben violar las restricciones de integridad especificadas. En la siguiente figura se muestra la taxonomía de las restricciones para las bases de datos orientadas a objetos.

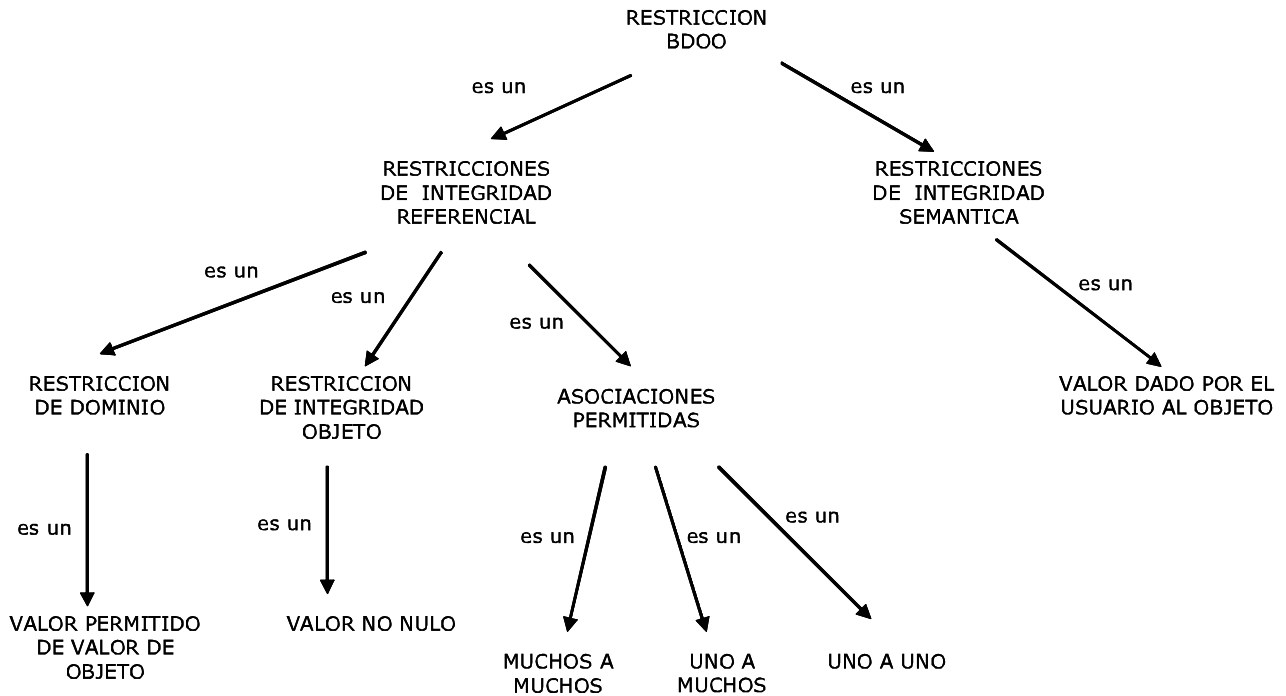


Figura 3. Taxonomía de Restricciones para Bases de Datos Orientadas a Objetos.

Los axiomas de las restricciones para Bases de Datos Orientadas a Objeto se muestran en la tabla 3.

B. Ontología de Bases de Datos Multimedia

En general, las bases de datos multimedia poseen conceptos, operaciones y restricciones como se muestra en la figura 4.

Tabla 3. Axiomas que conforman las restricciones para bases de datos orientadas a objetos.

| Sentencia | LPO |
|--|---|
| Una restricción de BDOO puede ser una restricción de integridad semántica o una restricción de integridad referencial | $\forall x \text{ RestriccionBDOO}(x) \Rightarrow \text{es_un}(x, \text{RestricciónIntegridadSemantica}) \vee \text{es_un}(x, \text{RestricciónIntegridadReferencial})$ |
| Una restricción de integridad semántica es un valor de objeto dado por el usuario | $\forall x \text{ RestricciónIntegridadSemantica}(x) \Rightarrow \text{EsUn}(x, \text{ValordefinidodeObjeto})$ |
| Una restricción de integridad referencial puede ser una restricción de dominio, o una restricción de integridad de objeto, o el tipo de asociación permitida | $\forall x \text{ RestriccionIntegridadReferencial}(x) \Rightarrow \text{es_un}(x, \text{RestricciondeDominio}) \vee \text{es_un}(x, \text{RestriccionIntegridadObjeto}) \vee \text{es_un}(x, \text{AsociaciónPermitida})$ |
| Una restricción de dominio es un valor permitido de objeto | $\forall x \text{ RestricciondeDominio}(x) \Rightarrow \text{es_un}(x, \text{ValorPermitido})$ |
| Una restricción de integridad de objeto es un valor no nulo | $\forall x \text{ RestriccionIntegridadObjeto}(x) \Rightarrow \text{es_un}(x, \text{ValornoNulo})$ |
| Una asociación permitida puede ser del tipo: uno-a-uno o uno-a-muchos o muchos-a-muchos | $\forall x \text{ AsociaciónPermitida}(x) \Rightarrow \text{es_un}(x, \text{Uno a Uno}) \vee \text{es_un}(x, \text{Uno a Muchos}) \vee \text{es_un}(x, \text{Muchos a Muchos})$ |

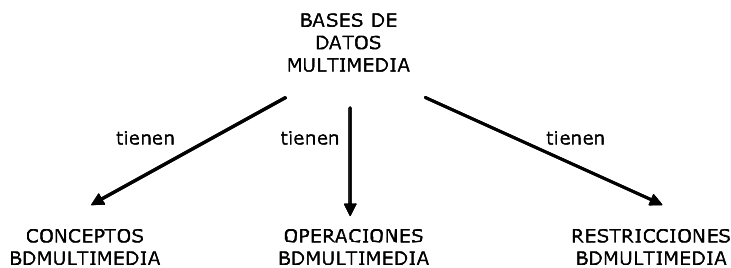


Figura 4. Taxonomía de Bases de Datos Multimedia

B.1 Conceptos de Bases de Datos Multimedia

Los tipos y funciones de los metadatos multimedia pueden clasificarse de acuerdo a su uso [15], [16] en:

- **Metadatos para Texto.** El texto es el objeto media más simple, en términos de requerimientos de almacenamiento, representación, identificación del contenido de información, etc. Los siguientes tipos de metadatos se usan para describir objetos texto:

- **Descripción de contenidos de documentos:** proporciona información adicional del contenido del documento.

- **Representación del texto:** describe el formato, técnicas de codificación y compresión utilizadas para almacenar los datos.

- **Documento Histórico:** describe la historia de creación del documento. Este metadato es descriptivo.

- **Localización del documento:** este metadato independiente del contenido, describe la localización en donde se encuentra el documento (estación de trabajo o PC dentro de una red).

- **Metadatos para el Habla.** Se refiere al lenguaje hablado, y no es un tipo de dato independiente, sino que es considerado como parte del audio. Los metadatos generados para el habla pueden ser:

- **Identificación de palabras habladas.** Este metadato consiste

del tiempo inicial y final del discurso.

- **Identificación del Hablante (speaker).** Los metadatos generados como parte del reconocimiento del hablante se consideran descriptivos del contenido.

- **Identificación de la información prosódica** que puede ser utilizada para llamar la atención de una frase o una oración, o alterar el significado de la palabra. Los metadatos que describen la información prosódica son dependientes del contenido.

