



Universidad Nacional de La Plata

Facultad de Informática

Tesis presentada para obtener el grado de Magister en Ingeniería de Software

Normalización de Información en Catálogos de Componentes OTS para Sistemas de Información Geográficos

Tesista: *Gabriela Gaetán*

Directora: *Alejandra Cechich*

CoDirectora: *Silvia Gordillo*

Septiembre 2011

DEDICATORIA

A mi hija Camila

AGRADECIMIENTOS

A Alejandra Cechich por su constante buena disposición y dedicación, y por compartir generosamente sus conocimientos.

A Agustina Bucella por sus comentarios y sugerencias durante todo el proceso de desarrollo de este trabajo.

A mis compañeros de la Universidad Nacional de la Patagonia Austral, especialmente a Viviana Saldaño, con quien compartí mucho tiempo aprendiendo y superando obstáculos.

A mis padres y a mis hermanas, por alentarme a lo largo de la carrera y por estar siempre a mi lado.

RESUMEN

Con el surgimiento del desarrollo de software basado en componentes, numerosas empresas fabricantes de SIG han comenzado a comercializar distintos tipos de componentes software orientados a las necesidades de los desarrolladores SIG. Para lograr un desarrollo más eficiente, los analistas se concentran en los atributos de reusabilidad e interoperabilidad. Sin embargo, se pierde mucho tiempo y esfuerzo en encontrar aquellos componentes que satisfagan la funcionalidad que se pretende implementar. Una de las necesidades clave para facilitar esta tarea, consiste en contar con información estándar de los componentes que permita agilizar la búsqueda de composiciones de software.

Considerando que uno de los problemas en el crecimiento del desarrollo de software basado en componentes es la dificultad para localizar y recuperar componentes de software existentes, nuestra propuesta se concentra en las ventajas que el conocimiento de un dominio específico aporta a la selección de componentes. De esa manera, estándares del dominio geográfico se usan para normalizar la información y permitir el modelado de una base conceptual que derive en una búsqueda automática.

En ese contexto, esta tesis presenta un proceso para publicar información sobre componentes SIG almacenada en repositorios Web, estructurándola en base a un esquema de información normalizado y enriquecido por medio de técnicas de Procesamiento del Lenguaje Natural.

La contribución de esta tesis se centra precisamente en la presentación de un prototipo que recibiendo información de catálogos de componentes SIG publicados en la Web, normaliza y clasifica la información de manera automática. Se describen los principales elementos de la herramienta que automatiza este proceso y se evalúan los resultados experimentales de un caso de estudio.

ABSTRACT

Because of the wide-spreading use of component-based developments, numerous companies that develop Geographic Information Systems (GIS) have begun to commercialize different types of software components oriented to the needs of developers. Then, to achieve a more efficient development, analysts concentrate on the attributes of reusability and interoperability.

However, selecting inappropriate components may result in increasing time and cost of software development. One of the key aspects to facilitate this task consists of relying on standard information of the components that allows improving the search of compositions of software.

In this context, this thesis presents a process to publish information about GIS components stored in Web repositories, structuring it on the basis of a standardized information scheme and enriched by means of the use of standards and Natural Language Processing (NLP) techniques.

The contribution of this thesis centers precisely on the presentation of a prototype that by receiving information of catalogs of GIS components published on the Web, automatically normalizes and classifies the information. This document describes the main elements of the tool that automates this process and introduces experimental results from a case study.

INDICE

1.	INTRODUCCION	13
1.1.	Objetivos	19
1.2.	Metodología.....	19
1.3.	Contexto	21
1.4.	Estructura de la Tesis	21
2.	ESTADO DEL ARTE Y FUNDAMENTOS DE ESTA TESIS.....	23
2.1.	Clasificación y Almacenamiento de Componentes OTS	23
2.1.1.	Clasificación de Componentes	24
2.1.1.1.	Framework Unificado de Especificación de Componentes (USCS).....	28
2.1.2.	Almacenamiento y Búsqueda de Componentes OTS.....	30
2.2.	Componentes para SIG.....	33
2.2.1.	Clasificación de Servicios Geográficos.....	34
2.2.2.	Estándares Internacionales para SIG.....	40
2.2.3.	Repositorios de Componentes SIG	42
2.3.	Uso de Técnicas Semánticas y del Lenguaje Natural en la Extracción de Información	43
2.3.1.	Extracción de Información	44
2.3.2.	Reconocimiento de Entidades Nombradas.....	47
2.3.3.	Descubrimiento de Patrones.....	49
2.3.4.	La Web Semántica.....	50
2.3.5.	Ontologías	52
2.3.6.	Anotación Semántica.....	53
2.3.6.1.	Sistemas de Anotación Manuales.....	55
2.3.6.2.	Sistemas de Anotación (semi)Automáticos.....	56

2.3.7.	Extracción de Información Basada en Ontologías	58
3.	NORMALIZACIÓN Y PUBLICACION DE DESCRIPCIONES DE COMPONENTES SIG	60
3.1.	Normalización de Descripciones de Componentes SIG.....	60
3.2.	Publicación de Componentes SIG.....	69
3.3.	Creación de la Ontología.....	71
3.4.	Extracción de Información	73
3.5.	Normalización de información.....	75
3.6.	Herramienta de Soporte.....	76
3.7.	Primer Prototipo	79
3.8.	Escenario de Uso	83
4.	EVALUACIÓN EXPERIMENTAL.....	86
4.1.	Definición.....	86
4.2.	Planificación.....	86
4.3.	Operación	88
4.4.	Análisis de Resultados.....	94
4.5.	Discusión.....	96
5.	CONCLUSIONES	99
5.1.	Análisis de la Consecución de Objetivos	99
5.2.	Principales Aportaciones.....	100
5.3.	Contrastación de Resultados	101
5.4.	Trabajos Futuros.....	102
6.	REFERENCIAS	104
	Apéndice A.....	112
	Apéndice B.....	118

1. INTRODUCCION

Las aplicaciones basadas en componentes se construyen como un conjunto de elementos preensamblados en un marco de aplicación específico. Esos marcos median y regulan las interacciones entre componentes. En ese contexto, las aplicaciones implementan nuevos procesos a partir de la infraestructura tecnológica. El proceso de desarrollo en su conjunto es guiado por un enfoque arquitectónico que facilita la confección de soluciones a través del uso de elementos existentes en un depósito de modelos y componentes de software.

Actualmente, los sistemas de software se tornan cada vez más complejos y brindan mayor funcionalidad. Para desarrollar estos sistemas, el Desarrollo de Software Basado en Componentes (DSBC) apela al uso de piezas pre-fabricadas, tal vez desarrolladas en momentos diferentes, por distintas personas y posiblemente con distintos objetivos de uso [64], [72]. La meta final, una vez más, es la reducción del tiempo de desarrollo, de los costes y del esfuerzo, a la vez que se mejora la calidad del producto final debido a la reutilización de componentes software ya desarrollados, probados y validados.

Los *componentes*, la parte central de un DSBC, se definen de varias maneras y desde diferentes puntos de vista. Si bien en la literatura existe el consenso de que los componentes COTS (Commercial Off-The-Shelf) son una clase de componentes reusables, este es un término que se aplica a una amplia variedad de productos ya sea de software o de hardware (o de ambos).

En [71] se definen componentes COTS como productos de software pre-existentes, que se venden en múltiples copias con mínimos cambios. Los clientes no tienen control sobre su especificación, calendario, y evolución, y tampoco tienen acceso al código fuente ni a la documentación interna.

En el trabajo presentado en [9] se define a los componentes COTS en términos de su origen y modificabilidad. Según su origen los componentes COTS son: componentes producidos por contrato, componentes existentes de fuentes externas, o componentes producidos en casa. Según su modificabilidad, los componentes COTS requieren re

trabajo extensivo del código, revisión de código interna, adaptación y personalización, parametrización simple, o modificación escasa o ninguna.

También Basili y Bohem [4] han propuesto otra definición según la cual, en un componente COTS el comprador no tiene acceso a su código fuente y quien controla su desarrollo y evolución es el proveedor.

En lo que se refiere a esta tesis, adoptaremos la definición propuesta en el trabajo de Torchiano y Morisco [66], en la cual se establece que: “Un producto COTS es una pieza de software disponible comercialmente o como Open Source, que otro proyecto puede reusar e integrar en sus propios productos”. Esta definición indica explícitamente que los productos Open Source también se consideran productos COTS. En este sentido consideramos la Open Source Definition (OSD)¹, donde se definen los criterios que debe cumplir una distribución de software Open Source, entre ellas: libre redistribución, código fuente incluido o de libre obtención, permiso para redistribución de modificaciones, no discriminación de personas o grupos, integridad del código fuente del autor, no exclusión de usuarios comerciales, iguales derechos a todo el que reciba el programa, la licencia no debe ser específica de un producto, la licencia no debe restringir otro software y la licencia debe ser tecnológicamente neutral.

Un modelo de componentes brinda una descripción abstracta que suministra a los desarrolladores de software una concepción uniforme sobre cómo construir componentes individuales y sobre cómo interactúan y se comunican entre sí. Por lo tanto, los modelos de componentes implican trabajar con composiciones de componentes o “*combinaciones de dos o más componentes software que producen un nuevo comportamiento a un nivel de abstracción diferente*” [37].

El DSBC cambia el enfoque de la ingeniería de software de una construcción tradicional a otra que requiere la consideración simultánea del contexto del sistema (características como requisitos, coste, tiempos, sistema operativo y entornos de soporte), capacidades de los productos en el mercado y arquitecturas de software viables.

¹ OSD (Open Source Definition). Consultado 2010. Disponible en: <http://www.opensource.org/docs/definition.php>.

El ciclo de desarrollo para un sistema basado en componentes es diferente con respecto a los tradicionales – modelo en cascada, iterativo, espiral o basado en prototipos. La Figura 1 (de [14]) muestra una comparación entre el modelo en cascada tradicional y el modelo para desarrollo basado en componentes. Los diferentes pasos en este proceso son:

- (1) Encontrar componentes (COTS y no COTS).
- (2) Seleccionar los componentes más adecuados para el sistema.
- (3) Crear una solución compuesta que integre la selección previa.
- (4) Adaptar los componentes seleccionados de manera que se ajusten al modelo de componentes o a los requisitos de la aplicación.
- (5) Componer y distribuir el producto.
- (6) Remplazar versiones anteriores o mantener las partes COTS y no-COTS del sistema.

En la Figura 1, las actividades “Requisitos” y “Diseño” del modelo en cascada se corresponden con las actividades “Buscar” y “Seleccionar” componentes. “Implementación”, “Prueba” y “Distribución” se corresponden con “Crear”, “Adaptar”, “Distribuir” y “Reemplazar”. Los primeros dos pasos – búsqueda y selección de componentes – son los más importantes del ciclo. Usar componentes que no sean los adecuados para el desarrollo de un sistema puede convertirse en extremadamente costoso para una empresa. En esta etapa, deberán tomarse decisiones a largo plazo sobre qué componentes serán empleados en el sistema.

Además de los problemas comunes a la reutilización (selección, integración, mantenimiento, etc.), los productos OTS presentan problemas específicos:

- *Incompatibilidad*: los componentes OTS pueden no tener la funcionalidad exacta requerida; es más, pueden no ser compatibles con software desarrollado en la empresa o con otros productos OTS.
- *Inflexibilidad*: por lo general, el código fuente de un componente OTS no se suministra, por lo que no puede ser modificado.

- *Complejidad*: los productos OTS pueden ser demasiado complejos para ser entendidos y usados rápidamente, lo que impone un esfuerzo de aprendizaje adicional importante.
- *Cambio de versiones*: diferentes versiones del mismo producto OTS pueden no ser compatibles, causando problemas adicionales durante el desarrollo.

Por lo tanto, establecer métodos sistemáticos y repetibles para evaluar y seleccionar componentes OTS es un aspecto importante para el DSBC.

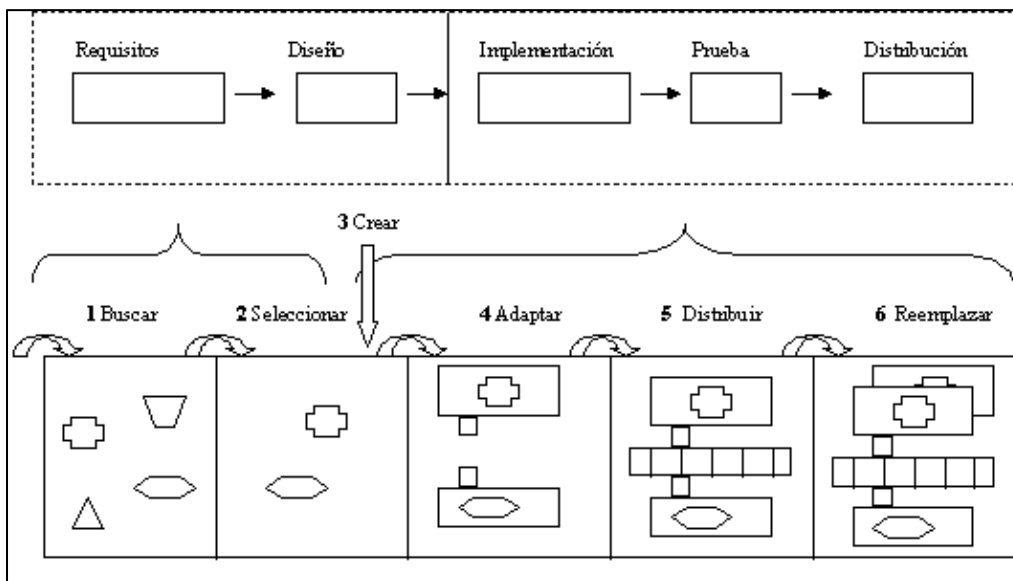


Figura 1. Ciclo de DBC comparado con el modelo en cascada (de [14])

Sin embargo, el proceso de desarrollo de aplicaciones se encuentra influenciado por la documentación de los componentes existentes, así como de otros factores que limitan su selección.

La documentación es una clave para el éxito de un DSBC. Una documentación de alta calidad asegura que los estándares de diseño e implementación se reflejen en el contenido de la aplicación construida con componentes. Por ejemplo, un lenguaje conceptual puede permitir la inclusión explícita de enlaces entre las propiedades de un componente y sus interacciones. En general, no se necesita conocer todos los detalles de un componente a la hora de realizar una composición, pero debería poder seleccionarse componentes que revelen poseer, al menos, la funcionalidad requerida para resolver un

problema específico. La documentación del componente podría extenderse permitiendo agregar detalles sobre la adaptación directamente en una biblioteca de documentos o dentro del mismo componente. Además, la documentación debería adecuarse a las herramientas usadas para componer. De esa manera, los componentes seleccionados podrían incorporarse fácilmente a una biblioteca de reutilización.

Por otra parte, con el surgimiento del desarrollo de software basado en componentes, numerosas empresas fabricantes de sistemas de información geográfica (SIG) han comenzado a comercializar distintos tipos de componentes software orientados a las necesidades de los desarrolladores SIG. Sin embargo, dado que las aplicaciones SIG son utilizadas en dominios muy diferentes, cada aplicación posee un punto de vista distinto sobre la información geográfica. Cada desarrollador ha definido modelos conceptuales, modelos de datos geográficos, formatos de almacenamiento, operaciones de análisis o procedimientos de representación especialmente adaptados a los requerimientos de la aplicación. Por lo tanto, en el proceso de desarrollo de aplicaciones SIG, encontrar información adecuada sobre componentes geográficos es una tarea crucial; los catálogos disponibles en la web ofrecen descripciones de los componentes, pero los mecanismos de selección, generalmente basados en búsqueda de palabra-clave, son poco efectivos.

En consecuencia, el DSBC para aplicaciones de dominio SIG, es un proceso complejo, en el que intervienen por un lado el desarrollador del componente y por otro, el usuario del componente, definiéndose así dos modelos, de *demanda* y de *oferta*, definiendo requerimientos especiales y soluciones de acuerdo a las restricciones y problemas que involucra cada uno de ellos.

El *modelo de demanda* hace referencia a los requerimientos o servicios solicitados por los clientes de SIG (empresas que desarrollan SIG), los cuales, en un desarrollo basado en componentes, buscan en la Web, componentes que brinden servicios que se adecuen a las necesidades de sus aplicaciones.

El *modelo de oferta* se refiere a los requerimientos o servicios que brindan los desarrolladores de componentes para SIG, para lo cual publican en la Web los

componentes desarrollados junto a la información técnica y funcional asociada a los mismos.

Por lo tanto, para que ambas partes logren entenderse, es fundamental la forma en que los componentes son publicados, llamado *proceso de publicación*, y la forma en que los mismos son seleccionados, llamado *proceso de selección*. Además, para integrar la funcionalidad de ambos procesos surge la necesidad de utilizar un *modelo de mediación* que facilite tanto la recuperación de la información como la selección automatizada de componentes candidatos [41].

