

Desarrollo de ontologías

Ingeniería de Sistemas



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática

Definición

Una ontología es una especificación formal y explícita de una conceptualización compartida

Definición de Gruber, clarificada por Borst 1997

Marco unificador

Es el **conocimiento compartido** de un **dominio de interés** que puede utilizarse como **marco unificador** para **facilitar el intercambio de la información** en un contexto social.

Ideas clave

- La ontología es conocimiento compartido, fruto del consenso dentro de un grupo.
- Una ontología debe ser especificada usando un lenguaje formal, que pueda ser procesado por ordenadores y no sólo por personas.
- Una ontología junto con un conjunto de individuos (instancias de clases) constituye una **base de conocimiento**.

Compromisos ontológicos

Son acuerdos sobre la utilización de un vocabulario común de forma **coherente y consistente** pero no son garantía de completitud.

Ontologías compartidas

Definen el vocabulario con el que se intercambian preguntas y respuestas (recuperaciones/afirmaciones) entre agentes.

Importancia

- El análisis ontológico clarifica la estructura del conocimiento. Sin ontologías o sin las respectivas conceptualizaciones que soportan el conocimiento no puede existir un vocabulario para representar el conocimiento.
- Las ontologías permiten compartir conocimiento. Compartir ontologías puede de este modo significar la base de la definición de un lenguaje de representación del conocimiento.

Utilidad de las ontologías

- Sirven para entender como diferentes sistemas comparten información.
- Se utilizan para descubrir distorsiones que puedan presentarse en los procesos cognitivos de aprendizaje en un mismo contexto.
- Sirven para formar patrones para el desarrollo de Sistemas de Información. En el ámbito del software se viene utilizando para describir las propiedades del software (componentes, arquitecturas, lenguajes de definición).
- Permiten compartir y reutilizar conocimiento común.

Utilidad de las ontologías

- Ayudan a establecer comunicación entre personas y organizaciones con el fin de unificar diferentes áreas de investigación.
- Permiten la interoperabilidad entre sistemas de software usando ontologías como un lenguaje intermedio para unificar diferentes lenguajes y herramientas.
- Aumentan los beneficios de la Ingeniería de Sistemas ya que el uso de ontologías facilita la construcción de software clásico o basado en el conocimiento porque permite que los sistemas se puedan reutilizar.

Formas de aplicación

- Autor Neutral (Neutral Authoring): una ontología es desarrollada como un simple lenguaje y es trasladada a diferentes formatos y utilizada en aplicaciones de múltiples usos.
- Ontología como una especificación: se crea una ontología de un dominio dado, la cual tiene la capacidad de proporcionar un vocabulario para la especificación de requisitos de una o más aplicaciones. En este caso, una ontología puede ser vista como un modelo del dominio. La ontología es utilizada como una base para la especificación y desarrollo de aplicaciones en el dominio seleccionado, permitiendo así la reutilización de conocimiento.

Formas de aplicación

- Para acceder a información: una ontología es utilizada para permitir a una aplicación de múltiples usos (o a las personas) tener acceso a diversas fuentes heterogéneas de información, la cual normalmente es expresada usando un vocabulario diverso o formatos inaccesibles.
- Búsquedas basadas en ontologías: una ontología es utilizada en la búsqueda de recursos en un repositorio de información, mejorando la precisión y reduciendo la cantidad de tiempo invertido en realizar la búsqueda.

Características deseables

Claridad	Una ontología debe comunicar eficazmente el significado de los términos. Las definiciones deben ser objetivas y estar expresadas en lenguaje natural.
Coherencia	Una ontología es incoherente si una frase que se puede deducir a partir de axiomas se contradice con una definición o ejemplo dado.
Extensibilidad	Se debería poder definir nuevas expresiones a partir del vocabulario existente en las ontologías.