- **Metadatos para Imágenes.** Los metadatos de imágenes dependen del tipo de imágenes a ser analizadas y la aplicación que se utilizará para analizar los datos. Algunos de estos tipos de imágenes son:

- Metadatos para Imágenes Satelitales.
- Metadatos para imágenes de diseño arquitectónico.
- Metadatos para Imágenes faciales.
- Metadatos para Video.

Las bases de datos multimedia poseen los siguientes atributos que describen sus características.

Donde:

BASE_DE_DATO_MULTIMEDIA (Id_BDMM, Nombre_BDMM, Dirección, Usuario, Tipo_BDMM, Esquema_BDMM, Modelo_BDMM, Lenguaje_de_Consulta, SABD)

Id_BDMM: identificador de la BDMM, este identificador es único
 Nombre_BDMM, donde se coloca el nombre de la base de datos
 Dirección: Ubicación de la base de datos
 Usuario: Propietarios de la BDMM
 Tipo_BDMM: tipo de base de datos multimedia, por ejemplo de video, de música, híbrida, etc.
 Esquema de la BDMM: nombre de los esquemas que conforman la base de datos multimedia. Por ejemplo: General, Metadata, etc.

Modelo_BDMM: modelo utilizado por la Base de Datos Multimedia para representar sus esquemas. Por ejemplo: Orientados a Objeto, Entidad-Relación.

Lenguaje de Consulta: Lenguaje de Consulta que utiliza el Sistema Manejador de Base de Datos. Por ejemplo: SQL

SABD: Sistema Administrador de Base de Datos. Por ejemplo: MySQL, Informix, etc.

Relaciones: tiene

En la siguiente tabla se muestran los axiomas para esta taxonomía.

| Sentencia | LPO |
|---|--|
| Una base de datos Multimedia tiene conceptos, operaciones y restricciones | $\forall x \text{ BDMultimedia}(x) \Rightarrow \text{tiene}(x, \text{ConceptoBDMM}) \wedge \text{tiene}(x, \text{OperacionesBDMM}) \wedge \text{tiene}(x, \text{RestriccionesBDMM})$ |

Tabla 4. Axiomas de las bases de datos multimedia

B.2 Restricciones de Bases de Datos Orientadas a Objetos

Las La figura5 muestra la representación taxonómica de la ontología de los conceptos de las bases de datos multimedia. A continuación se listan los atributos que poseen los conceptos de la Base de Datos Multimedia, se describen a manera de esquemas de Bases de datos.

BDMM (Nombre, ModeloBDMM)

METADATO_TEXTO (Formato_Texto, Tipo_Font, Tamaño_Pagina, Descripción_Contenido_Texto, Representación_Texto, Documento_Histórico, Localizacion_Documento)

METADATO_HABLA (Formato_Compresion, TiempoInicial_Discurso, Tiempo_Final_Discurso, Nombre_del_Hablante, Información_Prosodica=Significado)

IMAGEN_SATELITAL(Fila, Columna, Capa_Profundidad)

IMAGEN_DISEÑO_ARQUITECTÓNICO(Nombre_Edificación, Ubicación, Nombre_Arquitecto, Costo_Estimado, Nombre_Compañía)

METADATO_IMAGEN_FACIAL(Color_Cabello, Descripción_de_Ojos, descripción_Nariz, Descripción_Boca,

Nombre_Persona, Sexo_Persona, Identificación_Persona)

METADATO_VIDEO (Frames, Formato, Movimiento_de_Camara, Altura_Camara, Nivel_de_Luz, Objetos_en_Secuencia_de_Video, Enfoque_de_Camara, Angulo_de_Toma, Movimiento_de_la_Toma, Descripción_de_la_Acción, Tipo_de_Objetos_en_la_Toma, Fecha_de_Produccion, Nombre_Productor, Nombre_Director)

HERRAMIENTA_SOPORTE_USUARIO (NombreHerramientaSU, SistemaOperativoSU, LenguajeSU)

HERRAMIENTA_SOPORTE_APLICACION (Nombre_HerramientaSA, SistemaOperativoSA, LenguajeSA)

DATO_TEXTO (Titulo, Autor, Afiliacion_Autor, Resumen, Secciones, Subsecciones, Parrafos, Lenguaje_de_Representacion)

DATO_AUDIO (Tipo_Audio, Calidad_Digitalización)

DATO_IMAGEN (Tipo_Imagen, Color, Formato)

DATO_GRAFICO (Descripción, Nombre_Aplicacion)

DATO_VIDEO (Estandar, Esquema_Compresion)

DATO_MEDIA_GENERADO (Tipo_Presentacion, Formato)