Las características particulares de los SIG provocan que las composiciones de componentes no puedan construirse sólo como simples piezas de un producto, sino como un conjunto de elementos pre-ensamblados en un marco de aplicación específico. Una documentación de alta calidad basada en estándares aplicables a un dominio debería facilitar el proceso de clasificación y modelado de la información.

Existen múltiples definiciones de los Sistemas de Información Geográficos, pero entre las más reconocidas se encuentra la propuesta por National Center for Geographic Information & Analysis (www.ncgia.ucsb.edu), que lo define como un “sistema compuesto por hardware, software y procedimientos para capturar, manejar, manipular, analizar, modelizar y representar datos georreferenciados, con el objetivo de resolver problemas de gestión y planificación”.

En el ámbito de la información geográfica, existen organizaciones como International Organization for Standardization² (ISO) y Open Geospatial Consortium³ (OGC) dedicadas a la creación de especificaciones de estándares para facilitar la interoperabilidad entre implementaciones creadas por distintos desarrolladores. Estas especificaciones tratan distintos tópicos asociados con la representación, manipulación y visualización de información geográfica (Ej. familia de estándares ISO 19100 [27], [28], [29], [30], [31]). En particular el estándar ISO/IEC 19119 [30], relacionado con la

² <http://www.iso.org/>

³ <http://www.opengeospatial.org/>

semántica del procesamiento de información geográfica, define una taxonomía formada por las siguientes categorías de servicios geográficos: Interacción humana, Administración de información/modelo, Gestión de tareas/workflow, Procesamiento geográfico, Comunicación geográfica, y Administración de sistemas geográficos. Esta familia de estándares ISO 19100, junto con otras fuentes de información, podría contribuir a normalizar las descripciones de componentes OTS para SIG, facilitando de esa manera el proceso de publicación (oferta de componentes disponibles).

El presente trabajo abordará distintas temáticas en la mejora de la selección y aplicación de componentes OTS (*off-the-shelf*) al entorno de SIG (Sistemas de Información Geográficos), centrandó el estudio en (1) mejoras al proceso de definición de servicios ofrecidos y (2) incorporación de técnicas semánticas en el proceso de publicación, como medio para mejorar la posterior búsqueda y selección de componentes.

1.1. Objetivos

El objetivo de este trabajo es *desarrollar un esquema normalizado para describir componentes para Sistemas de Información Geográficos (SIG) comerciales o libres, de manera que la información pueda ser estandarizada para el proceso de búsqueda. Se considerarán aspectos funcionales y no técnicos.*

Objetivos específicos:

- a. Proponer un esquema de clasificación para describir información funcional y no técnica de un componente SIG.
- b. Proponer técnicas de recolección y adecuación de la información almacenada en portales Web para componentes SIG .
- c. Evaluar la propuesta con niveles de confianza de la información.

1.2. Metodología

En esta tesis se han combinado técnicas que provienen del procesamiento del lenguaje natural y de la web semántica para mejorar la publicación de componentes. Se ha trabajado bajo el paradigma de investigación-acción (I-A) que ha obtenido una amplia

aceptación y aplicación en la investigación en ingeniería del software en los últimos años. Este método presenta como principales características: Orientación a la acción y al cambio, focalización en un problema, un modelo de proceso “orgánico” que engloba etapas sistemáticas y algunas veces iterativas, y la colaboración entre los participantes. De las diferentes variantes de la I-A, aplicaremos la denominada “participativa”. Este método permite generar un beneficio al “cliente” de la investigación y, al mismo tiempo, generar “conocimiento de investigación” relevante. Por tanto, Investigación-Acción es una forma de investigar de carácter colaborativo que busca unir teoría y práctica entre investigadores y profesionales mediante un proceso de naturaleza cíclica.

También en el contexto de esta tesis se ha realizado el diseño e implementación de un caso de estudio para comprobar la aplicabilidad de nuestra propuesta. Los casos de estudio se utilizan para monitorizar proyectos, actividades o asignaciones. En este tipo de estrategia, los datos se recogen para un propósito específico. Un caso de estudio está orientado normalmente a analizar un determinado atributo o establecer relaciones entre diferentes atributos. El nivel de control de un caso de estudio es más bajo que en un experimento. Los casos de estudios son estudios observacionales, es decir, se llevan a cabo mediante la observación de un proyecto o actividad que está en marcha, mientras que los experimentos son estudios controlados [78].

Desarrollar un caso de estudio requiere de cinco etapas principales [74]:

1. **Definición:** En esta etapa se determina la base del experimento, el “por qué” del mismo. El propósito de la fase es definir los objetivos del experimento, formulado a partir de un problema a resolver.
2. **Planificación:** Después de la definición del experimento tiene lugar la planificación. Si la definición determina por qué se va a realizar el experimento, la planificación establece cómo se llevará a cabo.
3. **Operación:** Cuando un experimento ha sido diseñado y planificado debe llevarse a cabo para recoger los datos que se analizarán. En la etapa operacional de un experimento, los tratamientos son aplicados a los sujetos, es decir cuando el experimentador se encuentra con los sujetos.

-
4. **Análisis e Interpretación.** Una vez recogidos los datos empíricos, deben ser analizados adecuadamente. Hay tres elementos principales a considerar cuando se eligen las técnicas de análisis:
 - la naturaleza de los datos que se han recogido,
 - por qué se ejecuta el experimento,
 - el tipo de diseño experimental.
 5. **Presentación y Empaquetamiento.** Una vez realizado un experimento, los resultados obtenidos pueden publicarse en artículos, en informes para la toma de decisiones o como material educacional.

1.3. Contexto

Esta tesis se desarrolló en el marco de dos proyectos de investigación:

- Proyecto UNPA (Universidad Nacional de la Patagonia Austral) 29/B090. Título del proyecto: “Identificación de Servicios para Sistemas de Información Geográficos (SIG)”. Dirigido por la Dra. Alejandra Cechich, en el período 01-01-2008 al 31-12-2009.
- Proyecto UNPA (Universidad Nacional de la Patagonia Austral) 29/B107. Título del proyecto: “Mejora del Proceso de Selección de Componentes para Sistemas de Información Geográficos”. Dirigido por la Dra. Alejandra Cechich, en el período 01-01-2010 al 31-12-2011.

1.4. Estructura de la Tesis

En el capítulo 2 se presentan las nociones básicas del problema de estudio, por un lado los principales conceptos relacionados con la clasificación y almacenamiento de los componentes OTS, en particular, de los componentes para desarrollo de aplicaciones SIG. Por otro lado se muestran técnicas de procesamiento de lenguaje natural, y las características principales que favorecen la recuperación de información.

En el capítulo 3 se describe nuestra propuesta para normalizar la información de componentes por medio de un esquema de clasificación adaptado a estándares del dominio de los componentes SIG. También se describen las herramientas y el prototipo que implementan el proceso de Publicación propuesto, junto con un escenario de uso ilustrativo.

En el capítulo 4 se realiza la evaluación de la propuesta. Se determinan métricas genéricas sobre la información identificada automáticamente en las descripciones de los catálogos web y se muestra su aplicación.

Finalmente, en el capítulo 5 se presentan conclusiones, las principales contribuciones de la tesis y los trabajos futuros.

2. ESTADO DEL ARTE Y FUNDAMENTOS DE ESTA TESIS

2.1. Clasificación y Almacenamiento de Componentes OTS

Las distintas propuestas de procesos de DSBC en general se componen de diferentes etapas: identificación, evaluación, selección, integración y actualización de componentes, tal como se describe en [53]. Entre ellas, la evaluación y selección de componentes involucran aspectos complejos de identificación de propiedades – técnicas y no técnicas – que actualmente son materia de investigación y mejora.

La selección de componentes OTS (Off-The-Shelf) se puede ver como un proceso compuesto en el que intervienen por un lado el desarrollador del componente y por otro, el desarrollador de la aplicación. El desarrollador del componente es el responsable de suministrar la información necesaria para la posterior búsqueda, entendimiento y decisión sobre el uso del componente desarrollado.

A pesar de no existir un método comúnmente aceptado para la selección de componentes OTS, en todos ellos es posible reconocer dos subprocesos principales, uno relacionado con las actividades de publicación, y otro relacionado con las actividades de búsqueda (o más precisamente con actividades de recuperación).

Según se describe en [49], el proceso genérico de selección de componentes OTS está formado por el “Proceso de publicación”, que consiste en:

- (1) la clasificación del componente,
- (2) la documentación del componente y
- (3) el almacenamiento de toda la información en el repositorio.

Y por el “Proceso de Recuperación”, que consiste en:

- (1) la búsqueda de la funcionalidad y otros atributos de calidad deseados,
- (2) el entendimiento técnico de los componentes candidatos encontrados y
- (3) la toma de decisiones sobre su utilización.

En la Figura 2 se describe este proceso de selección y las relaciones entre las distintas actividades.

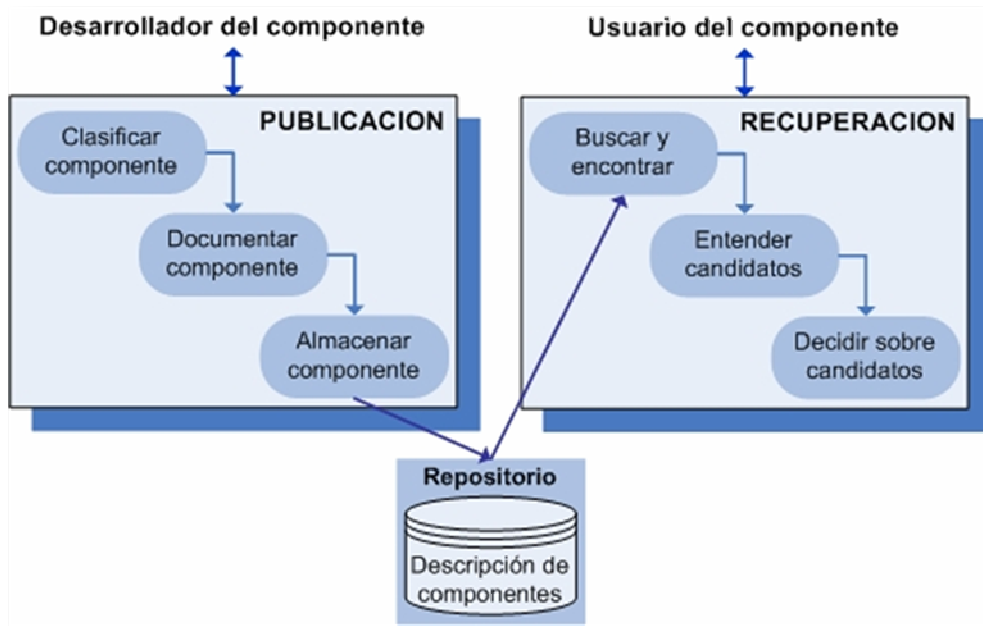


Figura 2. Proceso de selección de componentes

Uno de los problemas difíciles de resolver, tanto en el proceso de Publicación como de Recuperación, es la falta de documentación estándar para describir los componentes OTS. Los desarrolladores necesitan información más precisa sobre requerimientos técnicos (funcionales y no funcionales) y requerimientos no-técnicos que tienen que proveer los componentes para permitir la evaluación de la calidad de los mismos.

2.1.1. Clasificación de Componentes

Como hemos visto, tanto en la literatura como en la práctica existen distintas propuestas que tratan de resolver algunos de los problemas relacionados con la selección de componentes. Aunque, en general, todos los trabajos presentados reconocen la importancia de la clasificación durante la etapa de publicación del proceso de DSBC; en [10] se señala que aún no existe un entendimiento común sobre la información que se

hace disponible sobre los componentes, los modelos de descripción de estos catálogos no están estandarizados, y además, las propuestas existentes son muy diferentes en cuanto a su nivel de caracterización, punto de vista o intención y significado.

Prieto-Diaz & Freeman [60] proponen un esquema de clasificación facetada orientado al reuso de software. La clasificación facetada consiste en descomponer el campo científico o disciplina en sus facetas correspondientes a puntos de vista, perspectivas o dimensiones de un dominio conocido. Cada faceta tiene un conjunto de valores fijos (vocabulario controlado) y un conjunto extensible de términos de usuario. El esquema propuesto está organizado en seis facetas:

- *Función*: es sinónimo de “acción”, se refiere al nombre de la función primitiva específica realizada por el programa.
- *Objeto*: se refiere a los objetos manipulados por el programa, tales como caracteres, líneas y variables.
- *Medio*: se refiere a entidades donde la acción es ejecutada, tales como estructuras de datos (tablas, archivos, árboles, líneas).
- *Tipo de sistema*: se refiere a módulos independientes de la aplicación, identificables funcionalmente; por ejemplo formateador de reportes, analizador léxico, evaluador de expresiones.
- *Area funcional*: describe actividades dependientes de la aplicación, generalmente definidas por un conjunto de procedimientos establecidos. Por ejemplo, auditoría, control de costos, pagos, presupuestos.
- *Contextualización*: describe detalles de cómo conducir ciertas operaciones según el entorno en el cual se implementa la aplicación. Por ejemplo, planta química, ventas por catálogo, venta de autos.

En [46], Kallio & Niemelä proponen una plantilla, en la que participan distintos roles del desarrollo: arquitecto (describe requerimientos funcionales y de calidad), diseñador (describe detalles técnicos), reusador (agrega comentarios de errores y cambios realizados) y mantenedor (actualiza componentes y su historia con información de calidad). La plantilla se organiza en cuatro categorías:

-
- *Información básica*: incluye información general del componente (nombre, identificación, tipo, objetivos de diseño), descripción de su especificación funcional (entradas, salidas, funcionalidad detallada, manejo de datos, excepciones) e interfaces, y diferentes formas de composición.
 - *Información detallada*: esta información debería responder las preguntas: qué información incluye la documentación, por qué se ha producido la información, por quién y en qué etapa del proceso.
 - *Test de aceptación*: esta información describe el entorno de pruebas, soporte de pruebas, datos de prueba, caso de uso usados en las pruebas, métodos y resultados de las pruebas.
 - *Información adicional y de soporte*: esta información describe *bugs* y problemas no resueltos, y su severidad y estado actual. También se incluye algún contacto para obtener ayuda en situaciones problemáticas.

Asimismo, en [65], Torchiano et al. plantean un conjunto de atributos para caracterización de COTS. Este enfoque se focaliza en la descripción de software por medio de atributos externos. En base al estándar ISO 9126 [43] se definen un conjunto de atributos de calidad principales:

- *Seguridad*: es el soporte ofrecido por el producto para desarrollar sistemas críticos seguros.
- *Confiabilidad*: una descripción sobre cuán tolerante a fallas es el producto.
- *Madurez del producto*: descripción de la madurez del producto, por ejemplo, en términos de años en el mercado y estabilidad de las prestaciones.
- *Usabilidad*: es el grado de satisfacción en una escala de 1 a 5 y una descripción mas detallada de calidad de uso.
- *Documentación*: qué clase (web, on-line, etc.) de documentación es provista junto con el producto, y cuál es su tamaño.
- *Aprendizaje*: cuánto tiempo lleva aprender a usar este componente.

-
- *Desempeño*: se entiende como escalabilidad, la cual puede ser medida como el número de usuarios que puede soportar el componente sin disminuir su tiempo de respuesta.
 - *Modificabilidad*: cuán fácil es modificar este producto (no-modificable, parametrizable, provee API).
 - *Conformidad*: lista de estándares a los cuales los productos se adhieren.

Además de estos atributos principales, se agregan otros no relacionados directamente a la calidad en el uso, sino que son atributos externos que sirven para describir componentes de software cuando son comprados: mercado, requerimientos de software, requerimientos de hardware, soporte de producto, tipo de licencia, frecuencia de cambio, costo de adquisición, dominio específico.

En [42], Iribarne et al. presentan un modelo de documentación de componentes soportado por el lenguaje XML, y compuesto por cuatro partes:

- *Descripción funcional*: incluye el conjunto de interfaces que el componente implementa, el conjunto de interfaces que el componente requiere de otros componentes, y el conjunto de eventos emitidos/consumidos.
- *Descripción extra-funcional*: describe los aspectos extra-funcionales o atributos de calidad por medio de propiedades de servicios.
- *Restricciones arquitecturales/de empaquetamiento*: contiene información sobre cómo descargar, desplegar e instalar el componente que provee el servicio requerido, incluyendo detalles de implantación, contexto y restricciones arquitecturales.
- *Información de marketing*: describe otros detalles no-técnicos del servicio. Incluye información del vendedor, aspectos comerciales y de licencia, certificados, soporte del vendedor, nivel de personalización permitido, etc.

Dong et al. [18] sostienen que la información que acompaña a los componentes es escasa, y proponen otra plantilla de especificación de componentes COTS. Los atributos usados para describir los componentes son: nombre del componente, interfaces

funcionales, propiedades no funcionales, aplicabilidad, estándares, componentes relacionados y ejemplos de uso.