Componentes

Clases	representan los conceptos de un dominio.
Instancias	son objetos de una clase en particular.
Relaciones	se emplean para representar la conexión o correspondencia entre los conceptos de un dominio. Por ejemplo: pertenece-a, relativo-a, etc.
Funciones	identifican un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo: establecer-fecha, clasificar-elemento, etc.
Axiomas	son las restricciones aplicables a ciertos elementos de la ontología, necesarias para especificar correctamente el dominio de conocimiento. Por ejemplo: Si A y B son de la clase C, entonces A no es subclase de B.

Ontología ligera

- Si una ontología sólo posee conceptos y relaciones se llama taxonomía u ontología ligera.
- No siempre aparecen todos los elementos anteriores en una ontología.

Áreas de aplicación

- Ingeniería del conocimiento
- Representación de conocimiento
- Procesamiento del lenguaje natural
- Sistemas de información cooperativos
- Integración inteligente de información
- Recuperación de Información
- Gestión del conocimiento
- Comercio electrónico
- Ingeniería del software

Ingeniería del software

- Las ontologías desempeñan un papel en el desarrollo de sistemas relacionado con el modelado semántico, metadatos, análisis y diseño de patrones y creación de librerías de software reutilizable.

Ingeniería de sistemas

- Reutilización y compartición de representaciones formales en un sistema software.
- Búsqueda de información: la ontología se utiliza como meta-datos que sirven como índices en almacenes de información.
- Realización de chequeos de consistencia para asegurar la fiabilidad.
- Especificación de requisitos de un sistema de información.
- Adquisición de conocimiento.

Áreas relacionadas

- Web Semántica: es una Web extendida, dotada de mayor significado en la cual los usuarios de Internet pueden encontrar respuestas a sus preguntas de manera más rápida y sencilla gracias a una información bien definida.
- Ingeniería de Dominio (Domain Engineering): un dominio es un área funcional diferenciable con requisitos y rasgos similares (features) que puede ser soportada por algún tipo de sistema de software. La Ingeniería de Dominios se encarga de capturar, organizar y representar la información útil para el desarrollo de sistemas de software, de manera que esta pueda ser reutilizada para crear nuevos sistemas.
- Software Engineering Environments (SEEs): representan colecciones integradas de herramientas que facilitan las actividades de la ingeniería de software a través de todo su ciclo de vida.
- Gestión del Conocimiento (Knowledge Management): es un concepto utilizado para la transmisión de conocimiento y experiencia entre los empleados dentro de una organización.
- Inteligencia Artificial (IA): una de las principales áreas de estudio de la Inteligencia Artificial está representada por los Agentes Inteligentes, entidades de software con una arquitectura robusta que pueden funcionar en distintos entornos o plataformas de computación y que son capaces de realizar de forma “inteligente” y autónoma distintos objetivos, intercambiando información con el entorno, o con otros agentes humanos o computacionales.

Tipos de ontologías

según el alcance de su aplicación

- Ontologías de Dominio: Proporcionan el vocabulario necesario para describir un dominio dado.
- Ontologías de Tareas: Proporcionan el vocabulario para describir términos involucrados en los procesos de resolución de problemas los cuales pueden estar relacionados con tareas similares en el mismo dominio o en dominios distintos. Incluye nombres, verbos, frases y adjetivos relacionados con la tarea (objetivo, planificación, asignar, clasificar).
- Ontologías Generales: Representa los datos generales que no son específicos de un dominio. Por ejemplo, ontologías sobre el tiempo, el espacio, ontologías de conducta, de capacidad, etc.

Tipos de ontologías

según cantidad y tipo de conceptualización

- Terminológicas: Especifican los términos que son usados para representar conocimiento en el universo de discurso. Son utilizadas para unificar vocabulario en un dominio determinado.
- De información: Especifican la estructura de almacenamiento de bases de datos. Ofrece un marco para el almacenamiento estandarizado de información (estructura de registros de la BD).
- De modelado del conocimiento: Especifican conceptualizaciones del conocimiento. Poseen una rica estructura interna y suelen estar ajustadas al uso particular del conocimiento que describen (términos y semántica).

