Clasificación y ejemplos del uso de ontologías en Ingeniería del Software

Gabriela N. Aranda
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
Buenos Aires 1400, Neuquén, Argentina
garanda@uncoma.edu.ar

Francisco Ruiz
Departamento de Informática
Universidad de Castilla-La Mancha
Paseo de la Universidad 4, Ciudad Real, España
francisco.ruizg@uclm.es

Resumen. *Las ontologías se han convertido en herramientas que pueden asistir eficientemente en las actividades de desarrollo y mantenimiento de software ya que, al reducir la ambigüedad y proveer un marco de unificación, ayudan a compartir conocimiento, facilitan la comunicación y permiten una alta reutilización de conocimiento. En este trabajo se presentan varias clasificaciones de ontologías propuestas en la literatura y ejemplos de uso en ingeniería del software, que son analizados de acuerdo a ellas.*

Palabras clave. Ontologías. Ingeniería del Software.*

1. Introducción

En la literatura existen muchas definiciones de ontología. La más ampliamente citada es la dada por Gruber en 1993: *"una especificación formal, explícita de una conceptualización compartida"*, donde conceptualización se entiende como una visión abstracta y simple del mundo que se desea representar, una representación del conocimiento basada en objetos, conceptos y entidades que "existen" en el área de estudio, y las relaciones que existen entre ellas. Que sea explícita significa que los conceptos utilizados y las restricciones para usarlos están explícitamente definidos. Es formal si puede ser comprensible por una máquina. Por último, es compartida, si el conocimiento que captura tiene el consenso de la comunidad. [7]

Dentro del campo de análisis de la Ingeniería del Software, una ontología puede verse como un vocabulario de representación para un dominio específico, que representa elementos conceptuales y relaciones entre ellos; sin embargo la ontología no es el vocabulario en sí mismo, sino lo que él representa ya que, por ejemplo, si se traduce cada vocablo a otro idioma no significa que cambie la ontología. [2]

Algunos autores postulan la necesidad de que una ontología sea una "teoría", es decir, un vocabulario formal con un conjunto de axiomas definido para ese vocabulario, pues los axiomas expresan nuevas relaciones entre conceptos y restringen las posibles interpretaciones [12] [16].

En cualquier caso, es evidente que disponer de una ontología del dominio de aplicación de un sistema software, o de los procesos para su diseño y construcción, es una ayuda importante para evitar errores y problemas en todas las fases del ciclo de vida del producto software: desde el análisis de requisitos inicial (facilitando la interacción analista-cliente) hasta la etapa de mantenimiento (más fácil comprensión de las peticiones de modificación, mejor comprensión del sistema mantenido, etc.).

En la sección 2 se presentan varias maneras de clasificar ontologías propuestas en la literatura, de la

* *Workshop en Ingeniería del Software y Bases de Datos WISBD 2005*

misma manera que en la sección 3 se presentan varias propuestas de uso de ontologías en Ingeniería del Software. En la sección 4 se describen algunos ejemplos de ontologías aplicadas para mejorar los procesos de desarrollo y mantenimiento de software, continuando, en la sección 5, con una comparativa de las ontologías presentadas, de acuerdo a las clasificaciones de las secciones anteriores. Por último, la sección 6 incluye las principales conclusiones obtenidas en este estudio.

2. Clases de Ontologías

En la literatura se encuentran diversas clasificaciones de ontologías de acuerdo a distintos enfoques:

De acuerdo al nivel de generalidad

Clasificación según Guarino [10]:

- ***Ontologías de Alto Nivel***: Describen conceptos generales como espacio, tiempo, materia, objeto. Son independientes de un dominio o problema particular. Su intención es unificar criterios entre grandes comunidades de usuarios.
- ***Ontologías de Dominio***: Describen el vocabulario relacionado a un dominio genérico (por ejemplo medicina o automotores), por medio de la especialización de los conceptos introducidos en las ontologías de alto nivel.
- ***Ontologías de Tareas***: Describen el vocabulario relacionado a una tarea o actividad genérica (por ejemplo de diagnóstico o de ventas), por medio de la especialización de los conceptos introducidos en las ontologías de alto nivel.
- ***Ontologías de Aplicación***: Describen conceptos que pertenecen a la vez a un dominio y a una tarea particular, por medio de la especialización de los conceptos de las ontologías de dominio y de tareas. Generalmente corresponden a *roles* que juegan las entidades del dominio cuando ejecutan una actividad.