Relaciones: tiene, es_un

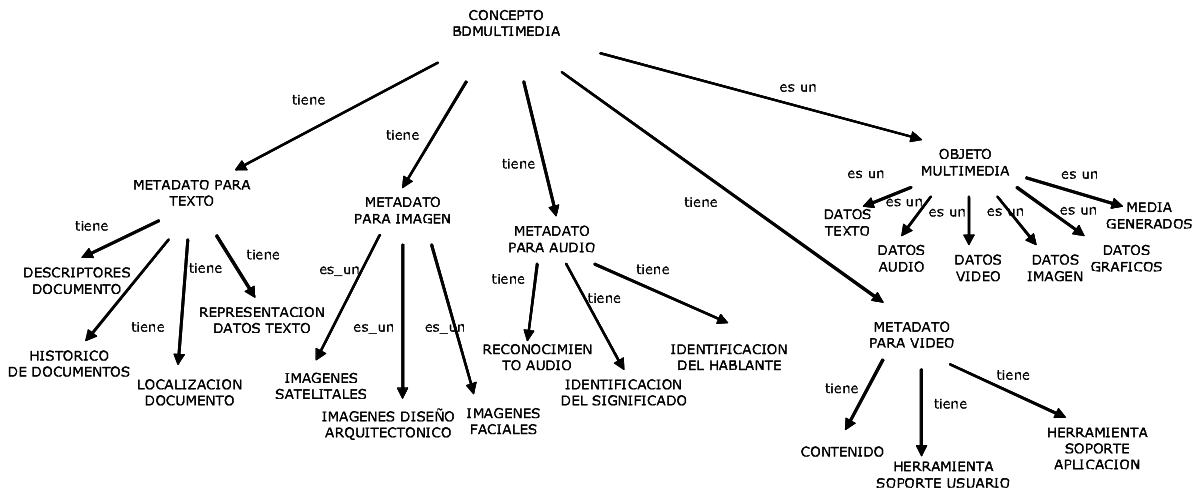


Figura 5. Taxonomía de Conceptos para Bases de Datos Multimedia

Tabla 5. Axiomas de los conceptos de bases de datos multimedia

| Sentencia | LPO |
|---|---|
| Un concepto de base de datos multimedia tiene metadato para texto, metadato para imagen, metadato para audio, o metadato para video | $\forall x \text{ ConceptoBDMultimedia}(x) \Rightarrow \text{tiene}(x, \text{MetadatoTexto}) \vee \text{tiene}(x, \text{MetadatoImagen}) \vee \text{tiene}(x, \text{MetadatoAudio}) \vee \text{tiene}(x, \text{MetadatoVideo})$ |
| Un concepto de base de datos multimedia es un objeto multimedia | $\forall x \text{ ConceptoBDMultimedia}(x) \Rightarrow \text{es_un}(x, \text{ObjetoMM})$ |
| Un metadato para texto tiene descriptores de documento, histórico del documento, Localización del documento y representación de datos texto | $\forall x \text{ MetadatoTexto}(x) \Rightarrow \text{tiene}(x, \text{DescriptorDocumento}) \wedge \text{tiene}(x, \text{HistoricoDocumento}) \wedge \text{tiene}(x, \text{LocalizacionDocumento}) \wedge \text{tiene}(x, \text{Representación_Datos_Texto})$ |
| Un metadato para imagen puede ser una imagen satelital, imagen de Diseño arquitectónico, o una imagen facial | $\forall x \text{ MetadatoImagen}(x) \Rightarrow \text{es_un}(x, \text{ImagenSatelital}) \vee \text{es_un}(x, \text{ImagenDeDiseñoArquitectonico}) \vee \text{es_un}(x, \text{ImagenFacial})$ |
| Un metadato para video tiene contenido de video, herramientas de soporte al usuario y herramientas de soporte de aplicación | $\forall x \text{ MetadatoVideo}(x) \Rightarrow \text{tiene}(x, \text{Contenido_de_Video}) \wedge \text{tiene}(x, \text{Herramientas_Sop_Usuario}) \wedge \text{tiene}(x, \text{Herramientas_Sop_Aplicacion})$ |
| Un metadato para audio tiene reconocimiento del habla, Identificación del significado e identificación del hablante | $\forall x \text{ MetadatoAudio}(x) \Rightarrow \text{tiene}(x, \text{Reconocimiento_del_Habla}) \wedge \text{tiene}(x, \text{Identificación_del_Significado}) \wedge \text{tiene}(x, \text{Identificación_del_Habla})$ |
| Un objeto multimedia es un dato texto, dato audio, dato imagen, dato gráfico, dato video o datos media generados. | $\forall x \text{ ObjetoMM}(x) \Rightarrow \text{es_un}(\text{Dato_Texto}, x) \vee \text{es_un}(x, \text{Dato_Audio}) \vee \text{es_un}(x, \text{Dato_Imagen}) \vee \text{es_un}(x, \text{Dato_Grafico}) \vee \text{es_un}(x, \text{Dato_Video}) \vee \text{es_un}(x, \text{Dato_MediaGenerado})$ |

Los axiomas de los conceptos para las Bases de Datos Multimedia se muestran en la tabla 5.

B.3 Operaciones de Bases de Datos Multimedia

En estas bases de datos, además de las operaciones propias de las bases de datos orientadas a objetos, poseen operaciones

dadas por los tipos de datos que ellas manejan: video, imagen, texto y audio. Las operaciones de las bases de datos multimedia son: representación de datos, consultas y actualizaciones, búsqueda y edición, determinación de calidad de servicios procesados, y administración de seguridad/integridad. Por otro lado, los datos multimedia manejan el tiempo de la siguiente manera:

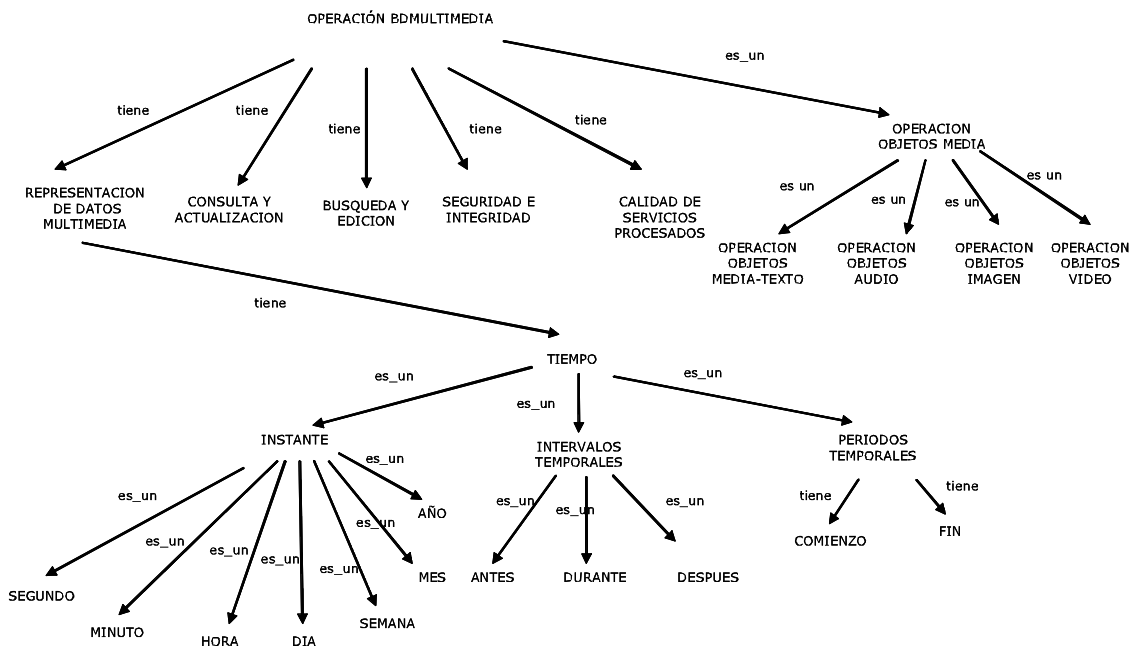


Figura 6. Taxonomía de Operaciones para Bases de Datos Multimedia

- *Instantes*. Representan chronons¹ simples, son utilizados para representar el instante en el que ocurre el hecho.
- *Periodos temporales*. Representa el conjunto de instantes contenidos entre dos instantes específicos comienzo y fin.
- *Intervalos Temporales*. Denota un tamaño de tiempo específico, sin precisar el instante de inicio y de fin. Por ejemplo, un día.
- *Conjunto de instantes*. Es utilizado para representar un conjunto de instantes finitos, no necesariamente contiguos, que eventualmente se repiten.
- *Elementos temporales*. Es un conjunto finito de períodos temporales no necesariamente contiguos.

En la siguiente figura se muestra la taxonomía de las operaciones de las bases de datos multimedia.

A continuación se presentan los atributos para las operaciones de las bases de datos Multimedia

OPERACIÓN_OBJETOSMEDIA (Iniciar_Presentación, Parar_Presentación)

OPERACIÓN_OBJETOSMEDIA_TEXTO (Presentar_Texto)

OPERACIÓN_OBJETOSMEDIA_AUDIO (Presentar_Audio)

OPERACIÓN_OBJETOSMEDIA_IMAGEN (Presentar_Imagen)

OPERACIÓN_OBJETOSMEDIA_VIDEO (Presentar_Video)

Los axiomas de las restricciones se muestran en la tabla 6.