Ejemplos de ontologías

- SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications). Orienta a los desarrolladores que no tienen experiencia en la representación del conocimiento a construir rápidamente aplicaciones basadas en ontologías.
- CONON (CONtext ONTology). Proporciona una ontología orientada a aplicaciones sensibles al contexto.
- FIPA (Foundation for Intelligent Physical Agents) Device Ontology. Expresa las capacidades de diferentes dispositivos en sistemas de computación ubicuas
- GUMO (General User Model Ontology). Útil para el modelado de conceptos generales.
- GAS (Gadgetware Architectural Style). Expresa las relaciones básicas entre diferentes dominios.
- Upper-Level Ontology . Representa conceptos habitualmente empleados en la computación ubicua.

Leguajes de ontologías

- SHOE (Simple HTML Ontology Extensions). Es el lenguaje de ontologías más primitivo y menos funcional.
- OIL (Ontology Inference Layer). Es la evolución de SHOE, y utiliza ya la sintaxis de XML (Extensible Markup Language) y está definido como una extensión de RDFS (Resource Description Framework Schema).
- DAML+OIL. Este lenguaje surge de la colaboración entre dos organismos y unifica los lenguajes DAML (DARPA's Agent Markup Language) y OIL. Se basa ya en estándares W3C. Incorpora más expresividad que el anterior.
- OWL (Web Ontology Language). Es una extensión de RDF, e incorpora etiquetas que permite definir el significado de conceptos y las relaciones entre ellos, dando una expresividad altamente potencial.
- KIF (Knowledge Interchange Format). Es un lenguaje basado en lógica de primer orden.

Metodologías

- Ciclo de vida: es el tiempo transcurrido durante la concepción, elaboración y mantenimiento de la ontología.
- Herramientas de apoyo: las metodologías deben ser apoyadas por herramientas ontológicas que faciliten la creación de la ontología.
- Nivel de modelado: indica cómo se realiza la extracción de los datos.
- Nivel de abstracción: indica el nivel de relación existente entre la información concreta y la forma en que se extrae dicha información. Cuanto mayor sea el nivel de abstracción, más fácil será para las aplicaciones extraer los datos.
- Uso de ontologías base: normalmente las ontologías reutilizan términos o características de otras ya creadas.

Methontology

- Propone modelar una ontología usando un conjunto de representaciones intermedias tabulares que permiten modelar los principales componentes de una ontología.
- La creación de la ontología puede empezar desde cero o en base a la reutilización de otras existentes.
- Desarrollada por el grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid.
- Recomendada por la FIPA (Foundation for Intelligent Physical Agents), organismo que promueve las tecnologías basadas en agentes físicos inteligentes y la interoperabilidad de sus estándares con otras tecnologías.

On-to-Knowledge

- Su objetivo es aplicar ontologías a la información disponible electrónicamente para mejorar la calidad de la gestión de conocimiento en organizaciones grandes y distribuidas, frecuentemente relacionadas con el World Wide Web.
- Está basada en el estudio de casos de uso, y contiene la identificación de los objetivos alcanzados por las herramientas de gestión del conocimiento.
- Desarrollada por la Universidad de Karlsruhe.

Otras metodologías

- Metodología Cyc
- Metodología de Uschold y King
- Metodología de Grüninger y Fox
- Metodología Kactus

Protégé

- Herramienta para el desarrollo de ontologías y sistemas basados en el conocimiento.
- Creada por la Universidad de Standford.
- Es software libre y código abierto.
- Desarrollada en JAVA.
- Funciona bajo LINUX y WINDOWS.
- Está disponible en:
<http://protege.stanford.edu/>

Desarrollo con Protégé

- El usuario puede construir ontologías de dominio, generar usuarios de entrada de datos y efectuar la propia entrada de datos.
- Es una herramienta que permite acceso a aplicaciones externas basadas en conocimiento.
- Además es una biblioteca a la que otras aplicaciones pueden acceder, permitiéndoles acceder a las bases de conocimiento de las cuales se dispone.