Clasificación según Fensel [5]:

- ***Ontologías Genéricas o de Sentido Común***: Capturan conocimiento general acerca del mundo. Proveen nociones básicas y conceptos para cosas tales como espacio, tiempo, estado, eventos. Son válidas en varios dominios.
- ***Ontologías Representacionales***: No se comprometen con ningún dominio en particular. Proveen entidades sin establecer que deberían representar, por lo tanto definen conceptos para expresar conocimiento de manera orientada a objetos o a marcos de trabajo.
- ***Ontologías de Dominio***: Capturan el conocimiento válido para un tipo particular de dominio (ej: electrónica, medicina, mecánica).
- ***Ontologías de Métodos y Ontologías de Tareas***: Las primeras proveen términos específicos para métodos particulares de resolución de problemas, mientras que las segundas proveen términos para tareas específicas. Ambas proveen un punto de vista de razonamiento sobre conocimiento del dominio.

Las clasificaciones dadas por ambos autores pueden alinearse según se muestra en la figura 1. Podemos destacar que el nivel de *Ontologías de Tareas* en la clasificación de Guarino y el de *Ontologías Representacionales* en la clasificación de Fensel no encuentran contraparte en la otra clasificación. Aún cuando ambas dicen representar tareas en forma genérica, las primeras las expresan para un dominio específico, mientras que las segundas se menciona claramente que son independientes del dominio.

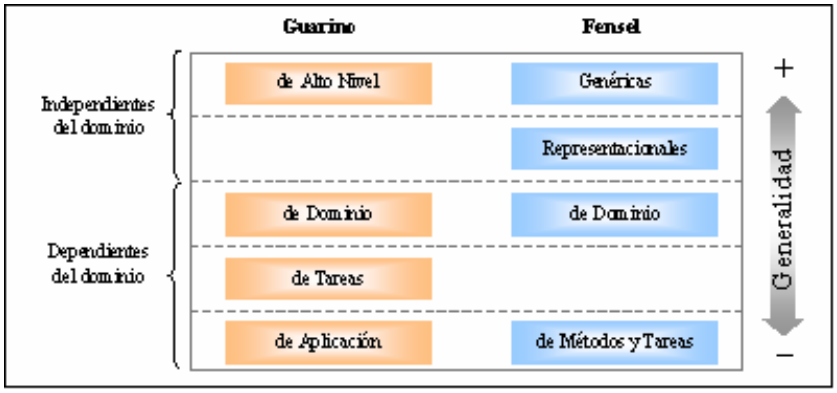

Figura 1: Comparación de clasificaciones según nivel de generalidad

De acuerdo al tipo de estructura de conceptualización [15]

- ***Ontologías terminológicas***: Especifican términos a utilizarse para representar el conocimiento en el dominio de estudio. Intentan obtener un lenguaje unificado sobre un tema específico, por ejemplo, el Sistema de Lenguaje Médico Unificado (ULMS)
- ***Ontologías de información***: Especifican la estructura de los registros de una base de datos, determinando un marco para el almacenamiento estandarizado de información. Un ejemplo es un marco de trabajo para modelar los registros médicos de pacientes.
- ***Ontologías de representación de conocimiento***: Especifican conceptualizaciones del conocimiento. Comparados con las ontologías de información, estas tienen una estructura interna más rica. Suelen estar enfocadas a un uso particular del conocimiento que describen.

De acuerdo a los aspectos del mundo real que intentan modelar [11]

- ***Ontologías Estáticas***: Describen las cosas que existen, sus atributos y las relaciones entre ellos. Esta clasificación asume que el mundo está poblado de entidades que están dotadas de una identidad única e inmutable.
  Términos que utilizan: entidades, atributos, relaciones
- ***Ontologías Dinámicas***: Describen los aspectos que pueden cambiar en el mundo que modelan. Para modelarlas se pueden utilizar máquinas de estados finitos, redes de Petri, etc.
  Términos que utilizan: procesos, estados, transición de estados.
- ***Ontologías Intencionales***: Describen aspectos que tienen que se refieren al mundo de las motivaciones, intenciones, metas, creencias, alternativas y elecciones de los agentes involucrados.
  Términos que utilizan: aspecto, objetivo, soporte, agente
- ***Ontologías Sociales***: Describen aspectos que se relacionan con lo social, estructuras organizacionales, redes, interdependencias.
  Términos que utilizan: actor, posición, rol, autoridad, compromiso.