| Sentencia | LPO |
|---|---|
| Una operación de Base de Datos Multimedia es una consulta, una actualización, una búsqueda y edición o una operación de objetos media | $\forall x \text{ OperacionBDMM}(x) \Rightarrow \text{es_un}(x, \text{ConsultaBDMM}) \vee \text{es_un}(x, \text{ActualizacionBDMM}) \vee \text{es_un}(x, \text{BusquedaEdicion}) \vee \text{es_un}(x, \text{Operación_Objetos_Media})$ |
| Una operación de bases de datos multimedia tiene seguridad, integridad y calidad de servicios procesados | $\forall x \text{ OperacionBDMM}(x) \Rightarrow \text{tiene}(x, \text{Seguridad}) \wedge \text{tiene}(x, \text{Integridad}) \wedge \text{tiene}(x, \text{CalidadServiciosProcesados})$ |
| Una representación de datos multimedia tiene tiempo | $\forall x \text{ RepresentacionDatosMM}(x) \Rightarrow \text{tiene}(x, \text{Tiempo})$ |
| El tiempo es un instante, un periodo temporal, o un intervalo temporal | $\forall x \text{ Tiempo}(x) \Rightarrow \text{es_un}(x, \text{Instante}) \vee \text{es_un}(x, \text{PeriodoTemporal}) \vee \text{es_un}(x, \text{IntervaloTemporal})$ |
| Un instante puede ser un segundo o un minuto o una hora o un día o una semana o un año | $\forall x \text{ Instante}(x) \Rightarrow \text{es_un}(x, \text{Segundo}) \vee \text{es_un}(x, \text{Minuto}) \vee \text{es_un}(x, \text{Dia}) \vee \text{es_un}(x, \text{Hora}) \vee \text{es_un}(x, \text{Semana}) \vee \text{es_un}(x, \text{Mes}) \vee \text{es_un}(x, \text{Año})$ |
| El intervalo temporal es un antes, un durante o un después | $\forall x \text{ IntervaloTemporal}(x) \Rightarrow \text{es_un}(x, \text{Antes}) \vee \text{es_un}(x, \text{Durante}) \vee \text{es_un}(x, \text{Después})$ |
| Un periodo temporal tiene un comienzo y tiene un fin | $\forall x \text{ PeriodoTemporal}(x) \Rightarrow \text{tiene}(x, \text{Comienzo}) \wedge \text{tiene}(x, \text{Fin})$ |
| Una Operación de Objetos media es una operación de objetos media texto, objetos media audio, objetos media video u objetos media imagen | $\forall x \text{ Operación_Objetos_media}(x) \Rightarrow \text{es_un}(x, \text{ObjetoMedia_Texto}) \vee \text{es_un}(x, \text{ObjetoMedia_Audio}) \vee \text{es_un}(x, \text{ObjetoMedia_Video}) \vee \text{es_un}(x, \text{ObjetoMedia_Imagen})$ |

Tabla 6. Axiomas que conforman las operaciones para bases de datos multimedia.

C. Restricciones de Bases de Datos Multimedia

Las restricciones vienen dadas por las restricciones de integridad de las bases de datos orientadas a objeto, así como de la ocurrencia de eventos para la presentación de datos, tales

como la representación de espacio y tiempo. Se debe tener en cuenta la ubicación del objeto multimedia en la pantalla. En la figura 7 se muestra la taxonomía para las restricciones de bases de datos multimedia.

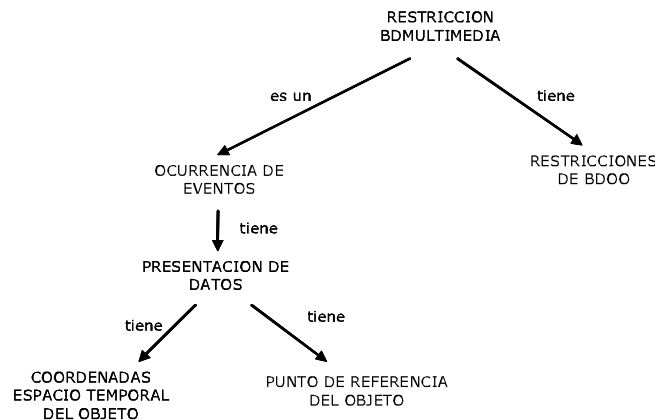


Figura 7. Taxonomía de Operaciones para Bases de Datos Multimedia

1. Unidad indivisible y discreta de tiempo (2x10⁻²³ seg.)

Tabla 7. Axiomas que conforman las restricciones de bases de datos multimedia

| Sentencia | LPO |
|---|--|
| Una restricción de bases de datos multimedia es una ocurrencia de eventos y tiene restricciones de BDOO | $\forall x \text{ RestriccionBDMultimedia}(x) \Rightarrow \text{es_un}(x, \text{OcurrenciaEventos}) \wedge \text{tiene}(x, \text{RestriccionBDOO})$ |
| Una ocurrencia de eventos tiene presentación de datos | $\forall x \text{ OcurrenciaEventos}(x) \Rightarrow \text{tiene}(x, \text{PresentacionDatos})$ |
| La presentación de datos tiene coordenadas espacio-temporal del objeto y tiene punto de referencia del objeto | $\forall x \text{ PresentacionDatos}(x) \Rightarrow \text{tiene}(x, \text{CoordenadasEspacioTemporal}) \wedge \text{tiene}(x, \text{PuntodeReferencia})$ |

Los axiomas de las operaciones se muestran en la tabla 7.

V. CASO DE ESTUDIO: BASE DE DATOS MULTIMEDIA

El ejemplo a considerar se trata del ingreso y consulta de un CURSO, llamado "Gestión de Conocimiento", en la Base de Datos Multimedia "Sistema Multimedia de Control de Curso" (SMCC). El CURSO contiene una colección de módulos, descritos como objetos de aprendizaje, como se muestra en la siguiente figura.

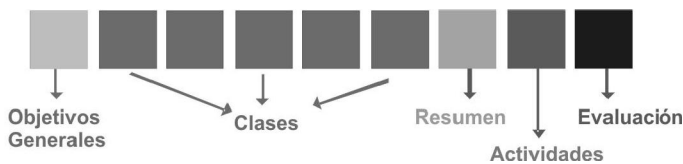


Figura 8. Módulos que conforman un curso

Donde:

Objetivos Generales: describe el curso, está elaborado en XML

Clases: objetos de aprendizaje que contienen los objetivos del curso desarrollados utilizando diferentes tipos de datos multimedia (video, texto, audio y presentaciones)

Resumen: contiene un resumen de cada una de las clases elaborado en XML

Actividades: objetos de aprendizaje que contienen actividades a realizar por el estudiante del curso desarrollados utilizando diferentes tipos de datos multimedia (video, texto, audio y presentaciones)

Evaluación: en el que se tienen varios tipos de archivos (presentaciones y texto)

Una de las operaciones para el objeto de aprendizaje CURSO es una consulta, donde se puede mostrar un elemento de CLASES (se consulta la clase1), un examen, o el listado de los objetivos de la materia. Otra posible operación es una actualización de la materia, por ejemplo cuando se quiere agregar un video (Video01) al objeto de aprendizaje CLASES.

A. Caracterización del caso de estudio en el modelo ontológico

En el caso particular del ejemplo: CURSO de "Gestión del Conocimiento", los atributos de RESUMEN son los siguientes:

General:

Título: Gestión de Conocimiento

Descripción: Permite al alumno adquirir los conocimientos sobre técnicas y herramientas para implementar Sistemas Basados en Conocimiento,

Idioma: Español

Alcance: Este curso está dirigido a estudiantes del 7mo semestre que hayan cursado Inteligencia Artificial y Sistemas de Información

Palabras Clave: Sistemas Basados en Conocimiento, Gestión de Conocimientos, Ontologías, Web Semántica

Educacional

Idioma: Español

Tipo: Tutorial

Tiempo establecido de Aprendizaje: 1 Semestre (16 semanas)

Nivel de Dificultad: Media

Tipo de Recursos de Aprendizaje: Simulación, Animación, Tutorial, Prácticas, Quiz/Test; Lecturas/Presentación, Referencias.