Diseño de ontología

- Implica la descripción explícita y formal de:
 - conceptos en un dominio de discurso (clases),
 - propiedades de conceptos (slots/roles o propiedades), y
 - restricciones sobre las propiedades (facetas/restricciones de rol).

Terminología

- Clase (Concepto): Descripción formal de una entidad del universo o dominio que se quiere representar. Constituye la pieza básica de estructuración del conocimiento. La decisión sobre qué considerar una clase del dominio no es fácil. Tal decisión debe ser tomada de acuerdo a los objetivos de la ontología (extraídos de las sesiones con los expertos, preguntas relevantes y de estándares existentes en el dominio). Una clase puede tener subclases que representan conceptos que son más específicos que dicha clase.
- Clase abstracta: Clase que no permite que existan instancias de ella. Se usa para agrupar conceptos, introducir cierto orden en la jerarquía, pero suelen ser demasiado generales para admitir instancias.

Terminología

- Instancia: Representan objetos concretos del dominio, pertenecientes a una clase. La colección de instancias constituye la base de hechos (también denominada base de datos o base de conocimiento) del modelo.
- Instancia indirecta: Cuando una clase es instancia indirecta de otra, quiere decir que es instancia de alguna de sus clases derivadas. En contraposición a instancia directa, donde no existen clases intermedias.
- Propiedad (Atributo, Slot): Característica que permite describir más detalladamente la clase y sus instancias. Establece que la clase o concepto posee una propiedad que se concretará mediante un valor. Los valores de las propiedades o atributos pueden ser tipos básicos como cadenas de caracteres o números, pero también pueden ser otras clases o instancias.
- Faceta (Restricción sobre las propiedades): Es alguna propiedad de la propiedad. Por ejemplo, la cardinalidad, si la propiedad es obligatoria o no, etc.

Terminología

- Relación: Interacción o enlace entre los conceptos o clases del dominio. Algunas relaciones semánticas básicas son: subclase de, parte de, parte exhaustiva de, conectado a, es un, etc. Suelen configurar la taxonomía del dominio. Las relaciones más simples se modelan mediante una propiedad de una clase cuyo valor es una instancia de otro concepto.
- Axioma: Regla que se añade a la ontología y que permite describir el comportamiento de los conceptos o clases. Se establecen a partir de valores específicos de las propiedades. A través de los axiomas es posible inferir conocimiento no codificado explícitamente en la ontología.

Terminología

- **Anotación:** Es el proceso de relleno de instancias a partir de texto libre. Existen dos maneras de anotar texto: la más usual es la inclusión de etiquetas semánticas dentro del texto que se está procesando. Esto implica que el formato del texto debe ser editable y procesable. En el caso de no disponer de este formato, la segunda manera consiste en rellenar las instancias directamente en el modelo, dejando el texto original sin modificar.
- **Herencia:** Propiedad de la relación 'es_un' que permite que las clases relacionadas (heredadas) cuenten con los atributos de la clase con la cual se relacionan (clase padre).

Terminología

- Herencia múltiple: Se da cuando una clase dada hereda o cuenta con las propiedades de dos clases padre con las que establece dos relaciones del tipo 'es_un'.
- Derivación: Organización de las clases de la ontología en un árbol de jerarquía mediante sucesivas relaciones 'es_un' (también llamadas kind_of, is_a, o herencias) con la propiedad de herencia. Esta organización permite el encadenamiento sucesivo de herencias desde las clases de nivel superior a las clases situadas en niveles inferiores, llamadas clases derivadas.

Reglas básicas

- No hay forma única de modelar un dominio. La mejor solución casi siempre depende de la aplicación a desarrollar.
- El desarrollo de ontologías es un proceso necesariamente iterativo.
- Los conceptos en la ontología deben ser cercanos a los objetos (físicos o lógicos) y relaciones en el dominio de interés. Esos son muy probablemente sustantivos (objetos) o verbos (relaciones) en oraciones que describen tu dominio.