2.1.1.1. Framework Unificado de Especificación de Componentes (USCS)

La diversidad de categorizaciones es común al problema de identificación de componentes OTS. Para categorizar la información específica del dominio de los componentes, que luego será usada por el reusador del componente, la terminología utilizada debe tener un significado unívoco. La categorización de la información publicada por el desarrollador debería utilizar un lenguaje claro y estandarizado para evitar conflictos de interpretación con los usuarios que buscan y seleccionan componentes.

En [54] se propone una especificación estándar de componentes de negocios, llamada Framework Unificado de Especificación de Componentes (*Unified Software Component Specifications- USCS*) que fue formulado en base a la experiencia de la Sociedad Alemana de Informática [1] en la estandarización de especificaciones de componentes de negocios, y que mantiene compatibilidad con el Framework de especificación estandarizado UDDI [68].

Su estructura se organiza en grupos temáticos representados por páginas de distintos colores y subdivididos en once aspectos tal como se observa en la Figura 3:

- *Páginas blancas: información general y comercial.* Contiene el nombre del componente, su identificación única, versión, descripción, productor, contactos administrativos, y dependencias a otros componentes. Esta información se refiere tanto a las condiciones de adquisición como de uso.
- *Páginas amarillas: clasificaciones.* Clasificaciones enumeradas y facetadas especifican el dominio al cual pertenece un componente. Además se incluye información sobre la arquitectura subyacente y la tecnología del componente.

- *Páginas azules: funcionalidad.* Se resume información relacionada al dominio y sobre la funcionalidad de un componente. Esta información es provista como un léxico de dominio que contiene conceptos relevantes, definiciones, y relaciones entre conceptos.
- *Páginas verdes: interfaces especificadas contractualmente.* Se provee información referida a las interfaces (provistas y requeridas) del componente, las cuales son necesarias para configurar correctamente e invocar los servicios del componente.
- *Páginas grises: información sobre la implementación.* Se provee información extra-funcional sobre la calidad de la implementación. El framework soporta especificaciones referidas a usabilidad, mantenibilidad, funcionalidad, confiabilidad, y eficiencia de la implementación de un componente.

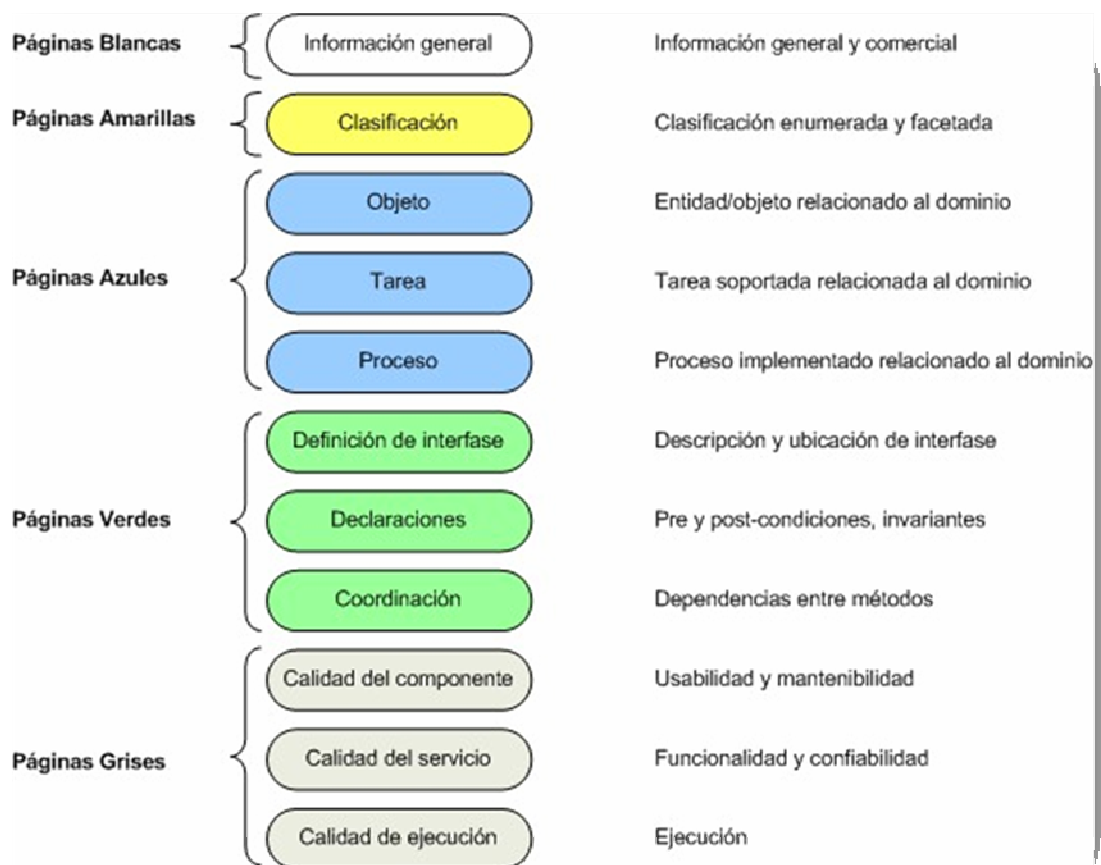


Figura 3. Framework Unificado de Especificación de Componentes

Una de las contribuciones de nuestro trabajo es la adaptación de este Framework Unificado de Especificación de Componentes para la construcción de una taxonomía de componentes SIG, que beneficie tanto a los desarrolladores que publican los componentes como a aquellos que los seleccionan. Esta elección se basa en las siguientes características que ofrece el Framework:

- Su estructura es modular y extensible.
- Soporta diferentes tipos de componentes.
- Mantiene compatibilidad con otras especificaciones estandarizadas (UDDI [68]).
- Determina precisamente qué se especifica y qué notaciones usar para denotar especificaciones.

2.1.2. Almacenamiento y Búsqueda de Componentes OTS

Existen en la literatura distintos aportes sobre búsqueda y almacenamiento de componentes, en [61] se realiza un survey sobre búsqueda de componentes y se propone una lista de requerimientos para un sistema de recuperación de componentes eficiente:

1. *Alta precisión y cobertura*: es un requerimiento básico que debe ser considerado por cualquier mecanismo de búsqueda. Alta precisión se refiere a que los elementos relevantes se recuperan primero. Alta cobertura se refiere que la menor cantidad de elementos relevantes sea dejado de lado.
2. *Seguridad*: es un aspecto a considerar, principalmente, en la búsqueda y recuperación dentro de mercados de componentes distribuidos, donde existen mas posibilidades de que personas no autorizadas quiebren la seguridad. Para resolver este problema se utilizan conocidas técnicas de autenticación y certificación.
3. *Formulación de consultas*: usualmente los componentes son descriptos en términos de su funcionalidad, y las consultas son formuladas en términos del problema; un mecanismo de búsqueda de componentes debe proveer medios

para ayudar al usuario cuando realiza una consulta y favorecer la reducción de este “gap” conceptual.

4. *Descripción de componentes*: durante una búsqueda, el usuario especifica una consulta, la cual luego es emparejada contra una serie de descripciones de componentes. El mecanismo de búsqueda es responsable de decidir cuáles componentes son relevantes para el usuario. Los enfoques existentes describen componentes usando lenguaje natural, o por medio de facetas o aspectos.
5. *Familiaridad con el repositorio*: un mecanismo de búsqueda debería ayudar al usuario a explorar y familiarizarse con los componentes, de modo que en futuras búsquedas sea más fácil localizarlos.
6. *Interoperabilidad*: un mecanismo de búsqueda que opera en repositorios distribuidos debería estar basado en tecnologías estándares, para facilitar su futura expansión e integración con otros sistemas.
7. *Performance*: este aspecto de los mecanismos de búsqueda generalmente se refieren a tiempo de respuesta.

En este trabajo nos enfocamos, por un lado, en la descripción de componentes y la familiaridad con el repositorio y por otro lado, nos concentramos en mejorar la precisión y cobertura; mientras que la seguridad, interoperabilidad y performance quedan fuera del alcance de nuestra propuesta.

Los siguientes son algunos enfoques relevantes en el almacenamiento y búsqueda de componentes.

Agora [62] es una máquina de búsqueda basada en agentes que buscan componentes en la web. Usa la técnica de introspección para descubrir dinámicamente las interfaces sintácticas del componente, que son indexadas y almacenadas por la máquina de búsqueda. Su principal limitación es que se basa solo en aspectos sintácticos del componente para realizar la búsqueda, y no considera otras propiedades.

El sitio web Componex⁴, permite realizar la búsqueda en su repositorio al organizar y clasificar los componentes de software, usando un esquema XML bien definido para especificar distintos parámetros (tecnología, plataforma, dominio, etc.). Este framework permite la clasificación de componentes y la especificación de la mayoría de sus atributos, pero no es flexible para extender las especificaciones de componentes.

ComponentXchange [69] usa un lenguaje de especificación de componentes basado en XML. Está orientada al dominio de comercio electrónico y soporta la incorporación de nuevas categorías.

Los trabajos enumerados hasta aquí no contemplan la especificación de relaciones semánticas entre los componentes, para superar esta limitación han surgido otras propuestas que, para realizar una especificación más descriptiva de los componentes de software, plantean la incorporación de técnicas de Representación de Conocimiento y de Procesamiento de Lenguaje Natural. En [63] se propone un enfoque basado en ontologías y modelos de dominio, orientado a incrementar la efectividad de la búsqueda y brindar información que relacione a los componentes recuperados. Estas ontologías contienen un conjunto de términos relacionados para describir el conocimiento, pero no incluyen axiomas formales.

El trabajo presentado en [75] describe un meta-modelo basado en ontologías para almacenamiento y recuperación de componentes de software, que mejora los enfoques tradicionales. El conjunto de metadatos es enriquecido con principios de ontologías que permiten representar la semántica del dominio e inferir conocimiento.

En [32] se propone un enfoque para recuperación de software basado en el análisis parcial de consultas y descripciones en texto libre de componentes de software. El procesamiento semántico del sistema está basado en la adquisición automática de roles temáticos del texto de entrada a través de la aplicación de procesamiento sintáctico y algunas heurísticas. El enfoque pretende mejorar la efectividad de los sistemas basados en palabras clave y ofrece una interfase amigable con el usuario.

⁴ <http://www.componex.biz>

OntoSeek [7], es una herramienta de propósito general para recuperación de información basada en ontologías. Con muy pocas capacidades de inferencia, pero lo suficientemente rápida y confiable, promete mejorar o reemplazar sistemas de recuperación de información basados en palabras clave. El sistema permite codificar y recuperar la información de los componentes transformando tanto las descripciones como las consultas en ontologías.

Todas las soluciones enumeradas se enfocan sólo en algún problema particular relacionado con la recuperación de información sobre componentes. Algunas soluciones están dirigidas a los aspectos relacionados con la clasificación; mientras que otras están enfocadas en aspectos de documentación y almacenamiento de la información relacionada con los OTS.

2.2.Componentes para SIG

Desde un punto de vista técnico, para que la comunidad de usuarios pueda construir diferentes aplicaciones SIG en base a componentes reusables, éstos deben ofrecer un conjunto de servicios que satisfagan tanto requerimientos funcionales como no funcionales de los SIG. De acuerdo a la funcionalidad que debe brindar una aplicación SIG, se puede clasificar en:

- *Ingreso y verificación de datos:* Esto se relaciona con todos los aspectos de la captura de datos geográficos, la verificación de su correctitud y la conversión a un formato digital.
- *Almacenamiento y gestión de datos:* Esto cubre la estructura y organización de la información geográfica tanto en términos de la forma en es percibido por el usuario (modelo conceptual) como la forma en la cual es manejada por la computadora (modelo físico).
- *Transformación y Análisis de datos:* Esta funcionalidad consiste en los procesos de edición de la información para mantenerla actualizada o para remover errores. El análisis de datos es una de las principales tareas de los SIG, y se refiere a la

aplicación de los métodos de análisis a la información para lograr respuestas a las preguntas realizadas a los SIG.

- *Presentación y salida de datos:* La funcionalidad de producir mapas y material basado en mapas es una de las características distintivas de los SIG. La salida o presentación, se refiere a la forma en que se extraen y se muestran los datos, en forma de mapas, tablas, gráficos y reportes.

La funcionalidad de un componente se puede representar por medio de los servicios que ofrece. Se denomina servicio a una parte del software bien definida que ofrece por sí misma una funcionalidad de negocio y cuyo funcionamiento resulta independiente del estado y de las características del resto de servicios con los que se relaciona [58]. El alcance de funcionalidad expuesto por un componente define su granularidad, que puede ser de grano fino o de grano grueso [57].

Dentro de los componentes SIG de grano grueso se encuentran los sistemas más sofisticados y con una gran diversidad de posibilidades y aplicaciones, por ejemplo: ArcGis⁵, Idrisi⁶, Miramon⁷, Grass⁸. Por otro lado, los componentes SIG de grano fino tienen el objetivo de satisfacer necesidades específicas y particulares de los usuarios como serían los visualizadores de información que permiten únicamente observar y consultar la información cartográfica y alfanumérica almacenada por alguno de los programas estándar de SIG; por ejemplo: ArcReader⁹, Miramon Map Reader¹⁰.

2.2.1. Clasificación de Servicios Geográficos

El propósito de esta sección es hacer un resumen de los servicios que ofrecen los componentes relacionados a SIG para ilustrar la gran variedad y complejidad de los

⁵ www.esri.com/software/arcgis

⁶ www.clarklabs.org

⁷ www.creaf.uab.es/MiraMon

⁸ www.gras.itc.it

⁹ www.esri.com/software/arcgis/arcreader

¹⁰ www.creaf.uab.es/MiraMon/mmr

requerimientos funcionales que deben satisfacer las aplicaciones dentro del dominio de información geográfica.

En la literatura se puede encontrar distintos trabajos de investigación sobre la funcionalidad de los SIG. En el análisis realizado en [50] se identifican las tareas genéricas de alto nivel en los SIG: captura de datos, transferencia de datos, validación y edición de datos, almacenamiento y estructuración de datos, reestructuración de datos, generalización de datos, consulta de datos, análisis y presentación de datos.

Albrecht [2] presenta un conjunto de 20 operaciones GIS universales o tareas, independientes de la estructura de los datos subyacentes, las cuales cubren la totalidad de las capacidades analíticas ofrecidas por el software GIS del mercado actual. Las mismas surgieron a partir de un relevamiento realizado a usuarios para determinar sus expectativas respecto de la funcionalidad de un software GIS. Los resultados obtenidos revelaron un amplio rango de complejidad, desde operaciones elementales a tareas compuestas, por lo que se debió realizar un proceso de “normalización” de operaciones GIS. El conjunto de operaciones encontrado puede verse en Tabla 1.

Tabla 1. 20 Operaciones SIG Universales.

Categoría	Operaciones
Búsqueda	<ul style="list-style-type: none"> • Interpolación • Búsqueda temática • Búsqueda espacial • (Re-)clasificación
Análisis de ubicación	<ul style="list-style-type: none"> • Límites • Rutas • Solapamiento • Polígonos de Thiessen/Voronoi
Análisis de terreno	<ul style="list-style-type: none"> • Pendientes • Zona de influencia • Sistema de drenaje • Análisis de visibilidad
Distribución/vecindario	<ul style="list-style-type: none"> • Costo/Difusión/Propagación • Proximidad • Vecino más cercano
Análisis espacial	<ul style="list-style-type: none"> • Análisis multivariado • Patrón de dispersión • Centralización/Conexión • Forma
Mediciones	<ul style="list-style-type: none"> • Mediciones

Las operaciones de “Búsqueda”, si bien no son operaciones estrictamente analíticas, se dividen en búsquedas temáticas y búsquedas bajo restricciones geométricas. La “Reclasificación” es una operación de base de datos, pero el filtro que se usa para realizar la misma, tiene un determinante espacial.

El grupo “Análisis de Ubicación” está formado por cuatro operaciones que están entre las más conocidas y más usadas de todas las operaciones SIG. La operación “Solapamiento” está formada por muchas otras como “cortar”, “borrar”, “identidad”, “unión”, “intersección” que se pueden aplicar en distintas combinaciones. La operación “Polígonos de Thiessen/Voronoi” contiene la operación “Flujo-entre-regiones” (al asignar peso a los nodos Voronoi, es posible simular el flujo entre los polígonos Thiessen).

El grupo “Análisis de terreno” trata implícita o explícitamente datos en 3-dimensiones. La operación “Pendientes” requiere como entrada valores de alturas. La operación “Sistema de drenaje” calcula o el flujo desde una única ubicación o la red completa. La operación “Análisis de visibilidad” requiere entradas que contienen valores de alturas; también se debe designar un punto de vista y la altura del visualizador.