Contexto: Se ubica en el pensum como electiva a partir del 7mo semestre de Ingeniería de Sistemas

Descripción: Recurso que contiene actividades, herramientas, etc., relacionados con sistemas basados en conocimientos

Clasificación

Propósito: Estos objetos se utilizan como herramienta de soporte para la materia Gestión de Conocimiento que se imparte en CEMISID

Descripción: Permite al alumno adquirir los conocimientos sobre técnicas y herramientas para implementar Sistemas Basados en Conocimiento,

Palabras Clave: IMS

Ciclo de vida

Versión: 1.0

Status:

Contribuyente: Profesor

Fecha: 21 de Abril 2007

Carlos García

Contribuyente: Profesor**Fecha:** 21 de Abril 2007

Alba Rojas

Características técnicas

Ubicación: <http://www.cemisid.ula.ve/GC.htm>**Formato:** XML, HTML,**Tamaño:** bytes**Requerimientos:** XML, GC.htm (tiene un enlace para cargar el archivo en la base de datos)**Otros Requerimientos:****Detalles de Instalación:**

Relación

Descripción: Video que muestra procesos en una organización**Recursos:** GC01, Video, VGC01**Clase:** <http://www.cemisid.ula.ve/VGC01.mpeg>; VGC01.mpeg

Derechos

Costo: no**Copyright:** si**Descripción:** Este video requiere establecer los derechos para visualizarlo**FUENTE:** <http://www.cemisid.ula.ve>**HISTORICO:**

A continuación se describen los conceptos de la base de datos multimedia bajo estudio de acuerdo al marco ontológico propuesto en este artículo.

Tabla 8. Axiomas para los componentes conceptuales de la BDMM bajo estudio usando el esquema ontológico de conceptos.

| CONCEPTO | LPO | COMENTARIO |
|---|--|---|
| El objeto de aprendizaje curso Gestión de Conocimiento tiene metadatos texto, metadatos audio, metadatos imagen | ConceptoBDMultimedia (ObjetoAprendizajeGC) => tiene(ObjetoAprendizajeGC, MetadatoTexto) V tiene(ObjetoAprendizajeGC, MetadatoAudio) V tiene(ObjetoAprendizajeGC, MetadatoImagen) V tiene(ObjetoAprendizajeGC, MetadatoVideo) | $\forall x$ ConceptoBDMultimedia(x) => tiene(x, MetadatoTexto) V tiene(x, MetadatoImagen) V tiene(x, MetadatoAudio) V tiene(x, MetadatoVideo) (Axioma tabla 5) |
| El objeto de aprendizaje curso Gestión de Conocimiento es un objeto multimedia | ConceptoBDMultimedia (ObjetoAprendizajeGC) => es_un (ObjetoAprendizajeGC, ObjetoMM) | $\forall x$ ConceptoBDMultimedia(x) => es_un(x, ObjetoMM) (Axioma tabla 5) |
| El metadato texto del curso GC tiene Descriptores de documento, histórico de documento, localización de documento y fuentes | MetadatoTexto (CursoGC) => tiene(CursoGC, Resumen) \wedge tiene(CursoGC, Historico) \wedge tiene(CursoGC, Localizacion) \wedge tiene(CursoGC, Fuente) | $\forall x$ MetadatoTexto(x) => tiene(x, DescriptorDocumento) \wedge tiene(x, HistoricoDocumento) \wedge tiene(x, LocalizacionDocumento) \wedge tiene(x, Representación_Datos_Texto) (Axioma tabla 5) |
| Un objeto de aprendizaje del curso Gestión de Conocimiento puede contener metadatos imagen | MetadatoImagen(CursoGC) => es_un (CursoGC, ImagenSatelital) V es_un (CursoGC, ImagenDiseñoArquitectonico) V es_un (CursoGC, ImagenFacial) | $\forall x$ MetadatoImagen(x) => es_un(x, ImagenSatelital) V es_un(x, ImagendeDiseñoArquitectonico) V es_un(x, ImagenFacial) (Axioma Tabla 5) |
| La materia GC tiene Metadatos audio | $\forall x$ MetadatoAudio(CursoGC) => tiene (CursoGC, ReconocimientoAudio) \wedge tiene (CursoGC, IdentificacionSignificado) \wedge tiene (CursoGC, Identificaciondel Hablante) | $\forall x$ MetadatoAudio(x) => tiene (x, Reconocimiento_del Habla) \wedge tiene (x, Identificacion_del Significado) \wedge tiene (x, Identificacion_del Hablante) (Axioma Tabla 5) |
| El curso GC tiene Metadato Video | $\forall x$ MetadatoVideo(CursoGC) => tiene (CursoGC, Contenido) \wedge tiene (CursoGC, HerramientaSoporteUsuario) \wedge tiene (CursoGC, HerramientaSoporteAplicacion) | $\forall x$ MetadatoVideo(x) => tiene (x, Contenido_de_Video) \wedge tiene (x, Herramientas_Sop_Usuario) \wedge tiene (x, Herramientas_Sop_Aplicacion) (Axioma Tabla 5) |
| El objeto Video01 es un objeto multimedia | Objeto (Video01) => es_un (Video01, ObjetoMM) | $\forall x$ Objeto (x) => es_un (x, ObjetoMM) (Axioma Tabla 1) |
| Video01 es un dato video como mpeg, mov, avi | ObjetoMM (Video01) => es_un (Video01, Dato_Video) | $\forall x$ ObjetoMM(x) => es_un(x, Dato_Texto) V es_un(x, Dato_Audio) V es_un(x, Dato_Imagen) V es_un(x, Dato_Grafico) V es_un(x, Dato_Video) V es_un(x, Dato_MediaGenerado) (Axioma Tabla 5) |

B. Operación de Consulta

La operación de consulta del curso de «Gestión de Conocimiento» se lleva a cabo cuando cualquier estudiante quiera realizar el curso. En esta operación se ejecutan operaciones de consulta a una base de datos multimedia, para lo cual se va caracterizando el objeto de aprendizaje a acceder (duración, tipo, etc.). A continuación, en la tabla 9, se muestran los axiomas requeridos según el modelo ontológico para realizar esa operación.

C. Operación de actualización del sistema con el Video01

A continuación se muestran los axiomas para la operación insertar el Video01.mpeg en la Clase1 de la materia Gestión de Conocimientos. En este caso, se deben definir las operaciones como el tipo de datos a insertar (entre otras cosas), el método que se invocará para realizar la operación de inserción (considerando que es un objeto), etc. Se supone que el software que soporta el video es QuickTime.