De acuerdo a la riqueza de la estructura interna y del sujeto de conceptualización [7]

Esta es una clasificación más extensa que las anteriores, basada en dos enfoques distintos: (1) la riqueza de la estructura interna, basada en una clasificación propuesta por Lassila y McGuinness; (2) el sujeto de conceptualización, una extensión de la clasificación propuesta en [15] por van Heist et al y presentada anteriormente

Los tipos de ontologías, de acuerdo a la **riqueza de su estructura interna** son:

- ***Vocabularios controlados***: lista finita de términos, por ejemplo un catálogo.
- ***Glosarios***: son listas de términos con sus significados expresados en lenguaje natural.
- ***Tesauros***: proveen semánticas adicionales entre términos, como por ejemplo información referida a sinónimos.
- ***Jerarquías Informales "Es-Un"***: Son jerarquías de términos que no corresponden a una subclase estricta, por ejemplo los términos "auto de alquiler" y "hotel" podrían ser modelados informalmente bajo la jerarquía "viaje" ya que se considerarían partes clave de un viaje.
- ***Jerarquías Formales "Es-Un"***: En este caso existe una relación estricta entre instancias de una clase y de las superclases correspondientes. Su objetivo es explotar el concepto de herencia.
- ***Marcos***: Son ontologías que incluyen tanto clases como sus propiedades, las cuales pueden ser heredadas por otras clases en los niveles mas bajos de una taxonomía formal "es-un".
- ***Ontologías que expresan restricciones de valor***: por ejemplo expresan restricciones de acuerdo al tipo de dato de una propiedad (por ejemplo, tipo fecha).
- ***Ontologías que expresan restricciones lógicas generales***: Son las ontologías más expresivas. Especifican, por medio de lógica de primer orden, restricciones entre los términos de la ontología.

De acuerdo al **sujeto de conceptualización**, los tipos de ontologías son:

- ***Ontologías de representación de conocimiento***: capturan primitivas de representación utilizadas para formalizar conocimiento bajo un paradigma de representación de conocimiento dado.
- ***Ontologías comunes o generales***: representan conocimiento de sentido común y reutilizable en distintos dominios, por ejemplo sobre vocabulario relacionado a cosas, eventos, tiempo, espacio, etc.
- ***Ontologías de alto nivel***: son ontologías que describen conceptos y nociones generales bajo las cuales pueden enlazarse los términos raíces de todas las ontologías. Un problema que existe es que hay varias de estas ontologías de alto nivel que difieren en los criterios para clasificar la mayoría de los conceptos generales de la taxonomía.
- ***Ontologías de dominio***: son aquellas ontologías reutilizables en un dominio particular (medicina, ingeniería, etc.). Proveen vocabularios sobre conceptos dentro del dominio y sus relaciones.
- ***Ontologías de tareas***: describen vocabulario relacionado a actividades genéricas. Proveen un vocabulario sistemático de términos utilizados para resolver problemas que pueden o no pertenecer a un mismo dominio.
- ***Ontologías de tareas de dominios***: a diferencia de las ontologías de tareas, estas ontologías son reutilizables en un dominio dado, y no entre dominios diferentes.
- ***Ontologías de métodos***: proveen definiciones de conceptos relevantes y relaciones aplicables a un proceso de razonamiento específico a fin de cumplir una tarea particular.
- ***Ontologías de aplicaciones***: son dependientes de las aplicaciones. A menudo extienden y especializan vocabulario de una ontología de dominio o de tareas para una aplicación particular.

3. Usos en Ingeniería de Software

Entre los principales aspectos por los cuales es importante el uso de ontologías en cualquier campo de actividad humana, se pueden destacar los siguientes:

- **Clarifican la estructura de conocimiento:** Durante el análisis ontológico se definen los conceptos

del dominio así como las relaciones entre ellos, de manera que este paso, ejecutado en forma adecuada, permite una clara especificación de la naturaleza de los conceptos y de los términos utilizados para representarlos, respecto del cuerpo de conocimiento que se quiere construir. [1]

- **Reducen la ambigüedad conceptual y terminológica:** El análisis ontológico provee un marco de unificación aún entre personas con necesidades y/o puntos de vista que dependen de sus contextos particulares. [14]
- **Permiten compartir conocimiento:** Mediante un apropiado análisis ontológico se consigue un conjunto de conceptualizaciones de un dominio específico, y un conjunto de términos que las soportan. Mediante una sintaxis adecuada, esas conceptualizaciones y las relaciones entre ellas, serán expresadas y codificadas en una ontología, la cual podrá ser compartida con cualquiera que tenga necesidades similares en el mismo dominio. [1]

Los usos posibles de las ontologías en Ingeniería del Software han sido analizados por distintos autores y se han determinado las siguientes utilidades:

Según Uschold-Gruninger [14]

- ***Comunicación*:** Las ontologías reducen la ambigüedad conceptual y terminológica puesto que proveen un marco de unificación. Por ello permiten compartir el conocimiento y facilitan la comunicación entre personas aún con distintas necesidades y/o puntos de vista de acuerdo a su contexto en particular.