El grupo de operaciones “Distribución/vecindario” se usan para responder consultas no estadísticas sobre las relaciones entre features espaciales. La operación “Costo/Difusión/Propagación” toma una feature dimensionada arbitrariamente y calcula el valor de los atributos vecinos de acuerdo a una función propagación. La “Proximidad” es una operación que puede ser aplicada a todas las features de un archivo o solo a aquellas seleccionadas. La operación “Vecino más cercano” usa diferentes algoritmos que generalmente dependen de los datos de entrada.

Todas las mediciones estadísticas de cierta complejidad están categorizadas en el grupo “Análisis espacial”. Se incluyen las mediciones “Patrones de dispersión” (frecuencia, índice de similaridad, diversidad, fragmentación, densidad), las de “Centralización/Conexión”, las de “Forma” y el conjunto completo de herramientas para “Análisis multivariado”.

El grupo de “Mediciones” consiste de un número de simples cálculos geométricos (distancia, dirección, perímetro, altura, volumen) que son extendidos por cálculos

estadísticos más simples (número, histograma, media) y por algunas mediciones topológicas (adyacencia).

La Unión Europea a través de sus proyectos ESPRIT [19], también presenta una lista de operaciones clave para GIS. La confección de esta lista, está motivada por la idea de que usuarios mejor informados pueden tomar mejores decisiones respecto a sus requerimientos y que pueden describir mejor esas necesidades a sus proveedores GIS y a sus consultores durante las fases de selección e implementación. Para usuarios inexpertos en el dominio esto se dificulta, por lo que se propone lista de funciones u operaciones comunes mostradas en la Tabla 2.

La categoría “Parametrización del proyecto” comprende un conjunto de operaciones para inicializar el software SIG, la base de datos y los parámetros de visualización.

La categoría “Entrada de datos” está formada por cuatro operaciones. La operación “Digitalización de mapas”, que es el proceso de pasar la ubicación de features geométricas (puntos y líneas) de formato análogo (mapa de papel) a formato digital, creando una base de datos cartográfica de vectores. La operación “Escaneo de mapas” es el proceso de digitalizar features geométricas de un mapa en papel usando un escaner, produciendo una base de datos en formato raster (celular, bitmap). La “Carga por lotes” es la función por la cual una gran cantidad de datos son leídos desde una fuente externa. La “Asignación de atributos” es el etiquetado de cada característica (*feature*) geométrica en el conjunto de datos cartográficos digitales, generalmente realizado durante la digitalización de la *feature*.

La categoría “Conversión de datos” incluye la conversión de formato “raster a vector” de datos espaciales (generalmente derivados de un proceso de escaneo), y la conversión de “vector a raster” de *features* geométricas en una base de datos a una representación usando un *grid* raster de celdas o pixels regulares.

En la categoría “Validación de datos” se encuentra la operación “Identificación/corrección de errores topológicos”, que es un proceso cíclico que permite que los errores puedan ser visualizados, categorizados, y luego editados en un

proceso manual. Otra operación de esta categoría es la “Identificación y corrección de datos en tablas”, que se refiere a las operaciones (unión, división, nuevas relaciones) que es necesario realizar para obtener los datos según el formato o estructura deseados.

Tabla 2. Operaciones Clave de SIG

Categoría	Operaciones
Parametrización del proyecto	<ul style="list-style-type: none"> • Inicialización de software SIG • Inicialización motor base de datos • Ajuste parámetros visualización
Entrada de datos	<ul style="list-style-type: none"> • Digitalización de mapas • Escaneo de mapas • Carga por lotes • Asignación de atributos
Conversión de datos	<ul style="list-style-type: none"> • Raster a vector • Vector a raster
Validación de datos	<ul style="list-style-type: none"> • Identificación/corrección de errores topológicos • Identificación y corrección de datos en tablas
Visualización/renderización	<ul style="list-style-type: none"> • Zoom/Pan/cambiar vista • Refrescar pantalla • Simbolización de <i>features</i> • Filtrado de <i>features</i> • Gestión de imágenes de fondo
Gestión de base de datos de mapas	<ul style="list-style-type: none"> • Georeferenciamiento • Unión de hojas/mosaicos de mapas • Rectificación/fusión de capas • Cambio de proyección
Gestión de atributos de base de datos	<ul style="list-style-type: none"> • Enlace de atributo básico a base de datos principal • Definición de relaciones complejas • Definición de conexiones (SQL, middleware) a sistemas o bases de datos secundarios
Análisis/procesamiento de datos	<ul style="list-style-type: none"> • Creación/guardado de vistas • Análisis de proximidad • Superposición de mapas • Análisis de redes (rutas óptimas, asignación de recursos) • Análisis raster (álgebra de mapas) • Generalización/suavizado • Generación de vistas 3D – DTM (Modelado Digital de Terreno) • Análisis secundarios de DTM
Salida: producción de mapas/reportes	<ul style="list-style-type: none"> • Generación de estadísticas resumen • Generación de reportes de texto a partir de atributos en base de datos • Generación de mapas simples • Generación de composición de mapas

La categoría “Visualización/renderización” incluye “Zoom/Pan/cambiar vista”, mediante la cual es posible magnificar o reducir la escala de un mapa o imagen mostrada en la pantalla (*zoom*), y también cambiar la posición en la cuál la vista es

mostrada sin modificar la escala (*pan*). La operación “Refrescar pantalla” acompaña a la operación anterior, como un proceso automático, ya que la imagen de la pantalla debe ser re-dibujada después de cambios de escala o desplazamientos. La “Simbolización de *features*” es el proceso de asociar ciertas *features* geográficas con patrones o colores seleccionados (por ejemplo, las rutas pueden ser coloreados de acuerdo a su clase o densidad de tráfico. El “Filtrado de *features*” involucra la selección de elementos gráficos que el usuario desea visualizar en un momento dado. Y la “Gestión de imágenes de fondo” es el proceso de visualización y gestión sobre la misma vista de objetos vector (puntos, líneas, áreas) en el frente y la imagen raster georeferenciada en el fondo.

Entre las operaciones de la categoría “Gestión de base de datos de mapas” se incluye el “Georeferenciamiento”, que es el proceso de asociar ubicaciones conocidas en el mundo real con las correspondientes ubicaciones sobre el conjunto de datos cartográficos. La “Unión de hojas/mosaicos de mapas” permite la unión de mosaicos y la concatenación de archivos de datos de mapas. La “Rectificación/fusión de capas” ocurre cuando es necesario ajustar distintas capas de datos de una base de datos SIG que provienen de distintas fuentes o poseen diferentes escalas o sistemas de coordenadas. El “Cambio de proyección” permite que varios mapas de entrada puedan tener diferentes proyecciones; estas proyecciones pueden ser cambiadas para adecuar un proyecto particular o para ser consistente con otras capas de datos.

La categoría “Gestión de atributos de base de datos” se refiere a operaciones relacionadas con las bases de datos relacionales como el “Enlace de atributo básico a base de datos principal”, por medio del cual el número de ID único asociado con cada *feature* es conectado a la base de datos principal con el fin de proveer capacidades de consulta gráficas. La “Definición de relaciones complejas” es una operación que usa Sistemas de Gestión de Bases de Datos Relacionales (RDBMS) para definir relaciones complejas entre atributos de *features* claves. La “Definición de conexiones (SQL, middleware) a sistemas o bases de datos secundarios” incluye las operaciones ofrecidas por aplicaciones middleware para realizar traducciones entre las aplicaciones SIG y los sistemas existentes en una organización.

Las operaciones de creación de vistas, análisis de proximidad, superposición de mapas, análisis de redes, análisis raster, generalización o suavizado de líneas/polígonos, generación de vistas 3D y otras funciones SIG fundamentales pertenecen a la categoría “Análisis/procesamiento de datos”. Estas funciones difieren significativamente dependiendo de la estructura de datos (raster o vector).

Las operaciones de la categoría “Salida: producción de mapas/reportes” son aquellas que permiten: Generación de estadísticas resumen (cálculo de sumatoria, media, desviación estándar de atributos clave), Generación de reportes de texto a partir de atributos en base de datos (reportes en formato de texto obtenidos a partir de las tablas de atributos, Generación de mapas simples, y Generación de composición de mapas (combinación de capas de mapas, tablas, leyendas, etc. para producir una salida gráfica que incluye información desde distintas fuentes).

Se puede observar que ambas clasificaciones difieren en el alcance; mientras que la primera describe, desde el punto de vista del usuario, la funcionalidad de los SIG en términos de operaciones analíticas, la segunda no sólo tiene en cuenta lo anterior (con un nivel mayor de abstracción), sino que considera la funcionalidad del software en su totalidad.

2.2.2. Estándares Internacionales para SIG

Dado que cada aplicación tiene diferentes puntos de vista sobre la información geográfica, cada desarrollador define modelos conceptuales, modelos de datos geográficos, formatos de almacenamiento, operaciones de análisis o procedimientos de representación especialmente adaptados a los requerimientos de la aplicación. Como consecuencia de esto surge un problema de interoperabilidad entre las herramientas: no es fácil usar una herramienta SIG para analizar información recolectada con otra herramienta. Una documentación de alta calidad basada en estándares aplicables a un dominio debería facilitar el proceso de clasificación y modelado de la información.

Con el objetivo de tratar este problema, representantes del gobierno, de la industria e investigadores han conformado organizaciones como *International Organization for Standardization (ISO)* y *Open Geospatial Consortium (OGC)* dedicadas, entre otras cosas, a la creación de especificaciones de estándares para facilitar la interoperabilidad entre implementaciones creadas por distintos desarrolladores. Estas especificaciones tratan distintos tópicos asociados con la representación, manipulación y visualización de información geográfica (familia de estándares ISO 19100 [27], [28], [29], [30], [31]).

En particular el estándar ISO/IEC 19119 [30], relacionado con la semántica del procesamiento de información geográfica, define una taxonomía formada por las siguientes categorías de servicios geográficos y, además, presenta una lista de ejemplos para cada categoría:

- *Interacción humana*: Servicios para gestión de interfases de usuario, gráficos, multimedia y presentación de documentos compuestos.
- *Administración de información/modelo*: Servicios para gestión de desarrollo, manipulación y almacenamiento de metadatos, esquemas conceptuales y conjuntos de datos.
- *Gestión de tareas/workflow*: Servicios que soportan el uso de recursos y desarrollo de productos que involucran una secuencia de actividades o pasos que pueden ser conducidas por personas diferentes.
- *Procesamiento geográfico*: Servicios que realizan cálculos de gran escala e involucran una sustancial cantidad de datos.
- *Comunicación geográfica*: Servicios para codificación y transferencia de datos por medio de redes de comunicaciones.
- *Administración de sistemas geográficos*: Servicios para la gestión de componentes, aplicaciones y redes. Incluyen gestión de cuentas de usuarios y privilegios de acceso.

2.2.3. Repositorios de Componentes SIG

Cuando los componentes pueden ser producidos y consumidos dentro de una misma organización se lo conoce como mercado interno. En este caso los usuarios tienen acceso al código de los componentes para modificarlo y satisfacer sus requerimientos. En cambio cuando los desarrolladores y los usuarios pertenecen a distintas organizaciones, el mercado es externo. El componente se usa “como está” sin posibilidades de cambiar el código.

En un estudio sobre el estado del mercado de componentes reportado en [67] se muestra que el medio más apropiado para ofrecer, promover y distribuir componentes es Internet, donde los sitios web relacionados al mercado se pueden clasificar en:

- *Productor*: sitio de una organización que vende los componentes que ésta desarrolla.
- *Catálogo*: sitio de una organización que compila una lista de enlaces a componentes, basados en los sitios de los productores; los catálogos en realidad no venden componentes.
- *Intermediario*: sitio de una organización que vende componentes construidos por terceras partes.

De acuerdo a Szyperski [64] el principal ejemplo dentro del mercado de componentes en Internet es ComponentSource [12], un sitio donde se ofrecen más de 3.000 componentes de distintas categorías. Sin embargo, también existen otros portales web que mantienen catálogos de componentes OTS:

- El proyecto FreeGIS [25] promueve el uso y difusión de software libre para SIG. En el sitio web se proveen enlaces a más de 300 productos de software, geodatos, documentos y proyectos de acceso libre, relacionados con GIS y GPS.
- Freshmeat [26] es un sitio que mantiene una gran base de datos con información y enlaces de productos, preferentemente Open Source.
- ESRI [20] es una empresa de desarrollo de SIG, en su sitio comercializa alrededor de 80 productos de software OTS interoperables.

Cada uno de estos sitios web propone su propio esquema de clasificación y estructura de almacenamiento para describir los componentes. En la Tabla 3 se puede observar que no existe un criterio común para categorizar la información, ya que cada sitio describe los componentes según distinto nivel de detalle y desde diferentes puntos de vista.

Tabla 3. Información publicada en los catálogos web.

ComponentSource	FreeGIS	Freshmeat	ESRI
Nombre	Nombre	Nombre	Nombre
Versión	Versión	Versión	Versión
Precio	Homepage	Autor	Cómo comprar
Detalles del editor	Tipo de producto	Homepage	Contacto
Licencia	Aplicación	Estado de desarrollo	Información legal
Compatibilidad	Sistema operativo	Audiencia	Tipo de producto
Características	Leng. de programación	Licencia	Req. del sistema
Revisiones	Licencia	Entorno	Estándares
	Estándares	Sistema Operativo	Demostraciones
	Estado	Leng. de programación	Características clave
	Descripción	Traducción	
	Características	Dependencias	
		Descripción producto	
		Tópico	

Uno de los principales problemas que enfrentan los usuarios de repositorios es la búsqueda dentro de un conjunto numeroso de posibilidades. Un principio de solución a este problema, es dotar a los repositorios con mecanismos de selección automática como categorización, búsqueda por aspectos, y búsqueda de texto completo. Algunos autores [38], [77] sostienen que otro aspecto a considerar para mejorar el resultado de una búsqueda es el conocimiento que el usuario tenga de la estructura del repositorio de almacenamiento.

2.3. Uso de Técnicas Semánticas y del Lenguaje Natural en la Extracción de Información

Las tecnologías asociadas al Procesamiento de Lenguaje Natural y a la Web Semántica, pueden contribuir al mejoramiento en la publicación de la información sobre componentes para el dominio de SIG. Las técnicas de Extracción de Información, en

particular la *anotación semántica* permite enriquecer a las descripciones de componentes encontradas en los catálogos, y luego almacenarlas en un repositorio que facilite la búsqueda de los mejores componentes candidatos a satisfacer los requerimientos del desarrollador de aplicaciones.

2.3.1. Extracción de Información

El objetivo de un sistema de extracción de información (*Information Extraction*) es obtener información acerca de hechos muy específicos (como fechas, nombre propios, eventos, relaciones entre eventos y entidades) a partir de un texto en lenguaje natural acerca de un dominio de interés. La información obtenida como salida puede ser mostrada directamente a los usuarios, o bien puede ser almacenada en una base de datos.

Por otro lado, los sistemas de recuperación de información (*Information Retrieval*) son usados para obtener los artículos relevantes respecto a un tema dentro de un volumen muy grande de textos. Una vez que se tienen todos los artículos relevantes, el usuario puede necesitar identificar cierta información dentro de ellos. Para encontrar los hechos específicos de interés sin necesidad de leer cada artículo, es necesario usar un sistema de extracción de información. El proceso de extracción de información se puede realizar más fácilmente en clases de textos que tengan información muy específica, expresable en forma de tabla o plantilla.

La importancia de la extracción de información es determinada por el crecimiento de la cantidad de información no estructurada (sin metadatos) disponible en la web. Una aplicación típica de extracción de información consiste en analizar un conjunto de documentos escritos en lenguaje natural y poblar una base de datos con la información extraída. Los enfoques actuales para extracción de información usan técnicas de procesamiento de lenguaje natural focalizadas en dominios muy restringidos. El desarrollo de sistemas de extracción de información fue fomentado por la competencia

entre diferentes grupos de investigación durante las MUC¹¹ (*Message Understanding Conference*). Dentro de éstas, Hobbs [40] definió lo que se considera la arquitectura general para construir sistemas de extracción de información. En la Figura 4 se presenta esta arquitectura genérica como una serie de módulos en cascada, en los cuales se agrega estructura al documento, o se filtra información relevante por medio de la aplicación de reglas.