Tabla 9. Ejemplo de axiomas para operaciones de consulta

| OPERACIONES | LPO | Comentarios |
|---|---|--|
| Consultar la Clase1 del CURSO Gestión de Conocimientos | OperacionBDMM(ConsultaClase1) => es_un(ConsultaClase1, ConsultaBDMM) V es_un(ConsultaClase1, Operación_Objetos_Media) | $\forall x$ OperacionBDMM(x) => es_un(x, ConsultaBDMM) V es_un(x, ActualizacionBDMM) V es_un(x, BusquedaEdicion) V es_un(x, Operación_Objetos_Media) (Axioma Tabla 6) |
| La CLASE1 tiene un tiempo de duración establecido | RepresentacionDatosMM (CLASE1) => tiene(CLASE1, Tiempo) | $\forall x$ RepresentacionDatosMM(x) => tiene(x, Tiempo) (Axioma Tabla 6) |
| El tiempo de duración de la CLASE1 es de 45 minutos | Tiempo(CLASE1) => es_un(CLASE1, 45Minutos) | $\forall x$ Tiempo(x) => es_un(x, Instante) V es_un(x, PeriodoTemporal) V es_un(x, IntervaloTemporal) (Axioma Tabla 6) |
| La CLASE1 muestra un video y una presentación que contiene texto e imágenes | Operación_Objetos_media (ConsultarCLASE1GC) => es_un(ConsultarCLASE1GC, ObjetoMedia_Texto) V es_un(ConsultarCLASE1GC, ObjetoMedia_Audio) V es_un(ConsultarCLASE1GC, ObjetoMedia_Video) V es_un(ConsultarCLASE1GC, ObjetoMedia_Imagen) | $\forall x$ Operación_Objetos_media (x) => es_un(x, ObjetoMedia_Texto) V es_un(x, ObjetoMedia_Audio) V es_un(x, ObjetoMedia_Video) V es_un(x, ObjetoMedia_Imagen) (Axioma Tabla 6) |
| La CLASE1 no puede tener un valor nulo | RestriccionIntegridadObjeto (CLASE1) => es_un(CLASE1, ValornoNulo) | $\forall x$ RestriccionIntegridadObjeto(x) => es_un(x, ValornoNulo) (Axioma de Tabla 3) |
| La CLASE1 puede ser un archivo de Video | $\forall x$ RestricciondeDominio (CLASE1) => es_un(CLASE1, ValorPermitido) | $\forall x$ RestricciondeDominio(x) => es_un(x, ValorPermitido) (Axioma de Tabla 3) |

Tabla 10. Ejemplo de conceptos, operaciones y restricciones de la BDMM bajo estudio usando el modelo ontológico para BDMM.

| Operaciones | LPO | Acción |
|--|---|---|
| El Video01.mpeg es un metadato video y se inserta a través de la Aplicación de Soporte Usuario QuickTime | $\forall x$ MetadatoVideo(Video01) => tiene(Video01, Contenido_de_Video) \wedge tiene(Video01, Herramientas_Sop_Usuario) \wedge tiene(Video01, Herramientas_Sop_Aplicacion) | $\forall x$ MetadatoVideo(x) => tiene(x, Contenido_de_Video) \wedge tiene(x, Herramientas_Sop_Usuario) \wedge tiene(x, Herramientas_Sop_Aplicacion) (Axioma Tabla 5) |
| Actualizar la CLASE1 con el Video01.jpg | $\forall x$ OperacionBDMM (ActualizarCLASE1) => es_un(ActualizarCLASE1, ActualizacionBDMM) | $\forall x$ OperacionBDMM(x) => es_un(x, ActualizacionBDMM) V es_un(x, Operación_Objetos_Media) (Axioma Tabla 6) |
| Insertar el Video01.jpeg es una operación de objetos media | $\forall x$ Operación_Objetos_media (InsertarVideo01.mpeg) => es_un(InsertarVideo01, ObjetoMedia_Video) | $\forall x$ Operación_Objetos_media (x) => es_un(ObjetoMedia_Texto, x) V es_un(ObjetoMedia_Audio, x) V es_un(ObjetoMedia_Video, x) V es_un(ObjetoMedia_Imagen) (Axioma Tabla 6) |
| Ejecutar método Ingresar Video01 | Solicitud (IngresarVideo01) => es_un(IngresarVideo01, InvocaciónOperación) | $\forall x$ Solicitud(x) => es_un(x, InvocaciónOperación) (Axioma Tabla 8 Operación BDOO) |

| Operaciones | LPO | Acción |
|---|---|--|
| Actualizar Video01.mpeg en el Objeto de Aprendizaje Clase1 de la materia Gestión de conocimientos | InterfazdeOperación(ActualizarVideo01) => tiene(ActualizarVideo01, InsertarVideo) \wedge tiene(ActualizarVideo01, Video01.mpeg) \wedge tiene(ActualizarVideo01, "CargadoVideo01.mpeg",) | $\forall x$ InterfazdeOperación(x) => tiene(x, NombredeMétodo) \wedge tiene(x, NombredeArgumento) \wedge tiene(x, TipodeArgumento) \wedge tiene(x, OrdendeApariciondelosArgumentos) \wedge tiene(x, ResultadosdeOperacion) (Axioma Tabla 2 Operación BDOO) |
| El archivo debe ser de tipo de video | RestriccionIntegridadReferencial(Video01) => es_un (Video01, TipoVideo) | $\forall x$ RestriccionIntegridadReferencial(x) => es_un (x, RestricciondeDominio) (Axioma Tabla 3) |
| El archivo Video01 es de tipo mpeg | $\forall x$ RestricciondeDominio(Video01) => es_un(Tipo_mpeg, ValorPermitido) | $\forall x$ RestricciondeDominio(x) => es_un(x, ValorPermitido) (Axioma Tabla 3) |

VI. VALIDACIÓN CON PROTÉGÉ

A continuación se describe la técnica de validación usada, y luego el proceso de validación de la ontología propuesta en este trabajo. Para ello se utilizó la herramienta Protégé-OWL, que es una extensión de Protégé (editor de ontologías basado en Java) que soporta el lenguaje Ontológico OWL (Ontology Web Language). OWL es un lenguaje estándar de ontologías, aprobado por el World Wide Web Consortium (W3C), el cual permite definir una semántica a través de clases, propiedades y restricciones y posee mecanismos de razonamiento.

Como cualquier otro recurso de software, el contenido de las ontologías (es decir, la definición de sus conceptos, taxonomía y axiomas) debe ser evaluado antes de ser usado o reutilizado en otras ontologías o aplicaciones. Gómez-Pérez [11] proporcionan una estructura para la evaluación ontológica que incluye la siguiente terminología:

Evaluación de la Ontología: es un juicio técnico del contenido de la ontología con respecto a un marco de referencia (un marco de referencia puede ser: especificación de requerimientos, preguntas de competencia, mundo real, etc.) durante su ciclo de vida. La evaluación de una ontología puede realizarse para los siguientes componentes de ella:

- Cada definición y axioma establecido explícitamente en la ontología.
- Definiciones importadas de otras ontologías usadas por ella.
- Definiciones que pueden ser inferidas desde otras definiciones y axiomas.

Verificación de la Ontología: se refiere a construir la ontología correctamente, esto es, asegurar que sus definiciones (una

definición escrita en lenguaje natural es una definición informal, y en lenguaje formal es una definición formal) implementan correctamente los requerimientos de la ontología.

Validación de la Ontología: se refiere a garantizar que las definiciones ontológicas modelan realmente el mundo real para el que la ontología fue creada. La meta es probar que el modelo del mundo (si existe y es conocido) es similar al mundo modelado formalmente.