  En una organización hay aspectos implícitos que pueden explicitarse por medio de ontologías, por ejemplo los modelos normativos y las redes de relaciones entre personas; mientras que por otro lado permiten conseguir consistencia, falta de ambigüedad e integración entre distintas perspectivas de los usuarios.
- ***Interoperabilidad*:** Cuando usuarios diferentes necesitan intercambiar datos o bien cuando un usuario utiliza diferentes herramientas de software el concepto de interoperabilidad no es menor.

  Desde un primer punto de vista las ontologías pueden actuar como "Inter-lengua", es decir, que pueden utilizarse para soportar la traducción entre diferentes lenguajes y representaciones; puesto que es más eficiente tener un traductor por cada parte involucrada con una ontología "de intercambio" que diseñar un traductor para cada par de partes (lenguajes o representaciones) involucradas.
- ***Ingeniería de sistemas*:** La aplicación de ontologías para soportar el diseño y desarrollo de sistemas de software puede darse con varios propósitos:

  Especificación: El rol que las ontologías juegan en la especificación depende del grado de formalidad y automatización dentro de la metodología de diseño del sistema. Desde un enfoque informal, las ontologías facilitan el proceso de identificación de requerimientos y la comprensión de las relaciones entre componentes. Esto es particularmente importante cuando existen conjuntos de diseñadores distribuidos trabajando sobre diferentes dominios.

  Desde un enfoque formal, una ontología provee una especificación declarativa de un sistema, la cual permite a los diseñadores razonar sobre "para qué" se está diseñando el sistema en lugar de "cómo" soportar la funcionalidad.

  Confiabilidad : Las ontologías informales pueden mejorar la confiabilidad del sistema sirviendo como base para el chequeo manual del diseño contra la especificación, mientras que las ontologías formales permiten el chequeo de consistencia (semi)automatizado del sistema de software con respecto a la especificación declarativa.

  Reusabilidad: Para ser eficiente, una ontología debe soportar que se puedan importar y exportar mó-

dulos entre diferentes sistemas de software.
Por medio de la caracterización de las clases de dominio y de las tareas dentro de esos dominios, las ontologías pueden proveer un marco de trabajo para determinar que aspectos de una ontología pueden ser reutilizados entre dominios y tareas diferentes. La meta es conseguir librerías de ontologías que puedan ser reutilizadas y adaptadas para distintas clases de problemas y entornos.

Según Gruninger-Lee [9]

- **Comunicación**
  - Entre sistemas computacionales, por ejemplo en el intercambio de datos entre distintas herramientas
  - Entre humanos, por ejemplo para la obtención de un vocabulario que unifique conceptos en un dominio específico
  - Entre humanos y sistemas computacionales, por ejemplo una ontología se puede desplegar en una ventana para que el usuario pueda utilizarla como parte del sistema, para comprender el vocabulario utilizado.
- **Inferencia computacional**
  - Para la representación interna y manipulación de planes e información de planificación.
  - Para el análisis de estructuras internas, algoritmos, entradas y salidas de sistemas en términos conceptuales y teóricos.
- **Reutilización y organización del conocimiento**
  - Para la estructuración u organización de librerías o repositorios de planes e información de planificación y de dominio.

Según Guarino [10]

Al analizar el impacto del uso de ontologías en el campo de los sistemas de información deben tenerse en cuenta dos dimensiones: (1) el **tiempo** en el que son utilizadas y (2) el **aspecto estructural**:

- **Respecto al momento en que son utilizadas**
  El uso de ontologías puede llevarse a cabo durante el desarrollo o en tiempo de ejecución. Cuando la ontología se utiliza en tiempo de ejecución el sistema se dice “dirigido por ontología”, mientras que cuando las ontologías son utilizadas durante el desarrollo del sistema, este se dice “desarrollo dirigido por ontología”:
  - Al utilizar ontologías **durante el desarrollo** puede que se cuente con un conjunto de ontologías reutilizables, organizadas en librerías de ontologías de dominio y de tareas, o que se cuente con una ontología genérica (con distinciones no muy detalladas a nivel dominio entre entidades básicas y distinciones a nivel meta sobre tipos de clases y de relaciones), donde el grado de reusabilidad es limitado. En el primer caso, el contenido semántico de las ontologías puede ser convertido en una componente del sistema, reduciendo los costos del análisis y asegurándose (dado que la ontología se presume correcta) la adecuación ontológica del sistema. En el segundo escenario, que parece más realista, la cantidad de conocimiento ontológico disponible es modesta, pero su calidad puede ayudar al diseñador en su tarea de análisis conceptual. En este caso, la ontología no juega el rol de un bloque fácilmente adaptable y reutilizable, sino el de una poderosa herramienta que incrementa la calidad del proceso de análisis.
  - Al utilizar ontologías **en tiempo de ejecución**, debe distinguirse entre sistemas “con conciencia de ontología” y sistemas “dirigidos por ontología”.
    En el primer caso una componente del sistema tiene conocimiento de la existencia de una (posible)