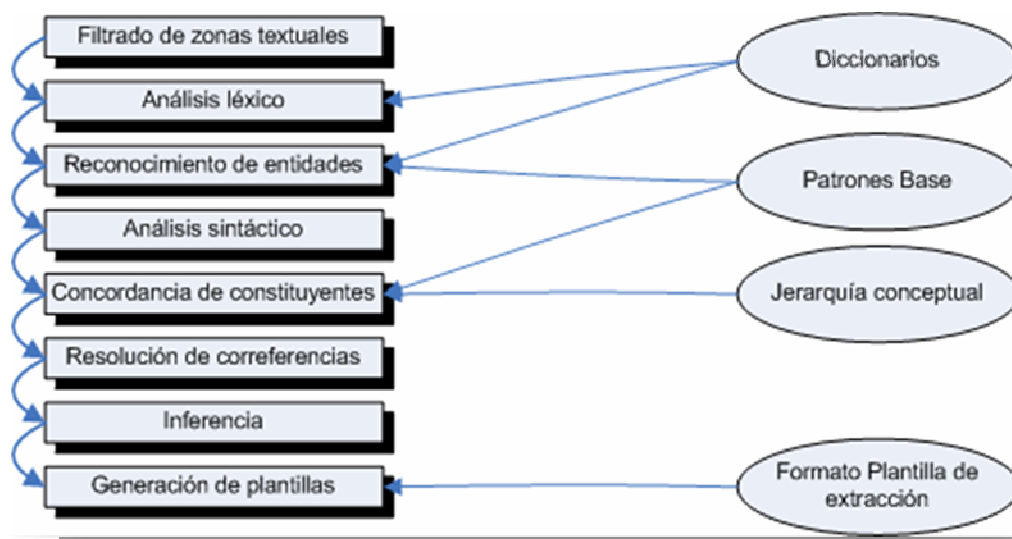


Figura 4. Arquitectura genérica de un sistema de Extracción de Información.

Dependiendo de los requerimientos del sistema de extracción de información, es posible combinar los módulos que componen la arquitectura para realizar alguna de las cuatro funciones genéricas siguientes:

1. En el *pre-procesamiento de los documentos*, se pueden aplicar los módulos:
 - *División en zonas textuales*. Segmentación de un texto en zonas, por ejemplo en párrafos.

¹¹ <http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html>

-
- *Segmentación del texto.* Transformación de las zonas en segmentos apropiados, usualmente en oraciones.
 - *Filtrado de texto.* Selección de los segmentos relevantes y eliminación de información irrelevante como ciertas etiquetas o marcas de formato.
 - *Tokenización.* Obtención de las unidades léxicas (i.e. tokens) en los segmentos, generalmente son palabras.
 - *Analizador léxico.* Análisis morfológico de los tokens (por ejemplo: género y número), y reconocimiento y clasificación de entidades.
 - *Desambiguación.* Tratamiento de palabras que desempeñan diferentes categorías sintácticas y palabras polisémicas que en función del contexto pueden tener un sentido u otro. Se usa *etiquetado de las partes de la oración* y *desambiguación del sentido de las palabras*.
 - *Obtención del lema y truncado.* Aplicación de las reglas de formación de las palabras para proporcionar su lema o raíz.
2. En el *análisis sintáctico* y *la interpretación semántica* se trata de identificar la forma en que las palabras se combinan para formar constituyentes a nivel sintáctico superior (i.e. los sintagmas); y posteriormente, se genera una forma lógica o una plantilla parcial. Los módulos aplicados para este propósito son:
- *Análisis sintáctico completo.* Se encarga de decidir si una oración es gramaticalmente correcta.
 - *Análisis sintáctico parcial.* Tiene como objetivo recuperar información sintáctica de forma eficiente y fiable, desde texto no restringido. Este tipo de análisis sacrifica la completitud y profundidad del análisis completo. Sólo los conceptos que forman parte del escenario de extracción son relevantes para ser detectados en el documento, por lo tanto el análisis sintáctico y semántico debe simplificarse a un análisis de frases.
 - *Concordancia de patrones.* Con los fragmentos del texto etiquetados sintácticamente (llamados *frases constituyentes*), el sistema puede determinar
-

dependencias entre ellos específicas del dominio de extracción. Se aplica un emparejamiento de patrones específicos del dominio (conocidos como *patrones de extracción*) para identificar la información relevante así como las dependencias entre los constituyentes.

- *Relaciones gramaticales*. El objetivo es determinar las dependencias entre constituyentes utilizando un modelo sintáctico más flexible. En el modelo se define un conjunto de relaciones gramaticales entre las entidades, que generalizan aspectos como tiempo, lugar, sujeto, objeto, entre otros. El modelo se representa con un grafo donde los nodos identifican las partes de los constituyentes y los arcos etiquetados las relaciones entre éstos.
3. En el *análisis del discurso*, se intenta resolver aspectos semánticos como la elipsis (i.e., omitir en la oración una o más palabras) y la anáfora (i.e., asumir el significado de una parte del discurso ya emitida). Para esto, los sistemas de extracción de información representan la información extraída como:
- *Plantillas llenadas parcialmente* y que posteriormente usan algún procedimiento de fusión.
 - *Formas lógicas* para usar procedimientos tradicionales de interpretación semántica.
4. En la *generación de plantillas de salida*, se realiza el enlace de las piezas de información extraídas con el formato de salida deseado. En esta fase puede requerirse algún tipo de inferencia.

2.3.2. Reconocimiento de Entidades Nombradas

El término entidad nombrada, en inglés *Named Entity* (NE), ha sido acuñado en las conferencias MUC¹² (*Message Understanding Conference*), y se la define como las frases que son identificadores únicos de entidades (organizaciones, personas y

¹² <http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html>

localidades), expresiones temporales (fechas o expresiones de tiempo como puede ser “mediodía”) y expresiones numéricas (porcentajes o cantidades monetarias).

La tarea de reconocimiento de entidades nombradas, en inglés *Named Entity Recognition* (NER), consiste en dadas unas categorías predefinidas de entidades nombradas de interés, localizar de forma automática en un texto dado todas las palabras que sean instancias de dichas categorías. Por otro lado, una vez detectadas las entidades nombradas se entiende por clasificación de las mismas el proceso de otorgar a cada entidad nombrada una determinada categoría semántica (ej: Diego Maradona es de tipo Persona).

El reconocimiento y clasificación de entidades nombradas puede servir tanto para poblar ontologías de determinados dominios como para aportar información a sistemas de pregunta-respuesta o sistemas de extracción de información. En el caso de los sistemas de Extracción de Información, la información provista es útil para correferencia y rellenado de plantillas [56].

Las aplicaciones que usan técnicas de Reconocimiento de Entidades Nombradas se basan en un conjunto de patrones lingüísticos orientados a identificar las entidades relevantes en el texto. Generalmente, están asociados a listas de *gazetteer* que contienen las entidades (o parte de ellas) combinadas con patrones lingüísticos. Sin embargo, aunque la identificación de conceptos y/o relaciones ontológicas se puede lograr usando los términos conocidos (por medio de un *gazetteer* o por acceso a etiquetas de una ontología), esto podría no ser suficiente por una variedad de razones:

- el concepto puede no estar aún en la ontología;
- el concepto puede existir en la ontología sólo como un sinónimo o una variación lingüística (singular en lugar de plural, por ejemplo);
- el concepto puede ser ambiguo;
- puede existir solo una superclase del concepto en la ontología.

Para superar estas limitaciones es necesario enfocarse más en el uso de patrones lingüísticos que en el simple uso de un *gazetteer*.

2.3.3. Descubrimiento de Patrones

El descubrimiento de patrones es una de las partes principales de la extracción de información. Es la tarea de identificar patrones de extracción que son relevantes para las necesidades del usuario, especificadas por las consultas del usuario. Los patrones se pueden descubrir *automáticamente*, *semi-automáticamente* y *manualmente*.

Los primeros sistemas de Extracción de Información eran adaptados *manualmente* para una tarea dada [3], [39]. A partir del proyecto Proteus [76] se desarrolla la creación de patrones *semi-automática*, en el cual un procedimiento de descubrimiento de patrones ayuda al usuario a crear las bases de conocimiento, tales como *lexicons* y patrones. La metodología incluye mostrar un patrón candidato e incorporar una ontología existente y un conjunto de patrones. Los enfoques *automáticos* toman un conjunto de documentos y producen un conjunto de patrones de extracción usando técnicas de Aprendizaje de Máquina.

Para extraer hechos y eventos, el sistema de Extracción de Información necesita patrones de extracción específicos del dominio. El diseño de estas reglas de extracción de patrones relevantes al dominio, se caracteriza por dos enfoques [45]:

1. **Enfoque molecular.** Es el más común. Incluye la coincidencia de todos o de la mayoría de los argumentos de un evento en un único patrón. El ciclo de desarrollo comienza con un pequeño número de reglas altamente confiables que capturan los casos más comunes del dominio, pero ignoran una amplia clase de patrones relevantes que ocurren menos frecuentemente. Este sistema comienza con alta precisión y baja cobertura, y evoluciona incrementando la cobertura y disminuyendo la precisión.

-
2. **Enfoque atómico.** Construye un módulo de dominio que reconoce los argumentos para un evento y los combina en estructuras de plantilla estrictamente sobre la base de conjeturas inteligentes más que en relaciones sintácticas. El ciclo de desarrollo se caracteriza por asumir eventos relevantes al dominio para cualquier entidad reconocida, conduciendo a una alta cobertura pero mucha sobre-generación, y por tanto, baja precisión.

Este enfoque tiene sentido cuando (1) las entidades del dominio tienen tipos determinados fácilmente y (2) las plantillas están estructuradas de modo que hay uno o un pequeño número de posibles *slots* (atributos) que una entidad de un tipo dado puede completar y sólo las entidades de un tipo dado pueden completar esos *slots*.

2.3.4. La Web Semántica

La Web semántica (del inglés *semantic web*) es la representación abstracta de los datos de la web añadiendo metadatos semánticos y ontológicos que describen el contenido, el significado y la relación de esos datos. En [4] se define “la Web semántica es una extensión de la web actual en la cual a la información se le da un significado bien definido, mejorando el trabajo en cooperación entre humanos y computadoras”. La Web Semántica es un área que propone nuevas técnicas y paradigmas basadas en la introducción de conocimiento semántico explícito que describa y/o estructure la información y servicios disponibles, de forma legible para ser procesada automáticamente por un programa.

Tim Berners-Lee [5] define una arquitectura en cinco capas para la web semántica, que se muestra la Figura 5.

La primera capa está formada por **Unicode**, un estándar de codificación de caracteres que permite que todos los lenguajes puedan ser usados (escritos y leídos) en la web usando una forma estandarizada; y por **URI** (Uniform Resource Identifier), una cadena de forma estandarizada que permite identificar de forma inequívoca cualquier recurso

en la Web. Un subconjunto muy conocido de éstos, son los URL (Uniform Resource Locators) que se usan a diario en la distinción de direcciones electrónicas en la Web.

La capa sintáctica asegura el uso de una sintaxis común en la web semántica. XML es un lenguaje de marcado para documentos que contienen información estructurada. XML *namespaces* permite especificar diferentes vocabularios de marcado en un documento XML y XML *schema* sirve para expresar el esquema de un conjunto particular de documentos XML.

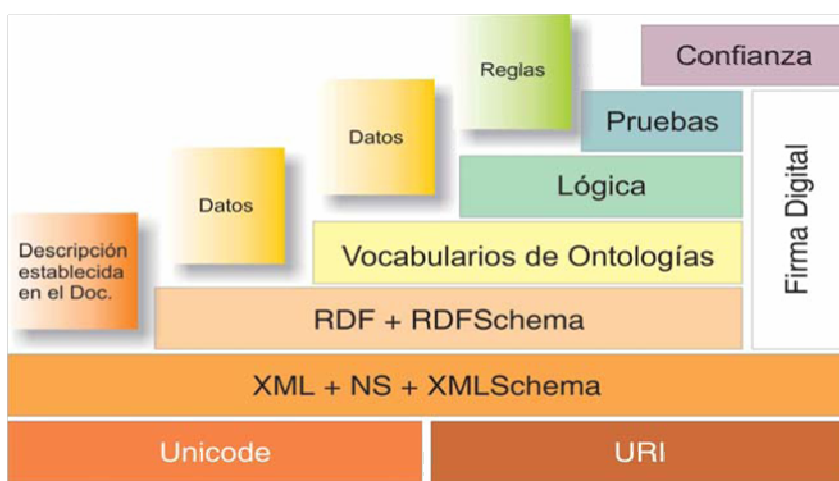


Figura 5. Modelo de capas propuesto por Berners-Lee para la web semántica.

En la capa de datos se encuentra RDF (*Resource Description Framework*), un framework para representar información sobre recursos de la web semántica en forma gráfica. Está basado en tripletas sujeto-predicado-objeto que forman el grafo de datos. Para permitir la descripción de taxonomías y otras construcciones ontológicas, se creó RDFS (*RDF Schema*). RDFS se puede usar para describir taxonomías de clases y propiedades y usarlas para crear ontologías livianas.

La tercera capa es la de ontologías. Según se define en [5], una ontología es “una especificación explícita y formal de una conceptualización compartida”, que se puede usar para describir fuentes de información distribuidas y heterogéneamente

estructuradas como las que se encuentran en la web. Para crear ontologías más detalladas se puede usar OWL (*Web Ontology Language*) [55], un lenguaje derivado de descripciones lógicas que ofrece más constructores que RDFS. Se presenta en tres formas: OWL *Lite* para taxonomías y restricciones simples, OWL DL para soporte de lógica de descripción completa, y OWL *Full* para la máxima expresividad y libertad sintáctica de RDF. También se ha definido una semántica formal para este lenguaje.

Finalmente, la capa lógica permite realizar consultas e inferir conocimiento y la capa de prueba permite la explicación de respuestas dadas generadas por agentes automatizados.

2.3.5. Ontologías

Una ontología está formada por una taxonomía relacional de conceptos y por un conjunto de axiomas o reglas mediante los cuales se podrá inferir conocimiento que no está explicitado en la taxonomía. Los componentes que las ontologías utilizan para representar conocimiento de algún dominio [33] son:

- *Conceptos*: ideas básicas que se intentan formalizar. Pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- *Relaciones*: representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- *Funciones*: tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo: categorizar-clase, asignar-fecha, etc.
- *Instancias*: representan objetos determinados de un concepto.
- *Axiomas*: teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son de la clase C, entonces A no es subclase de B”, “Para todo A que cumpla la condición C₁, A es B”, etc.

La comunidad ontológica distingue un tipo de ontologías, que son principalmente taxonomías, de las que modelan un dominio en una forma más ajustada y proveen más restricciones al dominio semántico. Se las suele distinguir como ontologías livianas y pesadas respectivamente [13].

- Ontología liviana: incluye conceptos, taxonomía de conceptos, relaciones entre conceptos y propiedades que describen conceptos.
- Ontología pesada: se agregan axiomas y restricciones a la anterior.

Las ontologías pueden ser clasificadas de acuerdo con el nivel de generalidad de la conceptualización especificada. Como generalidad se entiende la amplitud de la ontología; algunas ontologías intentan capturar todos los términos en lenguaje natural, mientras otras son muy específicas en ciertos dominios o términos. De acuerdo con Guarino [34], las ontologías se clasifican en:

- Ontología de alto nivel (*Top-level Ontology*). Aborda conceptos generales (e.g. entidad, objeto, idea, artefacto, etc.) los cuales son independientes de un dominio. Ejemplo: Ontología CyC y WordNet.
- Ontología de Dominio (*Domain ontology*). Enfocadas a cubrir alguna terminología sobre un dominio genérico como matemáticas, medicina. Ejemplo, la ontología de Ontolingua para modelado matemático.
- Ontología de Tarea (*Task ontology*). Se enfoca en describir el vocabulario sobre una tarea específica, por ejemplo comercialización, reparación, etc.
- Ontología de Aplicación (*Application ontology*). Este tipo de ontología describe conceptos de un dominio y de tareas particulares, comúnmente especializaciones de ambos.

2.3.6. Anotación Semántica

La Semántica Ontológica [52] estudia el significado del lenguaje natural, así como una aproximación al Procesamiento del Lenguaje Natural que utiliza un modelo abstracto

del mundo (la ontología) como recurso central para extraer y representar el significado de textos en lenguaje natural, al razonar con el conocimiento que se deriva a partir de estos textos. Asimismo, la ontología es también el eje central a la hora de generar textos en lenguaje natural basados en las representaciones de su significado.

Para que el contenido de los recursos disponibles en la Web actual tenga una descripción formal, procesable por las computadoras y sea posible alcanzar los objetivos de la Web Semántica, es necesario agregar metadatos semánticos a esos recursos. El proceso de agregar metadatos semánticos a los recursos Web se conoce como *anotación semántica*. La anotación semántica de un recurso Web (página Web, imagen, correo electrónico, etc.) requiere relacionar su contenido o una parte de él (por ejemplo, una palabra concreta) con cierto identificador (generalmente una URI), que determina inequívocamente un concepto que se menciona en el contenido del recurso. Estos identificadores son obtenidos de conceptualizaciones de ciertos dominios de conocimiento, que generalmente toman la forma de ontologías.