Modelo de proceso

- Debe ser un proceso iterativo, compuesto de aproximaciones sucesivas hasta llegar a una versión final de la misma.
- De este modo, inicialmente dispondremos de un primer borrador que debe ser revisado y refinado, debiendo ocuparnos por completar todos sus detalles.

Metodología

- Se basa en el empleo de las Competency Questions que son las preguntas que el sistema debería ser capaz de contestar y que servirá como prueba de control de calidad).

Proceso de diseño

- Determinar el dominio y alcance de la ontología además del objetivo de la misma.
- Considerar la reutilización de ontologías existentes.
- Enumerar términos importantes de la ontología.
- Definir las clases y la jerarquía de clases.
- Definir las propiedades/slots de las clases.
- Definir las facetas o restricciones de las propiedades o slots.
- Crear instancias.

Determinar el dominio y alcance

- Mediante las Competency Questions, deberíamos responder las siguientes cuestiones básicas:
 - ¿Qué dominio cubrirá la ontología?
 - ¿Para qué se va a emplear la ontología?
 - ¿Qué preguntas debería contestar la ontología?
 - ¿Quién usará y mantendrá la ontología?

Preguntas de competencia

- *¿Qué características debo considerar cuando elijo un vino?*
- *¿Bordeaux es un vino rojo o blanco?*
- *¿El Cabernet Sauvignon va bien con comida de mar?*
- *¿Cuál es la mejor elección de vino para acompañar carne asada?*
- *¿Qué características de un vino afectan su idoneidad con un pescado?*
- *¿El cuerpo o aroma de un vino específico cambia con su año de cosecha?*
- *¿Cuáles fueron buenas cosechas para el Napa Zinfandel?*

(Ejemplo Ontología de vinos - Natalya F., 2005)

Considerar la reutilización de ontologías existentes

- Conviene comprobar si podemos partir de una o algunas ontologías existentes, refinando o extendiendo sus límites.
- Si nuestro sistema necesita interactuar con otras aplicaciones que emplean determinadas ontologías, la reutilización de tales ontologías es un requisito necesario en nuestro proyecto.

Enumerar términos importantes

- Listar todos los términos que tienen relación con nuestro dominio. Estos términos pueden seleccionarse a partir de las fuentes documentales (artículos, libros, sitios web, etc.) de donde se obtuvieron algunas de las Competency Questions, junto con la ayuda de los expertos en el dominio.
- Al principio no debemos preocuparnos por las posibles relaciones entre ellos, si varios se enmarcan en un mismo concepto, o si corresponden a conceptos o propiedades.
- Algunos de estos términos se transformarán en clases, otros en propiedades y otros en instancias.

Definir las clases y la jerarquía

- Existen diversas aproximaciones a la hora de definir la jerarquía de clases:
 - Top-down
 - Botom-up
 - Combinación de las dos.

Reglas para las clases

- Una nueva subclase de una clase generalmente (1) debería tener nuevas propiedades que no posee la clase; o (2) debería tener diferentes valores para las propiedades que los de la clase; o (3) debería participar en diferentes relaciones que la clase.
- En jerarquías terminológicas, las nuevas clases no tienen por qué introducir nuevas propiedades.
- Si un factor es importante en el dominio y pensamos en los objetos con diferentes valores para ese factor como diferentes clases de objetos, entonces deberíamos crear una nueva clase o clases considerando dicho factor.
- Las instancias son los conceptos más específicos representados en una ontología.
- Si los conceptos de un dominio forman una jerarquía natural, podríamos representarlos como clases, aunque sean clases abstractas.

Reglas para la jerarquía de clases

- Si una clase A es una superclase de la clase B, entonces toda instancia de B es también una instancia de A.
- Una subclase de una clase representa un concepto que es un tipo especial o una subespecie dentro del concepto representado por la clase.
- Si B es una subclase de A y C es una subclase de B, entonces C es una subclase de A.
- No debemos emplear sinónimos de un mismo concepto para representar clases diferentes, sino que los sinónimos deben considerarse denominaciones diferentes para un mismo y único concepto.