Como cualquier otro recurso de software, el contenido de las ontologías (es decir, la definición de sus conceptos, taxonomía y axiomas) debe ser evaluado antes de ser usado o reutilizado en otras ontologías o aplicaciones. Gómez-Pérez (Gómez-Pérez, 2004) proporcionan una estructura para la evaluación ontológica que incluye la siguiente terminología:

Evaluación de la Ontología: es un juicio técnico del contenido de la ontología con respecto a un marco de referencia (un marco de referencia puede ser: especificación de requerimientos, preguntas de competencia, mundo real, etc.) durante su ciclo de vida. La evaluación de una ontología puede realizarse para los siguientes componentes de ella:

- Cada definición y axioma establecido explícitamente en la ontología.
- Definiciones importadas de otras ontologías usadas por ella.
- Definiciones que pueden ser inferidas desde otras definiciones y axiomas.

Verificación de la Ontología: se refiere a construir la ontología correctamente, esto es, asegurar que sus definiciones (una definición escrita en lenguaje natural es una definición informal, y en lenguaje formal es una definición formal) implementan correctamente los requerimientos de la ontología.

Validación de la Ontología: se refiere a garantizar que las definiciones ontológicas modelan realmente el mundo real para el que la ontología fue creada. La meta es probar que el modelo del mundo (si existe y es conocido) es similar al mundo modelado formalmente.

A continuación en las siguientes figuras se muestra la taxonomía de la base de datos multimedia y objeto con sus propiedades, restricciones y una instancia en OWL. La implementación en Protégé permite validar las ontologías a través de la taxonomía de las clases, las restricciones y el individuo existente.

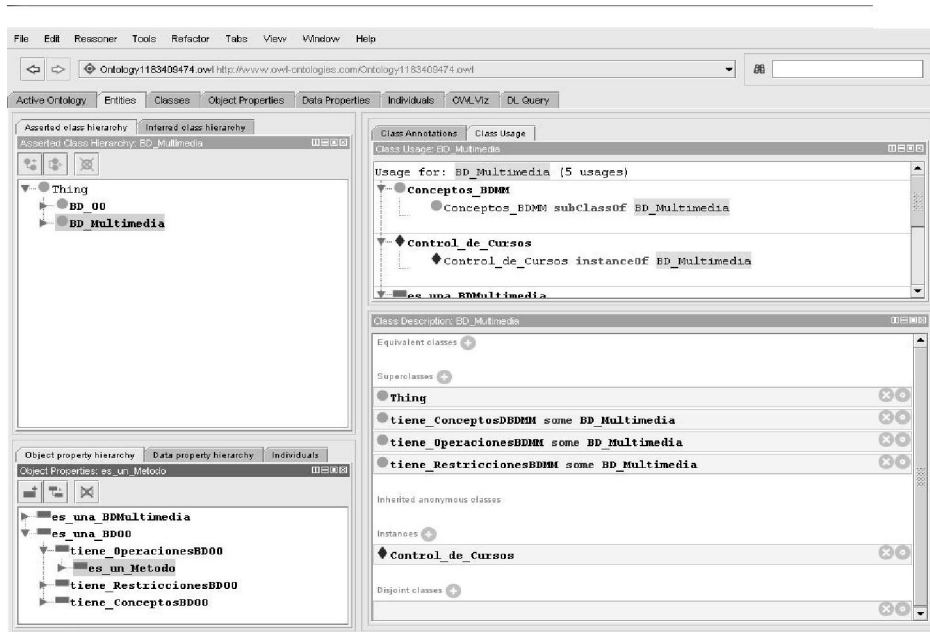


Figura 9. Ontología para Bases de Datos Orientadas a Objetos y Multimedia.

En esta figura se muestran las clases BD_OO y BD_Multimedia en el lado izquierdo superior de la figura, en la parte inferior izquierda se muestran las relaciones en tres clases o propiedades objeto en Protégé. En la parte derecha de la figura se muestran las restricciones de las bases de datos multimedia, en este caso

muestra el axioma en donde los conceptos de base de datos multimedia son una subclase de las bases de datos multimedia y en la parte inferior de la derecha se muestran todas las subclases de las bases de datos multimedia y la instancia Control de Cursos de la bases de datos multimedia.

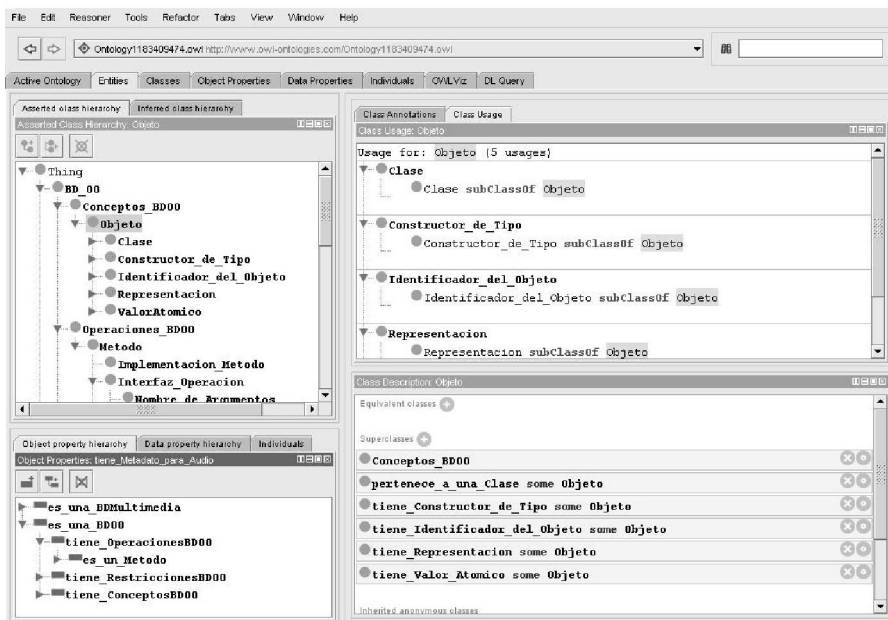


Figura 10. Ontología para Bases de Datos Orientadas a Objetos y Multimedia

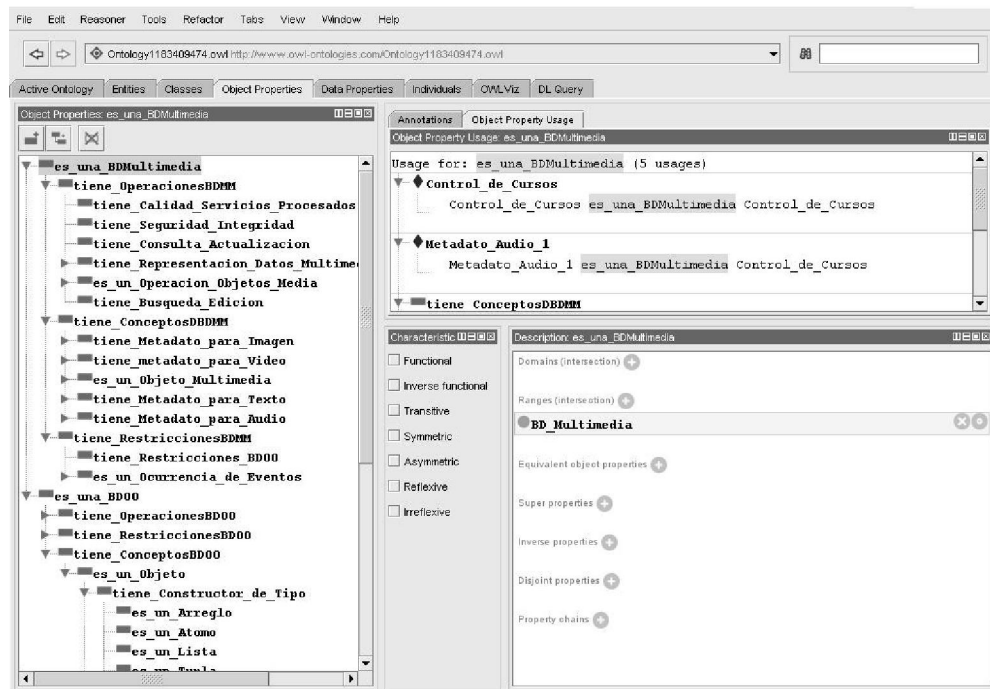


Figura 11. Propiedades Objeto para las Bases de Datos Objetos y Multimedia

En esta figura se muestran las propiedades objeto o relaciones entre las clases y subclases que conforman la ontología. Estas propiedades permiten definir los axiomas y el comportamiento de las bases de datos orientadas a objetos y multimedia a través de lógica de predicado de primer orden.