ontología y puede hacer uso de ella con algún propósito específico; mientras que en el segundo la ontología es una componente más (generalmente local al sistema) que coopera en tiempo de ejecución para alcanzar la meta del sistema.
Una razón para utilizar ontologías en tiempo de ejecución es habilitar la comunicación entre agentes de software, los cuales se comunican entre ellos por medio de mensajes que contienen expresiones de acuerdo a la ontología.

- **Respecto a la dimensión estructural**
  - Utilizar una ontología para la componente **bases de datos** parece ser el uso más obvio, pues en la práctica una ontología puede compararse con el esquema de una base de datos.
    Durante el desarrollo una ontología puede jugar un rol importante en las fases de análisis y de modelado conceptual. El modelo conceptual resultante puede representarse en un formato comprensible por un ordenador y desde allí proyectarse a una plataforma concreta.
    Durante el tiempo de ejecución hay varias maneras en que las ontologías y las bases de datos pueden cooperar. La disponibilidad de ontologías explícitas para los recursos de información son básicas en el enfoque basado en mediación para la integración de información.
  - Respecto a **interfaces de usuario** las ontologías han sido exitosamente utilizadas para generar interfaces basadas en formularios que chequean por restricciones de violación de tipos.
    En tiempo de ejecución, una ontología podría desplegarse en una ventana auxiliar para que el usuario la utilice como parte del sistema, por ejemplo para comprender el vocabulario utilizado.
  - Los **programas de aplicación**, suelen contener mucho conocimiento sobre el dominio en forma implícita, por ejemplo en las declaraciones de tipos o clases, en cuanto a reglas o políticas de negocios, etc.
    Durante el desarrollo se puede generar la parte estática de un programa con ayuda de una ontología. Además, las ontologías integradas con recursos lingüísticos pueden ser utilizadas para soportar el desarrollo de software orientado a objetos tal como se expresó en cuanto a las bases de datos.
    En tiempo de ejecución se puede representar en forma explícita el conocimiento que guarda el programa en forma implícita; convirtiendo el programa en un sistema basado en conocimiento. Esto puede mejorar el mantenimiento, la extensibilidad y la flexibilidad del sistema.

4. Ejemplos de Ontologías en Ingeniería de Software

Es un pensamiento compartido por la comunidad informática que las ontologías pueden ayudar a construir mejores sistemas software y con mayor interoperabilidad. Las características más importantes que las ontologías proporcionan a la Ingeniería del Software son [12]: (1) Una semántica y una taxonomía explícita; (2) Un enlace explícito entre conceptos y relaciones y teorías genéricas; (3) Ausencia de polisemia dentro de un contexto formal; (4) Modularización de contextos; (5) Axiomatización mínima para detallar diferencias entre conceptos similares; (6) Una buena política de elección de nombres; y (7) Una documentación rica.

En Ingeniería del Software puede hacerse uso de ontologías a distintos niveles de generalidad. Por ejemplo, las ontologías a nivel de dominio son especialmente útiles para el desarrollo de software reutilizable de alta calidad, gracias a que las ontologías proveen una terminología no ambigua que puede ser compartida por todos los procesos de desarrollo.

Además, gracias a las ontologías, la etapa de elicitación y modelado de los requerimientos puede ser llevada a cabo en dos fases [6]: en una primera se puede elicitar el conocimiento general del dominio y especificarlo en una o más ontologías, y en una segunda etapa las ontologías obtenidas en la etapa ante-

rior se utilizan como líneas para desarrollar las aplicaciones específicas. La ontología construida a partir de la primer etapa de adquisición de conocimiento sirve como *vocabulario básico* para hablar acerca del dominio y es la base para el desarrollo de las conceptualizaciones específicas de las aplicaciones que se quieren construir.

Otro campo de aplicación posible son los Entornos de Ingeniería de Software (*Software Engineering Environments*; SEE según sus siglas en inglés). Los SEE's combinan técnicas, métodos y herramientas para ayudar a los desarrolladores de software a construir productos software. Dado que en los SEE el conocimiento está embebido en alguna herramienta o en un asistente, este es prácticamente imposible de compartir o de reutilizar [4]. Con este fin se han comenzado a construir entornos basados en ontologías.

A continuación se presentan algunos ejemplos de propuestas que intentan integrar el conocimiento por medio de la utilización de ontologías.