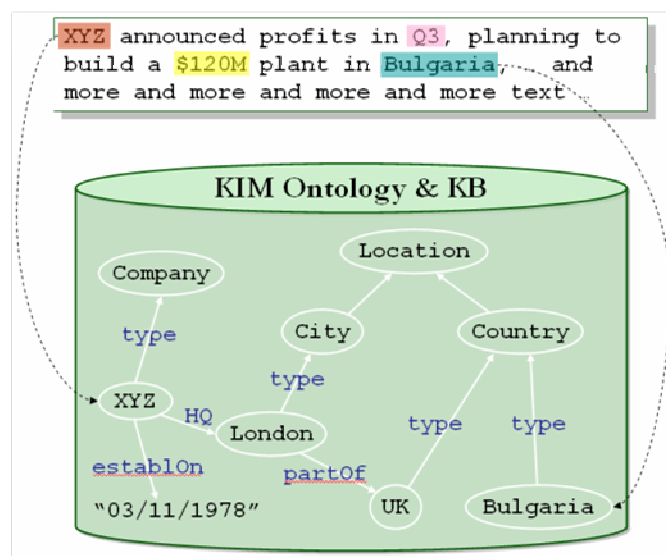


Figura 6. Anotación semántica.

En [47], se considera a la *anotación semántica* como la asignación de enlaces entre las entidades en un texto y sus descripciones semánticas (Figura 6). Esta clase de metadatos

proporciona tanto información de clases como de instancias de las entidades. La anotación semántica automática se puede considerar como una forma de *reconocimiento de entidades nombradas*, mejorada con la identificación y anotación con respecto a una ontología y una base de conocimiento de descripciones de las entidades.

Para añadir contenido semántico a las páginas web, existen diferentes herramientas de anotación. Estas se clasifican considerando el grado de automatización de las tareas de anotación o la clase de recursos web que pueden ser anotados. Según el grado de automatización, se pueden clasificar en sistemas de anotación manuales y (semi)automáticos.

2.3.6.1. Sistemas de Anotación Manuales

Estos sistemas proporcionan típicamente un interfaz de usuario que permite a los anotadores humanos ver y navegar simultáneamente las ontologías y los recursos Web, usando el conocimiento modelado en las ontologías para agregar anotaciones a recursos Web. Son ejemplos:

- Annotea [44]: En este proyecto se construyó una infraestructura basada en RDF, donde las anotaciones fueran una posible imagen de esa infraestructura. Las anotaciones que se manejan son colecciones de afirmaciones sobre un documento. Pueden ser comentarios, correcciones tipográficas, hipótesis o puntuaciones. Los metadatos de la anotación, descritos como esquema RDF especializado, dan información como la fecha de creación de la anotación, el nombre del autor, el tipo de anotación, la URI del documento anotado, y un puntero a la parte del documento que fue anotada. Los metadatos incluyen también la URI del cuerpo de la anotación, que se asume que es un documento XHTML. Los metadatos no dicen nada sobre cómo se presenta la anotación al usuario; esta es una decisión que se deja al diseñador de la aplicación.
- CREAM (*Creating RELational, Annotation-based Metadata*) [35]: Es un framework para un entorno de anotación que permite construir metadatos relacionales:

metadatos que comprenden instancias de clases e instancias de relaciones. Estas instancias no están basadas en una estructura fija, sino en una ontología de dominio.

- SMORE¹³: Herramienta que permite a los usuarios etiquetar sus documentos en RDF utilizando ontologías Web asociadas a los elementos y términos específicos. Ofrece al usuario un entorno flexible en el cual etiquetar los documentos con mínimo conocimiento de términos y sintaxis de RDF. El usuario debe clasificar semánticamente su conjunto de datos para la anotación (dividir las oraciones en el modelo básico sujeto-predicado-objeto).

2.3.6.2. Sistemas de Anotación (semi)Automáticos

Estos sistemas explotan técnicas de procesamiento de lenguaje natural para extraer las referencias en el texto a ciertos conceptos descritos en ontologías. Generalmente requieren como entrada patrones o corpus de documentos utilizados para entrenar el sistema. En relación con la clase de recursos que pueden ser anotados, la mayoría de los enfoques existentes, están dedicados a la anotación de recursos con contenido textual. Por ejemplo:

- MnM [70] es una herramienta que permite la anotación de documentos HTML con los elementos de una ontología cargada previamente en la aplicación. Incorpora un navegador que permite visitar las distintas páginas Web. La anotación se realiza seleccionando una cadena de texto de la página Web visualizada en el navegador, así como el elemento ontológico correspondiente. Esta acción coloca un par de etiquetas XML de inicio y cierre alrededor de la cadena seleccionada.
- AeroDAML [48] usa un enfoque basado en patrones y está diseñada para mapear sustantivos propios y relaciones comunes a las correspondientes clases y propiedades de las ontologías DAML (*DARPA Agent Markup Language*)¹⁴. La ontología consiste de dos niveles: el nivel superior usa la jerarquía WordNet [22], y el nivel inferior usa la Base de conocimiento provista por AeroText. El sistema AeroText integrado está formado por: 1) el compilador de la Base de Conocimiento

¹³ <http://www.mindswap.org/2005/SMORE/>

¹⁴ <http://www.daml.org/>

(KB) para transformar los datos lingüísticos en una Base de Conocimiento ejecutable; 2) la Máquina de la Base de conocimiento para aplicar la Base de conocimiento a los documentos fuente; 3) un entorno de desarrollo IDE (*Integrated Development Environment*) para construir y probar las Bases de conocimiento, y 4) la Base de conocimiento Común que contiene las reglas independientes del dominio para extraer nombres propios y relaciones. AeroDAML está disponible en dos versiones: basado en web y cliente-servidor.

- Armadillo [17] usa el sistema de extracción de información Amilcare¹⁵ para realizar minería en sitios web que tengan una estructura altamente regular. Usa un enfoque basado en patrones para encontrar las entidades. Los patrones manuales se usan para el reconocedor de entidades nombradas. Los documentos del corpus no requieren anotación manual.
- KIM (*Knowledge and Information Management*) [59] contiene una ontología, una base de conocimiento, una anotación semántica, un servidor de recuperación e indexación y una interfase con el servidor. Para el almacenamiento de la ontología y la base de conocimiento usa el repositorio SESAME RDF [8] y para la búsqueda usa una versión modificada de la máquina de búsqueda Lucene¹⁶. El proceso de anotación está basado en una ontología liviana pre-construida (KIMO) y en una base de conocimiento inter-dominios. KIMO define un conjunto base de clases de entidades, relaciones y restricciones de atributos. Las entidades nombradas encontradas durante el proceso de anotación son emparejadas a su tipo en la ontología y también a una referencia en la base de conocimiento. El componente de extracción de información de la anotación semántica se realiza usando componentes de GATE [15].
- Ont-O-Mat [36] es una implementación del framework de anotación semántica SCREAM (*Semiautomatic CREAtion of Metadata*). El componente de extracción de información está basado en Amilcare, un sistema basado en aprendizaje automático que requiere de un corpus de entrenamiento de documentos anotados manualmente. Amilcare usa ANNIE (*A Nearly-New IE system*), una parte de GATE para realizar la extracción de información.

¹⁵ nlp.shef.ac.uk/amilcare/

¹⁶ <http://jakarta.apache.org/lucene>

-
- SemTag [16] es el componente de anotación semántica de la plataforma Seeker, para realizar anotaciones de páginas web a gran escala. Realiza la anotación en tres fases: Señalamiento (*Spotting*), Aprendizaje, y Marcación.

2.3.7. Extracción de Información Basada en Ontologías

Recientemente ha surgido un sub-campo de la Extracción de Información, conocido como Extracción de Información Basado en Ontologías (OBIE, del inglés *Ontology-Based Information Extraction*). Las ontologías se usan para el proceso de extracción de información y también para presentar los resultados. Por tratarse de un campo relativamente nuevo, no existe en la literatura una definición totalmente aceptada para este término; sin embargo, en el estudio presentado en [73], se provee la siguiente definición: “un sistema de extracción de información basado en ontologías es un sistema que procesa lenguaje natural no estructurado o semi-estructurado a través de un mecanismo guiado por ontologías para extraer cierto tipo de información y presentar el resultado usando ontologías”.

Los factores clave que caracterizan a los sistemas OBIE y los distinguen de otros sistemas de Extracción de Información son:

- Procesan texto en lenguaje natural semi-estructurado (páginas web que usan plantillas particulares, como Wikipedia) o no estructurado (archivos de texto). Sistemas que usan como entrada imágenes, diagramas o videos no se pueden categorizar como sistemas OBIE.
- Presentan la salida usando ontologías. Generalmente, los sistemas OBIE usan ontologías tanto como entrada del sistema como salida resultante. Algunos sistemas, en lugar de tratar a la ontología como entrada, construyen la ontología a ser usada por medio de procesos de Extracción de Información.
- Usan un proceso de Extracción de Información guiado por una ontología. La ontología guía el proceso para extraer elementos tales como clases, propiedades e instancias.

La Extracción de Información basada en ontologías provee mecanismos automáticos para generar contenidos semánticos al convertir la información contenida en páginas web existentes en ontologías. La diferencia entre los sistemas OBIE y los basados en anotaciones semánticas está en que los primeros presentan la información semántica resultante en la ontología, y los últimos lo hacen utilizando metadatos semánticos.

En el siguiente capítulo presentaremos nuestra propuesta para la normalización y publicación de descripciones de componentes SIG. Para esto hemos diseñado un Esquema de Clasificación Normalizado (de aspectos funcionales y no técnicos) basado en la adaptación del Framework USCS [54], un modelo de especificación de componentes flexible. Con el objetivo de uniformizar las categorías de clasificación, analizamos información disponible en catálogos web de componentes SIG; y para expresar la funcionalidad de los componentes en términos de categorías estandarizadas propias del dominio SIG, adecuamos la taxonomía de servicios geográficos del estándar ISO/IEC 19119 [30].

También desarrollamos una herramienta que, por medio de la aplicación de técnicas de Extracción de Información, permite poblar automáticamente el Esquema de Clasificación Normalizado de componentes SIG con la información disponible en portales Web especializados. Para esto, proponemos enriquecer con anotaciones semánticas a las descripciones de componentes encontradas en los catálogos, y luego almacenarlas en un repositorio que facilite la búsqueda de los mejores componentes candidatos a satisfacer los requerimientos del desarrollador de aplicaciones.

3. NORMALIZACIÓN Y PUBLICACION DE DESCRIPCIONES DE COMPONENTES SIG

Uno de los elementos principales de nuestra propuesta es orientar la clasificación de componentes a un dominio con el fin de proveer mayor especificidad. En el dominio seleccionado (SIG) existe una taxonomía de servicios geográficos estándar [30] que puede incorporarse a la clasificación propuesta en el Framework USCS [53] de manera de proveer un esquema de clasificación que pueda ser entendido y utilizado por cualquier seleccionador de componentes OTS geográficos.

3.1. Normalización de Descripciones de Componentes SIG

En esta Tesis proponemos un esquema de clasificación basado en la adecuación del Framework USCS a la clasificación de componentes SIG.

Para conocer la información disponible, compararla y poder determinar posibles categorías de clasificación de componentes SIG en el marco del Framework USCS, relevamos la información publicada en sitios web que mantuvieran catálogos con componentes SIG:

1. ComponentSource¹⁷ es un sitio web para la adquisición de componentes. En el sitio se ofrecen más de 3.000 componentes de distintas categorías, en la categoría “*Mapping & GIS Components*” se ofrecen 27 componentes SIG.
2. El proyecto FreeGIS¹⁸ promueve el uso y difusión de software libre para SIG. En el sitio web se proveen enlaces a más de 350 productos de software, geo-datos, documentos y proyectos de acceso libre, relacionados con GIS y GPS.
3. Freshmeat¹⁹ es un sitio que mantiene una gran base de datos de productos, preferentemente Open Source. Se puede encontrar información y enlaces a 150 productos etiquetados en la categoría “*Geographical*”.

¹⁷ <http://www.componentsource.com/> (Ultimo acceso diciembre/2010)

¹⁸ <http://freegis.org/> (Ultimo acceso diciembre/2010)

4. ESRI²⁰ es una empresa de desarrollo de SIG, en su sitio comercializa alrededor de 80 productos de software OTS interoperables.

En la Tabla 4 se presenta el resultado del relevamiento de la información disponible en los sitios web antes mencionados; en la misma se puede observar que no existen criterios uniformes para categorizar los componentes y además, que la información relacionada con algunos grupos definidos en el USCS (*Interfaces e Información extrafuncional*) no se encuentra accesible en forma directa (no sería posible la búsqueda automática de esta información).

Tabla 4. Información publicada en catálogos web y el Framework USCS.

	ComponentSource	FreeGIS	Freshmeat	ESRI
Información general	Nombre Versión Precio Detalles del editor	Nombre Versión Homepage	Nombre Versión Autor Homepage	Nombre Versión Cómo comprar Contacto
Clasificación	Licencia Compatibilidad	Tipo de producto Aplicación Sistema operativo Leng. programación Licencia Estándares Estado	Estado de desarrollo Audiencia Licencia Entorno Sistema Operativo Leng. programación Traducción Dependencias	Inf. legal Tipo producto Req. de sistema Estándares
Funcionalidad	Características Revisiones	Descripción Características	Descripción producto Tópico	Demostraciones Caract. clave
Interfaces	INAD*	INAD*	INAD*	INAD*
Inf. extrafuncional	INAD*	INAD*	INAD*	INAD*

(*INAD: Información No Accesible directamente)

¹⁹ <http://freshmeat.net/> (Ultimo acceso diciembre/2010)

²⁰ <http://www.esri.com/> (Ultimo acceso diciembre/2010)

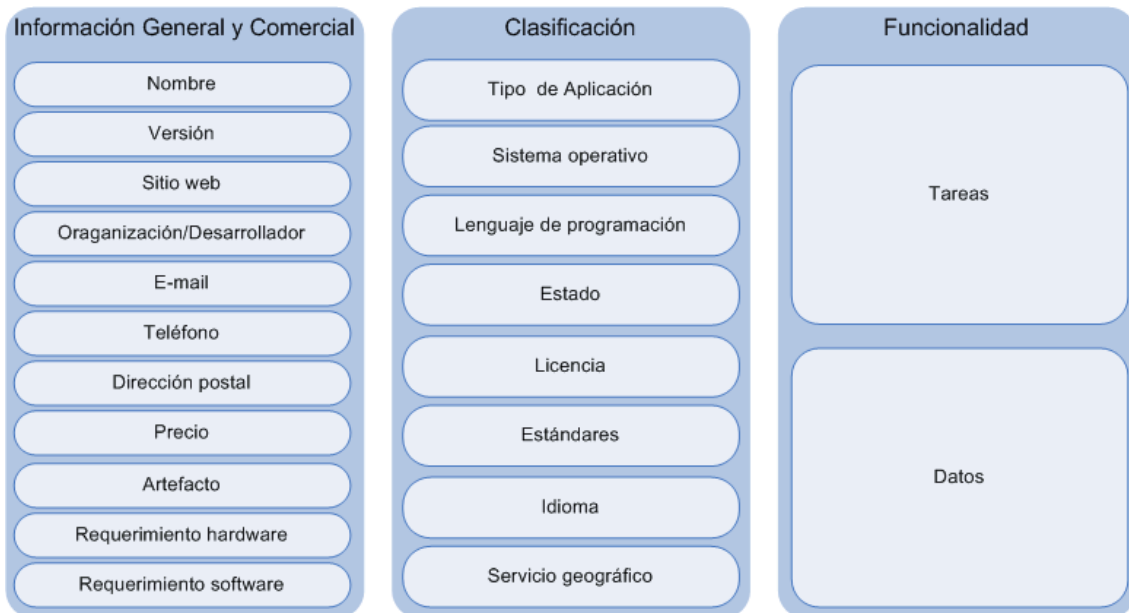
Como resultado de la comparación y composición de la información publicada en los sitios web descriptos y de la adaptación de la taxonomía de servicios geográficos ISO/IEC 19119, se elaboró un esquema organizado en 21 categorías distribuidas en tres grupos correspondientes con la clasificación propuesta en el Framework USCS: Información general y comercial, Clasificación y Funcionalidad. El Esquema de Clasificación normalizado para categorización de componentes SIG presentado en la Tabla 5, se construyó teniendo en cuenta las siguientes consideraciones:

- Las categorías pertenecientes al grupo *Información general y comercial*, se determinaron como resultado de la consolidación de información presente en los catálogos web analizados.
- Las categorías pertenecientes al grupo *Clasificación*, surgieron como resultado de la consolidación de categorías presentes en los catálogos web analizados. A excepción de la categoría *Servicios geográficos*, que surge de la adaptación de la Taxonomía de servicios geográficos del estándar ISO/IEC 19119.
- El grupo *Funcionalidad* incluye la categoría *Tareas*, que precisa la funcionalidad que brinda el servicio geográfico, y la categoría *Datos*, que registra términos propios del dominio, útiles para describir el componente SIG.
- Para determinar las categorías adecuadas en los grupos *Interfaces* e *Información extrafuncional*, no se encontró este tipo de información disponible (de acceso directo) en los catálogos web analizados. Por esta razón, el presente trabajo sólo considerará los grupos *Información general y comercial*, *Clasificación* y *Funcionalidad*.