VII. CONCLUSIONES

En este trabajo se presenta un modelo ontológico para las bases de datos orientadas a objetos y multimedia. La ontología permite describir el comportamiento semántico de las bases de datos a través de sus conceptos, operaciones y restricciones. Es una ontología de dominio pues proporciona el significado de los conceptos y sus relaciones dentro del dominio de las bases de datos multimedia, y describe las actividades que ocurren en ese dominio, así como las teorías y principios elementales que dirigen las bases de datos multimedia.

Este modelo ontológico puede ser la base de un sistema Web Semántico que permite la caracterización de las bases de datos orientadas a objeto y multimedia, así como facilitar la búsqueda semántica en el área de bases de datos orientadas a objetos y multimedia. También sirve como herramienta de adquisición de conocimientos, como base para el diseño de repositorios de conocimientos, y como soporte para el desarrollo de modelos de integración para bases de datos federadas.

Al describir el modelo a través de axiomas en lógica de predicado de primer orden, se pueden diseñar mecanismos de manipulación, de razonamiento, y de aprendizaje.

Finalmente se presenta una instanciación del modelo ontológico a través de un ejemplo de objetos de aprendizaje. En el ejemplo se puede observar como se instancian los axiomas generales para los conceptos, operaciones y restricciones de las bases de datos multimedia. También se describen las propiedades o atributos de los conceptos ontológicos, lo que permitirá implementar la ontología en un lenguaje ontológico como OWL. Basado en este ejemplo, se puede desarrollar un sistema de soporte en el área de entornos virtuales de aprendizaje. Esta ontología es parte de la Tesis doctoral "Ontología para el proceso de Integración de Bases de Datos Federada" [18].

REFERENCIAS

- [1] Aberer, K. 2003; "Special issue on peer to peer data management". SIGMOD Record 32(3), pp. 29-33.
- [2] Arjen P. de Vries, Mark G.L.M. van Doorn, Henk M. Blanken, Peter M.G. Apers, The Mirror MMDBMS architecture, technical demo at VLDB 99 Edinburgh, 1999.
- [3] Batini, Ceri, Navathe 1994; Diseño Conceptual de Base de Datos: Un enfoque de Entidades-Interrelaciones; Addison Wesley/Díaz de Santos.
- [4] Becarević D., Roantree M. 2004; "A Metadata Approach to Multimedia Database federations", Information and Software Technology 46(3), pp. 195-207
- [5] Cattell R., Barry D. 2000; The Object Data Standard: ODMG 3.0; Morgan Kaufmann Publishers,
- [6] Corcho O., Fernandez-López M., Gomez-Perez A. 2003; "Methodologies, tools and languages for building ontologies. Where is their meeting point?". Data & Knowledge Engineering (46), pp. 41-64.

- [7] Date C.J. 2001; *Introducción a los Sistemas de Bases de de Datos*; Pearson Prentice Hall.
- [8] Dean M., Schreiber G., 2003; "OWL Web Ontology Language Reference". W3C Working Draft. <http://www.w3.org/TR/OWL-REF/>
- [9] Fernandez L, Gomez-Perez A., Jurista N., 1997; "METHONTOLOGY: From Ontological Arts Towards Ontological Enngineering", Symposium on Ontological Engineering of AAAI Stanford University, pp. 33-40
- [10] Fridman N., Hafner C. 1997; "The state of art in ontology design. A survey and comparative review". American Association for Artificial Intelligence AI Magazine, Fall, pp. 53-74.
- [11] Gómez-Pérez A., Fernández-López M., Corcho O., 2004; *Ontological Engineering*. Springer-Verlag.
- [12] Gruber, T. R., 1993; "A Translation Approach to Portable Ontology Specifications". *Knowledge Acquisition*, 5(2) pp. 199-220.
- [13] Grüninger M., Fox MS., 1995; "Methodology for the design and evaluation of ontologies", Skuce D (ed) *IJCAI95 Workshop on Basic Ontological issues in Knowledge Sharing*, <http://isye.gatech.edu>
- [14] Holger Knublauch, Ray W. Ferguson, Natalya F. Noy and Mark A. Musen., 2004; "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications", *Third International Semantic Web Conference*, pp 229-243.
- [15] Kosch H. and M. Döller., 2005; "Multimedia database systems: Where are we now?" *IATED DBA Konferenz*, Innsbruck, Österreich. Disponible en: www.itec.uni-klu.ac.at/~harald/MMDBoverview.pdf
- [16] Kosh Harold., 2004; *Distributed Multimedia DB Technologies Supported by MPEG-7 and MPEG-21*. CRC Press
- [17] Lassila O., Swick R., 1999; "Resource Description Framework (RDF) Model and Syntax Specification" W3C Recommendation. <http://www.w3.org/TR/REC-rdf-syntax/>
- [18] Muñoz A., 2008; *Ontología para el Proceso de Integración de Bases de Datos*; Tesis Doctoral Universidad de Los Andes.
- [19] Muñoz A., Aguilar J., 2007; "Ontological Model for Intelligent Database", *Computers Science and Technology, Proc. 11th WSEAS International Conference on Computers*, pp. 1-6.
- [20] Muñoz A, Aguilar J., Martínez R., 2006; "Integration Ontology for Distributed Database". *Advanced Software Engineering: Expanding the Frontiers of Software Technology* (Ed. S.Ochoa, G. Roman), IFIP, Springer, pp. 85-93
- [21] Muñoz A, Aguilar J., Martínez R., 2005, "Modelo Inteligente para Bases de Datos Distribuidas". *Revista Gerencia Tecnológica Informática* 4(10), pp. 91-116.
- [22] Muñoz A., Aguilar J., 2005a; "Architecture for Distributed Intelligent Databases". *IEEE, 13th Euromicro Conference on Parallel, Distributed and Network-based Processing, Euromicro-PDP 2005*, pp. 322-328.
- [23] Vincent Oria, M. Tamer Ozsu, Paul J. Iglinski; *Foundation of the DISIMA Image Query Languages*. 2003 Kluwer Academic Publishers
- [24] Prabhakaran B., 1997; *Multimedia Database Management Systems*. Kluwer Academic Publisher Boston/London/Dordrecht.
- [25] Uschold M., King M., 1995; "Towards a Methodology for Building Ontologies", Skuce D (eds) *IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing*, pp 6-10