4.1. MANTIS (Entorno para la Gestión del Mantenimiento de Software)

MANTIS [13] es un "SEE extendido", cuyo objetivo es la gestión del mantenimiento de software. Incluye una ontología, común a todos sus elementos, que tiene dos usos principales:

(1) ayudar a la compartición del conocimiento sobre el mantenimiento entre todos los actores que intervienen en el proceso de mantenimiento de software;

(2) filtrar el conocimiento, mediante el uso de modelos y meta-modelos, que por definición muestran sólo una parte de la realidad y que de esta manera ayudan a decidir, en el momento de construir los modelos (del nivel M1 de la arquitectura conceptual basada en MOF), que es lo que debe ser extraído de los sistemas reales.

La ontología general está compuesta por 3 ontologías:

- Ontología de los Flujos de Trabajo (para los aspectos dinámicos)
- Ontología de la Medida (gestionar es medir)
- Ontología del Mantenimiento, que a su vez, se subdivide en 4 subontologías: de los Productos, de las Actividades, de Organización del Proceso, y de los Agentes.

La ontología MANTIS está representada gráficamente mediante diagramas de clase UML que son complementados con una representación textual semi-formal en lenguaje GLEO.

Para cada ontología (y subontología) se definen:

- uno o más **diagramas de clases** que muestran los conceptos de esa ontología y las relaciones entre ellos.
- para cada concepto (representado por una clase UML) se crea una entrada en un **glosario de conceptos** (donde se define su superconcepto, descripción, propósito)
- para cada relación del diagrama UML se crea una entrada en una **tabla de interrelaciones** (donde consta su nombre, descripción)
- para cada atributo de cada concepto se crea una entrada en una **tabla de atributos**, donde consta su nombre, descripción, cardinalidad.
- para formalizar algunos de los aspectos importantes de los conceptos anteriores se definen **axiomas**,

que son utilizados principalmente para representar restricciones.

4.2. TABA Workstation

TABA Workstation es un meta-SEE, capaz de generar, mediante su instanciación, SEE's adecuados para las particularidades de un proceso software, un dominio de aplicación o un proyecto específico. [4]

Dado que el meta-entorno, la instancia del SEE creada y las herramientas en la estación de trabajo TABA necesitan manejar conocimiento sobre el proceso de desarrollo de software, se ha definido una ontología con el fin de "soportar la adquisición, organización, reutilización y compartición del conocimiento del Proceso Software".

La ontología del Proceso de Desarrollo de Software consta de varias ontologías, en forma de niveles, donde la ontología del proceso software está por encima de las ontologías denominadas de la actividad, del procedimiento y del recurso.

Para la representación gráfica de estas ontologías se utiliza el lenguaje gráfico GLEO además de un conjunto de axiomas definidos según el mismo lenguaje, basado en lógica de primer orden. Además, para cada ontología, el vocabulario utilizado es definido en una tabla que cuenta con dos columnas, una para el nombre del concepto y otra para describir su función y la manera en que se relaciona con otros conceptos.

4.3. Ontología del Conocimiento Utilizado durante el Mantenimiento

Las personas involucradas en el proceso de mantenimiento necesitan tener mucho conocimiento además del dominio de aplicación. También es necesario que conozcan de la organización que utiliza el software, del pasado y presente de las prácticas en Ingeniería del Software, de diferentes lenguajes de programación (en sus diferentes versiones), etc. [3]. Para ayudar a las personas encargadas del mantenimiento, esta propuesta apunta a la construcción de herramientas que provean un acceso directo a las diversas fuentes de conocimiento necesarias. Con ese objetivo, los autores han desarrollado una ontología del conocimiento necesario durante el mantenimiento.

Dada la diversidad de conceptos a modelar, la ontología general se organiza alrededor de 5 aspectos: el dominio de aplicación, el sistema, las habilidades (de los programadores), el proceso de modificación y la estructura organizacional. Cada uno de estos aspectos está descrito por una subontología.

Para definir la ontología se utiliza una notación gráfica propia con la idea de sólo plasmar el conocimiento y luego formalizarlo mediante alguna de las herramientas disponibles para editar ontologías. Por la misma razón se utiliza lógica de primer orden para formalizar las restricciones.

Posteriormente a la construcción de la ontología, los autores la validaron por medio de encuestas a expertos en mantenimiento y a la confrontación de la ontología con el trabajo de personas en situaciones reales de mantenimiento. En ambas evaluaciones obtuvieron que cerca del 70% de los conceptos definidos fue utilizado en las pruebas.