Ya que la información de componentes disponible en los catálogos web se describe generalmente en términos del lenguaje natural, la búsqueda e identificación automática se hace particularmente compleja. En nuestro trabajo validamos de forma manual, con la ayuda de un experto en el dominio, la aplicación de este Esquema de clasificación para describir componentes disponibles en catálogos web, probando ser útil tanto en su

definición como en su uso. Ejemplos de esta instanciación manual de las categorías del Esquema de Clasificación se muestran en la Tabla 6.

Tabla 5. Esquema de Clasificación normalizado



En una siguiente etapa, y como otra manera de validar, se realizó una evaluación de la cantidad de información efectivamente publicada que permita completar el Esquema de clasificación normalizado antes descrito. Para esto, exploramos la información de 50 componentes publicados entre los distintos catálogos web de componentes detallados.

En la Tabla 7 se muestra el resultado de la exploración, indicando el porcentaje de componentes que disponen información (accesible directamente en los sitios web explorados) para completar las categorías del Esquema de Clasificación normalizado. Tal como se reporta en [6], comprobamos que existen diferencias entre la información requerida para evaluar componentes y la efectivamente disponible.

Tabla 6. Ejemplos de clasificación de componentes SIG

Categoría	Componente 1	Componente 2	Componente 3
Nombre	MM Contours ActiveX	ArcPad	UDIG
Versión	2.3	7.1.1	N/D*
Sitio web	http://www.componentsource.com/products/mm-contours-activex-control/index.html	http://www.esri.com/software/arcgis/arcpad/index.html	http://udig.refractions.net/
Organización / desarrollador	MapMakers Group	ESRI	Jody Garnett
e-mail	info@componentsource.com	profservices@esri.com	jgarnett@refractions.net
Teléfono	893 850 9911	800-447-9778.	N/D*
Dirección postal	655 Claremore Professional Way. Woodstock. GA 30188-5100, USA	380 New York Street Redlands, CA 92373-8100	N/D*
Precio	US\$ 293.02	US\$495.00	N/A*
Artefacto	Ejecutable, Ayuda,	N/D*	Código Fuente
Requerimientos de software	N/D*	ArcPad Data Manager , ArcGIS Desktop	GeoTools
Requerimientos de hardware	Disco: 6MB, RAM: 16MB	Disco 100MB, RAM:256MB	N/D*
Tipo de aplicación	GIS de escritorio	GIS móvil	GIS Web
Sistema operativo	Windows	Windows	Linux, Windows, Mac OS X
Lenguaje de programación	N/A*	N/A*	Java
Estado	N/A*	N/A*	Estable
Licencia	1 desarrollador	1 desarrollador	GNU LGPL
Estándares	N/D*	N/D*	WMS, WFS
Idioma	N/D*	Chino, Alemán, Francés, Portugues, Español, etc.	Inglés, Francés
Servicio geográfico	Procesamiento geográfico	Interacción Humana, Comunicación	Procesamiento geográfico
Tareas	Conversión de coordenadas, Transformación de coordenadas	Visualización geográfica Gestión remota de archivos	Visualización geográfica
Datos	contorno, coordenada, interpolación	vector, raster, gps, deteccion remota	datos espaciales

(*N/D: No Disponible; *N/A: No Aplicable)

Tabla 7. Información publicada del Esquema de Clasificación normalizado.

Grupo	Categoría	% de
Información general y comercial	Nombre	100
	Versión	90
	Sitio web	100
	Organización / desarrollador	98
	e-mail	96
	Teléfono	42
	Dirección postal	42
	Precio	86
	Artefacto	10
	Requerimientos de software	84
	Requerimientos de hardware	40
Clasificación	Tipo de Aplicación	80
	Sistema operativo	94
	Lenguaje de programación	96
	Estado	94
	Licencia	88
	Estándares	36
	Idioma	16
	Servicio geográfico	100
Funcionalidad	Tareas	100
	Datos	100

En la actualidad, la mayoría de los repositorios que almacenan información de componentes sólo permiten realizar compatibilidad sintáctica. Para mejorar la eficiencia de la búsqueda decidimos combinar este esquema con información ontológica que enriquezca semánticamente la información disponible en los catálogos web.

Basamos el diseño conceptual de la ontología en las guías propuestas en *Methontology* [23]. Para la actividad de conceptualización, *Methontology* propone el siguiente conjunto de tareas que nos permite identificar conceptos, atributos, relaciones, constantes, axiomas formales, reglas e instancias:

1. **Construir el glosario de términos:** En primer lugar, a partir de la información disponible en los catálogos web de los componentes, se construye un glosario de términos que incluye todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etc.), sus descripciones en lenguaje natural, y sus sinónimos y acrónimos. Las categorías pertenecientes a los grupos

“Clasificación” e “Información General y Comercial” del esquema de clasificación (mostrado en la Tabla 1) determinan en forma directa los principales términos: *Identificación, SitioWeb, Contacto, Idioma, Requerimientos, SistemaOperativo, Aplicación, LenguajeProgramacion, Estado, Licencia, Estandar, etc.* Las categorías del grupo *Funcionalidad*, se modelan principalmente por medio de las relaciones entre la clase *ComponenteSIG* y las clases *Tarea* y *Datos*.

- 2. Construir taxonomías de conceptos:** Una vez que el glosario de términos contiene suficientes términos, se construye la taxonomía de conceptos que define su jerarquía. METHONTOLOGY propone la construcción taxonomías considerando las relaciones *Subclase-Of, Disjoint-Decomposition, Exhaustive-Decomposition* y *Partition* que son las relaciones taxonómicas definidas en la *Frame Ontology* [21] y *OKBC Ontology* [11]. En nuestro caso las relaciones se establecen principalmente entre el concepto *ComponenteSIG* y el resto de los conceptos que representan propiedades o información relativa a él. Con el fin de enriquecer la semántica de las descripciones de componentes SIG, se define la entidad *ServicioGeografico* por medio de una relación de especialización, con las 6 subclases correspondientes a la taxonomía de servicios geográficos del estándar ISO/IEC 19119: *InteraccionHumana, AdministracionDeInformacion/Modelo, GestionDeTareas/Workflow, ProcesamientoGeografico, ComunicacionGeografica, AdministracionDeSistemasGeográficos*. Antes de proseguir con la especificación de nuevos conocimientos, se debe examinar que las taxonomías no contengan errores.
- 3. Construir diagramas de relaciones binarias ad hoc:** Una vez construida y evaluada la taxonomía, se definen las relaciones *ad hoc* existentes entre los conceptos de la misma. Por ejemplo, el concepto *Componente*, que representa un componente SIG, tiene asociado una relación *traducido-en* que hace referencia al idioma en el que está disponible un componente de software. La cardinalidad es como mínimo 1, pero un mismo *componenteSIG* puede estar traducido en más de un *Idioma*.

-
4. **Construir el diccionario de conceptos:** Una vez que las taxonomías de conceptos y las relaciones *ad hoc* han sido identificadas, se debe especificar cuáles son las propiedades que describen cada concepto de la taxonomía, así como las relaciones identificadas y las instancias de cada uno de los conceptos. En la Tabla 8 se muestra el diccionario de conceptos donde se indican las propiedades y relaciones que describen cada concepto de la taxonomía; además, contiene sus instancias y las clases e instancias de sus atributos.

 5. **Describir relaciones ad hoc, atributos de instancia, atributos de clase, y constantes:** Una vez construido el diccionario de conceptos, se debe describir con detalle cada una de las relaciones *ad hoc* y de los atributos de clase e instancia que aparecen. Además, se debe describir de manera precisa cada una de las constantes detalladas en el glosario de términos.

 6. **Definir axiomas formales y reglas:** En nuestro caso particular no existen reglas pero si ciertos axiomas que aseguran la correctitud de la ontología construida. Por ejemplo, se indican los valores posibles que puede tomar una determinada clase o concepto. En el caso de *Estado*, esta clase sólo podrá tomar valores *estable*, *alfa*, *beta*, *archivado*, que son representados como instancias de dicho concepto.

 7. **Describir instancias:** Finalmente se definen las instancias relativas a los conceptos representados en la ontología para asignar valores a las propiedades de dichos conceptos y los relacionados con ellos.

El modelo resultante, que se muestra en la Figura 7, representa las entidades que se refieren a los conceptos principales del dominio de conocimiento de los componentes SIG. Estas entidades tienen atributos y pueden relacionarse con otras entidades por medio de relaciones.

Tabla 8. Resumen del diccionario de conceptos de la ontología *ontocompoSIG*.

Concepto	Instancia	Atributo de clase	Atributo de instancia	Relacion
componenteSIG			ID, Nombre, Versión	publicadoEn desarrolladoEn liberadoEn ...
Sitio web			URL	
Contacto		Tipo de desarrollador	Nombre, e-mail, teléfono, dirección postal	
Idioma	español, alemán, francés			
Artefacto		Formato de entregable	Nombre, extensión	
Requerimientos				sonRequeridosPor
Software		Nombre, versión		
Hardware		Procesador, Memoria, Disco, otros		
Sistema operativo	GNU/Linux, otros Unices, Windows, MacOS X, Otros		Versión, licencia	
Aplicación	Escritorio, Móvil, Web, Base de Datos, Librería			
Lenguaje	C, Java, C++, Python, Tcl/Tk, PHP, Perl, (Visual) Basic, C#, Fortran, Delphi, Ruby, SWF, Javascript, Otros			
Estado	estable, alfa, beta, archivado			
Licencia				esOfrecidaPor
Comercial			Precio	
No Comercial	GNU GPL, GNU LGPL, MIT, Public Domain, Creative Commons			
Estándar	OGC WMS, OGC WFS			
Servicio Geográfico	Taxonomía del estándar ISO/IEC 19119			
Tarea	Tareas que realizan los componentes SIG			
Datos	Mapas, censos, encuestas, reportes, mapas, fotografías, estadísticas			

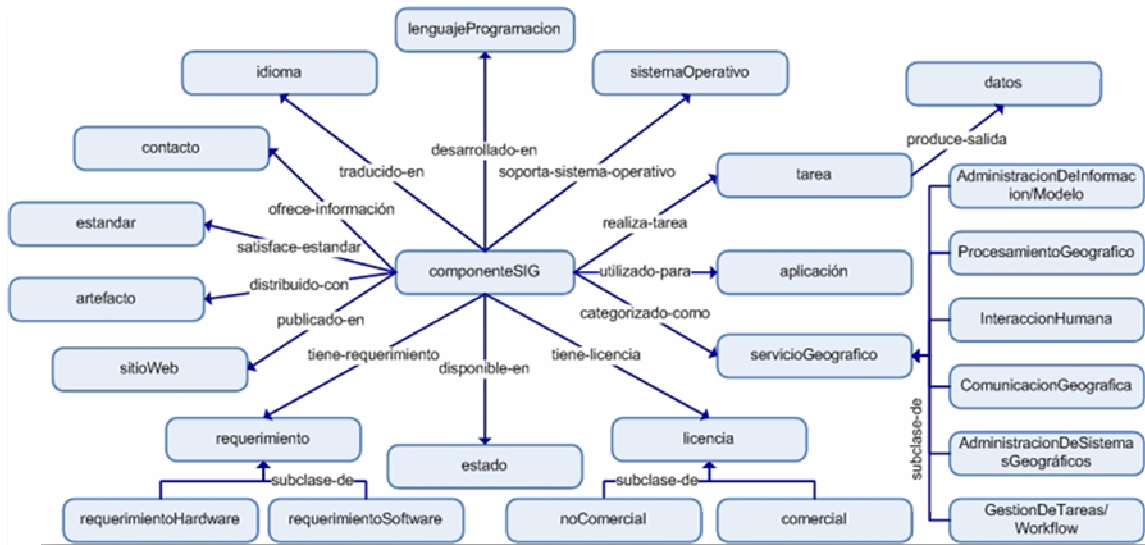


Figura 7. Relaciones entre conceptos de la ontología

Aunque las ontologías livianas (pobres en axiomas) presentan algunas deficiencias por ser menos expresivas, consideramos, al igual que en [47], que una ontología liviana es suficiente para la definición de las entidades, sus atributos y relaciones y además es suficiente para la semántica de nuestro trabajo.

En la siguiente sección presentamos un Proceso de Publicación de componentes, aplicado al dominio de los SIG. Primero describimos en forma general dicho proceso y luego detallamos cada una de sus partes. Al final, brindamos algunos detalles sobre el prototipo que implementa nuestra propuesta.

3.2. Publicación de Componentes SIG

En este trabajo proponemos un Proceso de Publicación de componentes SIG, basado en el Esquema de Clasificación normalizado (Tabla 5) que considera información funcional y no técnica de componentes SIG. Lo novedoso de esta propuesta es que cubre todas las tareas generales del proceso de Publicación mencionadas en la Sección 2.1: *Clasificar, Documentar y Almacenar el componente*; y, además introduce una

ontología liviana y técnicas de procesamiento de lenguaje natural para mejorar la identificación y posterior recuperación de componentes descriptos en los catálogos web. El Proceso de Publicación de componentes se basa en un conjunto de elementos clave: (1) definición de un esquema de clasificación, adaptado a estándares del dominio SIG y a información disponible en catálogos Web; (2) extracción de información desde catálogos web por medio de técnicas de anotación semántica; y (3) normalización de información.

La Figura 8 representa esquemáticamente el Proceso de Publicación de Componentes propuesto.

1. La *Ontología* que usa el Proceso de Publicación es generada por el módulo *Creación de Ontología*. Este proceso es asistido por un *Experto* humano y se implementa por medio de un editor de ontologías como Protégé.
2. El *Usuario "Publicador"* procesa una *Descripción web* (que representa la información sobre los componentes encontrada en los catálogos) por medio del módulo *Extracción de Información*. Este módulo aplica técnicas de anotación semántica (descrita en la sección 2.3.6) para lo cual se utiliza la *Ontología* generada.
3. El módulo *Normalización de información* estructura el documento que contiene la *Descripción web* en forma de una *Descripción normalizada*. Antes de almacenar definitivamente la información normalizada en el *Repositorio Estandarizado*, también se evalúa la validez de las anotaciones obtenidas automáticamente.
4. El proceso de Publicación se puede relacionar con otro proceso externo: *Proceso de Consulta*, mediante el cual, un *Usuario ("Buscador")* que busca información sobre componentes podría hacer uso de la información extraída y almacenada en el *Repositorio estandarizado*.

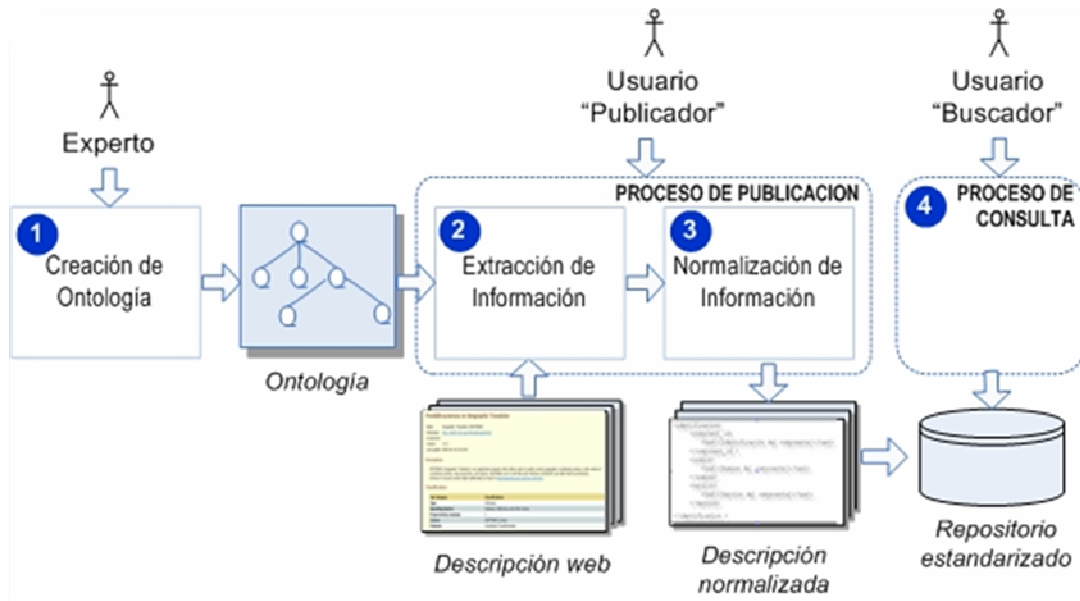


Figura 8. Principales partes del Proceso de Publicación de componentes.