Reglas para la jerarquía de clases

- No deben aparecer ciclos o bucles en la jerarquía de clases.
- Los conceptos de un mismo nivel de la jerarquía o clases hermanas (excepto los que derivan directamente de la raíz) deben presentar el mismo nivel de generalidad.
- Cada clase debería tener entre 2 y 12 subclases directas.
- La ontología no debería contener toda la información posible sobre el dominio: no es necesario especializar o generalizar más de lo que necesitemos para la aplicación.

Definir las facetas de los slots

- Las propiedades (slots) pueden poseer diferentes facetas (restricciones) que describan o caractericen el tipo de valores que posee una cierta propiedad, los valores permitidos, el número de valores posible (cardinalidad), entre otras.

Definir las propiedades (slots)

- Debemos determinar qué clase describe cada propiedad de la lista de términos.
- Las características de los objetos que pueden convertirse en propiedades o slots en una ontología son las siguientes:
 - Características intrínsecas.
 - Características extrínsecas.
 - Partes del objeto, si el concepto o la clase está estructurado, tanto físicamente como de manera abstracta o convencional.
 - Relaciones con otros conceptos o clases.

Reglas para las propiedades

- Todas las subclases de una clase heredan las propiedades de dicha clase, por lo que una propiedad debería ser adscrita a la clase más general que posea dicha propiedad.
- Garantizar la consistencia de la base de conocimiento cuando existan propiedades o relaciones inversas en una ontología.

Facetas más comunes

- Cardinalidad. Establece cuántos valores puede tener una propiedad o slot.
- Tipo de valor. Describe qué tipo de valores puede poseer una propiedad.
- Dominio y Rango de una propiedad o slot. Se suele denominar Rango de una propiedad a las clases permitidas para una propiedad de tipo Instancia. Se suele denominar Dominio de una propiedad al conjunto de clases que describe o caracteriza dicha propiedad.

Cardinalidad

- Algunos sistemas distinguen únicamente entre cardinalidad simple (como máximo un valor) y cardinalidad múltiple (se permiten cualquier número de valores).

Tipo de valor

- String [Cadena]. Es el tipo de valor más simple, empleado en propiedades como 'nombre'. Indica que su valor es un conjunto de caracteres.
- Número. Describe propiedades con valores numéricos.
- Boolean. Son propiedades cuyos valores son SI o NO.

Tipo de valor

- Enumerado. Las propiedades enumeradas especifican una lista de los valores permitidos para el slot.
- Instancia. Las propiedades tipo Instancia permiten definir relaciones entre clases. Los slots con tipo de valor Instancia deben especificar la lista de clases permitidas de las cuales pueden proceder las instancias que componen la relación.

Dominio y rango de un slot

- Se denomina Rango de una propiedad a las clases permitidas para una propiedad de tipo Instancia.
- Se denomina Dominio de una propiedad al conjunto de clases que describe o caracteriza dicha propiedad.

Reglas para el dominio y el rango

- Cuando se definan el Dominio o el Rango de una propiedad, emplee la clase o clases más generales que puedan ser respectivamente Dominio o Rango de dicha propiedad.
- No defina nunca un Dominio o un Rango que sea excesivamente general: todas las clases del dominio de una propiedad deben ser caracterizadas o descritas por dicha propiedad, mientras que todas las instancias de todas las clases del rango de una propiedad son valores permitidos para dicha propiedad de tipo Instancia.

Crear instancias

- La decisión sobre qué considerar clases y cuándo introducir las instancias es una decisión sobre el nivel de granularidad o grado de especificidad de la representación que deseemos introducir en la ontología.
- La solución a esta decisión viene impuesta por la Competency Questions, pues los conceptos más específicos que constituyan las respuestas a estas preguntas relevantes constituyen las instancias o individuos de nuestro dominio.