5. Clasificación de las ontologías de ejemplo

A continuación se presenta una clasificación de las ontologías comentadas en los apartados anteriores

respecto a las clasificaciones de uso, de generalidad, etc. presentadas en secciones previas. Esta clasificación está basada en el análisis de cada una de las características ya referidas en las ontologías y subontologías involucradas.

- **Respecto al nivel de generalidad**: La ontología para el entorno de gestión del proceso de mantenimiento MANTIS (sección 4.1) define conceptos básicos (a alto nivel) que se manejan durante el proceso de mantenimiento. Los mismos serán instanciados, al aplicarse en un dominio específico, para identificar actividades, productos, procesos, etc. Lo mismo ocurre en el enfoque TABA. Los términos utilizados en la ontología tienen por objetivo representar un vocabulario en forma genérica sobre las distintas etapas del proceso software, para que, por medio de una instanciación del meta-entorno se genere un SEE adecuado a las particularidades de un proceso software específico. Respecto al enfoque de la ontología del conocimiento utilizado en mantenimiento, no se intenta determinar en forma genérica las actividades, productos, etc involucrados en el proceso de mantenimiento como en MANTIS, sino que su objetivo es definir un marco de trabajo a ser utilizado en herramientas adecuadas para brindar información a los encargados del mantenimiento. Por ello su nivel de generalidad corresponde al nivel representacional (según Fensel).
- **Respecto al tipo de estructura de conceptualización**: De acuerdo a la clasificación propuesta por van Heist, en los tres casos presentados las estructuras internas expresan conceptos, relaciones entre ellos y restricciones. Por lo tanto, no se ajustan a la clasificación terminológica puesto que no son simples vocabularios unificados, ni tampoco a la clasificación de información pues su objetivo va más allá de estandarizar un formato de almacenamiento. En los tres casos se trata de una ontología que intenta modelar una conceptualización en forma completa.
- **Respecto a los aspectos del mundo real que intentan modelar**: En los tres casos, además de incluir términos que modelan aspectos estáticos y dinámicos (especialmente referidos a procesos), las ontologías incorporan conceptos referidos a los agentes y a los roles que estos ocupan en la organización.
- **Respecto a la riqueza de su estructura interna y al sujeto de conceptualización**: De acuerdo al primer enfoque de la clasificación propuesta por Gómez-Pérez, las tres ontologías presentadas expresan restricciones lógicas generales, es decir que son del tipo más expresivo pues utilizan lógica de primer orden para expresar restricciones entre los términos de la ontología.
  De acuerdo al segundo enfoque, las ontologías presentadas a modo de ejemplo parecerían encuadrarse dentro de la clasificación “ontologías de tareas”, puesto que describen vocabulario relacionado a actividades genéricas dentro del proceso software y del mantenimiento, proveyendo un vocabulario de términos utilizados en dichos campos pero a su vez estos pueden aplicarse a distintos dominios.
- **Respecto a su uso en Ingeniería del Software:** Según lo expresado por sus autores, en las dos primeras propuestas se especifica que la intención es que las ontologías definidas sirvan como vocabulario de unificación respecto al proceso de mantenimiento (en el primer caso) y del proceso software (en el segundo), a fin de facilitar la comunicación entre las personas involucradas en esos procesos. En el caso primero, además, se especifica que su intención es la de filtrar conocimiento, mediante la organización del mismo en distintos niveles. Respecto a la clasificación de Uschold-Gruninger, este objetivo no se puede representar, pero si respecto a la clasificación de Gruninger-Lee. En los casos segundo y tercero se plantea como objetivo que el conocimiento adquirido sea reutilizado en otros proyectos. Dentro de la clasificación dada por Guarino, las tres ontologías analizadas, por su objetivo, están orientadas a facilitar el proceso de desarrollo.

A fin de obtener una visión comparativa de la clasificación hecha según los distintos criterios expresados anteriormente, en la tabla 1 se los muestra en forma compacta.

Tabla 1: Resumen de la comparativa entre las ontologías analizadas.

	MANTIS	**TABA**	**Ontología del Conocimiento en Mantenimiento**
Nivel de generalidad	De Alto Nivel	de Alto Nivel	Representacional
Tipo de estructura conceptualización	De modelado de conocimiento	De modelado de conocimiento	De modelado de conocimiento
Aspectos que modela	Estáticos Dinámicos Intencionales Sociales	Estáticos Dinámicos Intencionales Sociales	Estáticos Dinámicos Intencionales Sociales
Riqueza de la estructura interna	Expresa restricciones lógicas generales	Expresa restricciones lógicas generales	Expresa restricciones lógicas generales
Tipo de sujeto de conceptualización	De Tareas	De Tareas	De Tareas
Uso según Uschold-Gruninger	Comunicación	Comunicación Reusabilidad	Reusabilidad
Uso según Gruninger-Lee	Comunicación (entre humanos) Reutilización y organización del conocimiento	Comunicación (entre humanos) Reutilización y organización del conocimiento	Reutilización y organización del conocimiento
Uso según Guarino	Durante el desarrollo	Durante el desarrollo	Durante el desarrollo

6. Conclusiones

Con el pasar de los años las ontologías han dejado de ser teorías destinadas a la investigación en el campo de la inteligencia artificial para convertirse en herramientas que asisten eficientemente en las tareas de desarrollo y mantenimiento de sistemas software, y que pueden ser utilizadas tanto para el análisis de un dominio específico como para la especificación de una base de conocimiento.

En las primeras secciones de este trabajo se han presentado distintas clasificaciones de ontologías con el propósito de permitir comprender por qué dos ontologías pueden presentar diferencias aún cuando se refieren a un mismo dominio. Aunque puede deberse a que la conceptualización ha sido hecha por distintos grupos de personas, puede existir un segundo motivo y es el enfoque que se le quiso deliberadamente dar a la ontología, es decir, su objetivo final. Por ello, estas clasificaciones han sido analizadas, a modo de ejemplo, en tres casos de ontologías aplicadas a procesos de Ingeniería del Software. La clasi-

ficación lograda a partir de dicho análisis permite una comprensión más acabada de los conceptos vertidos teóricamente en las secciones primeras, pero también sirve como marco de referencia para que, quienes estén interesados en construir una nueva ontología puedan utilizarlo para decidir de qué manera quieren que esta se encuadre respecto a su objetivo final y quienes estén en la búsqueda de ontologías para su reutilización puedan encontrar en este tipo de marco de referencia información útil para su objetivo.

7. Referencias

1 Chandrasekaran B., Josephson J.R., Benjamins V.: Ontology of Tasks and Methods,. In Proceedings of KAW'98, Inn, Banff, Alberta, Canada, (1998).

2 Chandrasekaran B., Josephson J.R., Benjamins V.: What Are Ontologies, and Why Do We Need Them?. IEEE Intelligent Systems, v.14 n.1, (1999) pp 20-26.

3 Dias M., Anquetil N., Oliveiva K.: Organizing the Knowledge Used in Software Maintenance. Journal of Universal Computer Science (J.UCS), vol. 9, no. 7, (2003) pp 641-658.

4 Falbo R., Menezes C., Rocha A.: Using Ontologies to Improve Knowledge Integration in Software Engineering Environments. In Proceedings of the World Multiconference on Systemic, Cybernetics and Informatics / 4th International Conference on Information Systems Analysis and Synthesis, SCI'98/ISAS'98, Orlando, USA, (1998).

5 Fensel, D: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag, Second Edition, Berlin, Heidelberg (2004).

6 Girardi, R., Faria C.: A Generic Ontology for the Specification of Domain Models. In Proceedings of 1st International Workshop on Component Engineering Methodology (WCEM'03) at Second International Conference on Generative Programming and Component Engineering. Erfurt, Germany, (2003).

7 Gómez Pérez, A., Fernández López, M. Corcho, O.: Ontological Engineering. Springer-Verlag, London (2004).

8 Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition, 5, (1993) pp 199-220.

9 Gruninger, M. y Lee, J. (2002): Ontology Applications and Design. Communications of the ACM. 45(2), (2002) pp. 39-41 .

10 Guarino, N.: Formal Ontology in Information Systems. In Proceedings of FOIS'98, Trento, Italy, IOS Press, Amsterdam, (1998).

11 Jurisica I., Mylopoulos J., Yu E.: Using ontologies for knowledge management: An information systems perspective. In Proceedings of 62nd Annual Meeting of the American Society for Information Science (ASIS99), (1999) pp 482-496.

12 Pisanelli D.M., Gangemi A., Steve G.: Ontologies and Information Systems: the Marriage of the Century?. In Proceedings of Lyee Workshop, Paris, (2002)

13 Ruiz F., Vizcaino A., Piattini M. y Garcia F.: An Ontology for the Management of Soft-ware Maintenance Projects. International Journal of Software Engineering and Knowledge Engineering, Vol. 14, No. 3, (2004) pp 1-27 (próxima publicación).

14 Uschold M., Gruninger M.: Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, Vol. 11, No. 2, (1996) pp 93-115.

15 Van Heijst, G., Schereiber, A.T. Y Wielinga, B.J.: Using Explicit Ontologies in KBS Development. International Journal of Human and Computer Studies, (1996).

16 Wang X., Chan C., Hamilton H.: Design of knowledge-based systems with the ontology-domain-system approach. In Proceedings of SEKE 2002, (2002) pp 233-236.