3.3. Creación de la Ontología

Este módulo se encarga de la implementación de la ontología *ontoCompoSIG*, para ello se utiliza el editor de ontologías Protégé v4.1.0 y se genera una ontología en formato OWL-DL. Como punto de partida usamos una ontología disponible (“*ISO 19119 Service Type*”²¹ de GEOBRAIN) que representa la taxonomía de servicios geográficos del estándar ISO/IEC19119.

Como se describió en la Sección 3.1, el modelo conceptual de la ontología se basa principalmente en el Esquema de clasificación normalizado, que pretende organizar en forma estructurada la información que describe a los componentes SIG. Estos elementos conceptuales se implementan en tres clases principales:

- *Funcionalidad*: esta clase representa los aspectos funcionales de los componentes SIG y está formada por dos sub-clases:

²¹ http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl

-
- *Tarea*: con esta clase se identifica la tarea que realiza el componente. Para modelar las jerarquías funcionales no existen reglas universalmente aceptadas. El estilo de modelado OWLS/ IRS/WSMO incluye tanto el verbo de la acción como un elemento de datos involucrado directamente en la funcionalidad (por ejemplo, AlmacenarArchivo); mientras que otros estilos modelan las funcionalidades como conceptos que denotan acción (Remover) sin ninguna conexión a la estructura de datos. En esta versión de la ontología se presentan jerarquías funcionales de ambos estilos, (o sea, creando frases verbo-nombre, o solo verbo), ya que nuestro trabajo está mas enfocado con la correcta extracción de la información necesaria que con la forma en que está conceptualizada.
 - *Datos*: esta clase representa todos los objetos que pueden ser considerados como entradas o salidas de las tareas que realiza el componente.

 - *Descripcion*: esta clase representa los datos particulares que describen a los componentes SIG. Se organiza en las siguientes subclases:
ComponenteSIG, Contacto, RequerimientoSoftware, RequerimientoHardware, Artefacto, SitioWeb.

 - *Clasificacion*: esta clase representa facetas de clasificación de los componentes SIG. Se organiza en las siguientes subclases:
Estandar, SistemaOperativo, Aplicacion, EstadoDesarrollo, Idioma, Licencia, LenguajeProgramacion, ServicioGeografico (las diferentes sub-clases se organizan usando la ontología “*ISO 19119 Service Type*” de GEOBRAIN).

El documento que contiene la ontología completa se encuentra disponible en el Apéndice B.

3.4. Extracción de Información

El primer paso del *Proceso de Publicación* consiste en extraer toda la información relevante relacionada con el dominio de los componentes SIG a partir de documentos disponibles en la web. El módulo Extracción de información analiza los documentos de acuerdo a un conjunto de vocabularios y patrones de extracción; localiza la información a ser extraída en el documento y genera los meta-datos semánticos correspondientes.

El mapeo entre los meta-datos semánticos y la ontología *ontoCompoSIG* se realiza usando reglas gramaticales. Estas reglas mapean uno o más meta-datos semánticos con un elemento de la ontología de dominio. Una regla identifica el meta-dato semántico que activa el proceso de anotación.

Como se muestra en la Figura 9, esto es realizado por una serie de pasos, en los que el procesamiento de cada paso agrega anotaciones a las ya existentes.

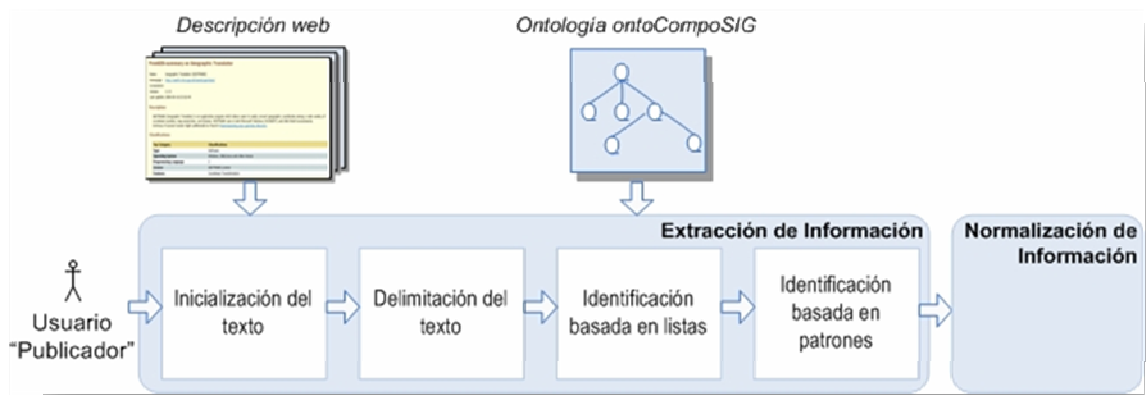


Figura 9. Detalle del Módulo de Extracción de Información basado en la ontología

El primer paso de *Inicialización del texto* realiza la preparación del documento a ser procesado liberándolo de cualquier etiqueta que no sea necesaria para el resto del proceso.

En el paso de *Delimitación del texto*, se divide al texto en bloques de construcción básica (palabras, sentencias o párrafos) y se generan etiquetas que identifican estos bloques básicos para usos futuros.

Para la Identificación de las entidades del Esquema de clasificación normalizado dentro del texto disponible se plantean dos estrategias:

1. *Identificación basada en listas enumeradas*. Esta identificación se basa en una lista que permite enumerar los valores posibles de un concepto y asociarlos a una ontología. Por ejemplo, la categoría ‘Estado de desarrollo’ de un componente tiene como valores posibles: ‘stable, alfa, beta, archived’, y se asocia al concepto ‘EstadoDesarrollo’ de la ontología *OntoCompoSIG*. Se aplica a aquellas entidades cuyos valores posibles son fácilmente enumerables: ‘Estado de Desarrollo’, ‘Estándar’, ‘Sistema Operativo’, ‘Aplicación’, ‘Idioma’, ‘Licencia’, ‘Lenguaje de Programación’, ‘Tarea’, ‘Datos’.
2. *Identificación basada en patrones del contexto*. Esta identificación se basa en la aplicación de reglas gramaticales solo si ocurre una cierta situación en el texto. Por ejemplo, se define una regla para la categoría ‘Nombre de Componente’, que indica que el nombre de un componente solo puede ser reconocido si, en el texto, éste ocurre precedido por la palabra ‘Name’. En este paso también sería posible aplicar reglas que permitan inferir conocimiento no declarado explícitamente en el documento original. Se aplica a aquellas categorías cuyos valores posibles no son fácilmente enumerables: ‘Versión de Componente’, ‘Dirección de Contacto’, ‘Email de Contacto’, ‘Nombre de Contacto’, ‘Teléfono de Contacto’, ‘Requerimiento de Software’, ‘Requerimiento de Hardware’, ‘Precio’, ‘Artefacto’, ‘Sitio Web’.

El resultado final del proceso de Extracción de información está formado por el texto (encontrado en el documento original) mas un conjunto de anotaciones conceptuales que enlazan partes de ese texto con conceptos de la ontología específica del dominio SIG (*ontoCompoSIG*).

3.5. Normalización de información

Como se muestra en la Figura 10, este módulo está formado por dos pasos secuenciales: Evaluación de anotaciones y Estructuración de información.

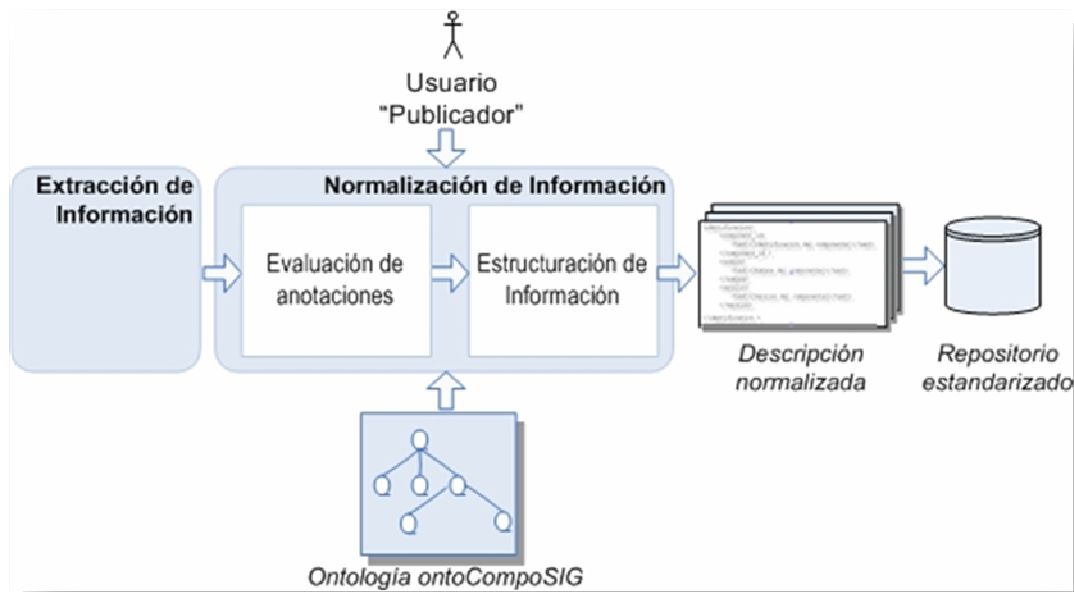


Figura 10. Detalle del Módulo Normalización de Información.

En el paso de Evaluación de anotaciones se pretende garantizar la completitud y correctitud de la información que se almacenará en el Repositorio estándar. En esta versión de nuestra propuesta, el usuario interviene para analizar las anotaciones automáticas que realizó el módulo Extracción de Información, de modo que sea posible completar correctamente la máxima cantidad de elementos del Esquema de clasificación normalizado.

Durante la Estructuración de información se organiza la información disponible (el contenido original del documento y las anotaciones realizadas) para dar forma a un documento que contenga la *Descripción normalizada*. Esta información, que se almacenará en el *Repositorio estandarizado*, consiste en un conjunto de meta-datos que

permiten completar el Esquema de clasificación normalizado (Tabla 5). Luego, otros usuarios pueden acceder a este repositorio para encontrar y recuperar aquellos componentes que más se adecuen a sus necesidades.

3.6. Herramienta de Soporte

Una parte importante de nuestro trabajo fue la implementación de un prototipo para automatizar el proceso de Publicación propuesto. Nuestro objetivo era construir un sistema que sea fácil de usar, modular, fácilmente modificable y con posibilidades de adaptación a nuevos dominios de aplicación.

Existen numerosas herramientas que brindan soporte para desarrollar aplicaciones en el dominio del Procesamiento de Lenguaje Natural. Para este trabajo, se eligió GATE (*General Architecture for Text Engineering*) porque, según se sostiene [15], ha sido desarrollado para ofrece robustez, re-usabilidad y escalabilidad. Además, ofrece soporte para aplicar técnicas de procesamiento de lenguaje basadas en ontologías, está basada en Java y se encuentra disponible bajo licencia LGPL (*Lesser GNU Public Licence*).

GATE posee una arquitectura que define la estructura organizacional de una aplicación de procesamiento de lenguaje natural, y también un framework, que proporciona un conjunto de bloques o módulos que se pueden reusar, extender o adaptar para construir aplicaciones de procesamiento de lenguaje natural. Un ambiente de desarrollo gráfico ayuda al usuario a disminuir el tiempo de desarrollo.

GATE permite construir aplicaciones modulares combinando Recursos de Lenguaje (*Language Resources -LRs-*), entre los que se incluyen los documentos, las corporas (colecciones de documentos), ontologías y léxicos, y Recursos de Procesamiento (*Processing Resources -PRs-*), que representan cualquier recurso algorítmico o que realice algún procesamiento sobre los recursos de lenguaje como intérpretes, generadores, modeladores, etc.

Además de la posibilidad de combinación de PRs, GATE provee una interfase de usuario para la inspección de los resultados producidos por cada uno de estos recursos.

Las aplicaciones en GATE son conjuntos de Recursos de Procesamiento agrupados y ejecutados en sucesión sobre algún Recurso de Lenguaje. Una vez que se cargan todos los recursos, se puede crear y ejecutar una aplicación.

Los recursos de procesamiento (tokenizadores, parseadores, etc.) producen información acerca de los textos sobre los cuales se ejecutan. Cuando se ejecuta un tokenizador, a cada palabra se le asigna un *token-type* (*word, number, punctuation, etc*); cuando se ejecuta un etiquetador morfosintáctico, a cada palabra se le asigna una categoría gramatical (*proper noun, verb, etc*). Esta información se representa como un conjunto de anotaciones.

Una anotación en GATE está formada por: ID (una identificación única en el documento al que la anotación se refiere), *type* (el cual denota el tipo de anotación), *StartNode* y *EndNode* (los cuales denotan la posición inicial y la posición final que ocupa en el texto original la palabra que se está anotando) y un conjunto de *features* (pares atributo/valor que proporcionan información adicional acerca de la anotación).

JAPE (*Java Annotation Patterns Engine*) es un módulo de GATE que permite realizar transformaciones en las anotaciones de un texto previamente anotado (al menos con la anotación *token*). Utiliza como insumo una gramática JAPE (basada en expresiones regulares) y un texto anotado. La gramática es un conjunto de fases, cada una con reglas patrón/acción, que se ejecutan en forma sucesiva sobre el texto anotado.

Las reglas patrón/acción especifican en el lado izquierdo un patrón de anotación y en el lado derecho, una acción que se ejecuta si es que el lado izquierdo se satisface.

El lado izquierdo de la regla contiene un patrón de anotación que puede contener operadores de expresiones regulares (e. g. *, ?, +). El lado derecho contiene el nombre de la anotación. Junto con esta anotación se pueden especificar los atributos que se quieran asignar a la anotación. También es posible agregar un bloque válido de código Java.

ANNIE (*a Nearly-New Information Extraction System*) es un sistema de extracción de información que se distribuye con GATE y usa técnicas de máquinas de estado finito para implementar tareas como tokenización, etiquetado semántico y particionamiento de frases. ANNIE cuenta con los siguientes módulos:

- *Tokeniser*. Consta de dos partes, una divide al texto en *tokens* simples (números, símbolos de puntuación, palabras) y la otra, mediante el uso de JAPE, permite construir *tokens* más complejos.
- *Sentence Splitter*. Delimita las oraciones de un texto. Es una cascada de traductores de estado finito los cuales segmentan el texto en oraciones. Usa una lista de abreviaturas (*Gazetteer*) para distinguir los puntos finales de otras clases de puntuación. Cada oración es anotada con el tipo Sentencia. Este módulo es independiente del dominio. El módulo *RegEx Sentence Splitter*, que intenta mejorar aspectos de tiempo de ejecución y robustez con respecto al módulo estándar está basado en expresiones regulares.
- *Part of Speech Tagger*. Este módulo produce un *tag* como una anotación sobre cada palabra o símbolo. El *tagger* usa un *lexicon* y un conjunto de reglas por defecto, las cuales se pueden modificar manualmente si es necesario.
- *Gazetteer*. Una lista *gazetteer* es un archivo índice (*list.def*) que contiene el conjunto de listas (cada una de las cuales también es un archivo) y el tipo asociado para cada lista. Cada lista representa un conjunto de nombres de entidades, como nombres de ciudades, organizaciones, días de la semana, etc. El archivo que define el conjunto de listas debe ser escrito en una sola línea y contener: nombre del archivo, ‘majorType’ (tipo principal), ‘minorType’ (tipo secundario). Por ejemplo, para la siguiente línea del archivo índice:

club_futbol.lst:club:futbol

y el siguiente contenido de la lista club_futbol.lst:

Real Madrid
Boca Juniors
River Plate

el *gazetteer* anotará el texto ‘Real Madrid’ con los atributos ‘majorType’= club y ‘minorType’=futbol.

- *Orthomatcher*. Este módulo ayuda a identificar relaciones entre entidades para realizar coreferencias. También mejora el reconocimiento de las entidades

nombradas al asignar anotaciones a nombres previamente no clasificados, en base a las relaciones con entidades existentes.

Una aplicación GATE consiste en una secuencia de estos módulos, en la cual cada uno realiza el procesamiento correspondiente. ANNIE se basa en el transductor JAPE y puede ser fácilmente extensible para agregar nuevos diccionarios y reglas.

3.7. Primer Prototipo

Este primer prototipo contruido como soporte a la propuesta, se enfoca en la extracción de información de componentes disponibles en sólo uno de los catálogos web evaluados: *FreeGIS*, que al momento de esta implementación, ofrece información de 350 productos.

Los pasos del módulo Extracción de Información, que se muestran en la Figura 11, se implementaron completamente desarrollando una aplicación, llamada ‘Extracción de información’, conformada por una secuencia de componentes disponibles en GATE.

El paso de *Inicialización del texto* se implementó con el componente *Document Reset*, que fue configurado para transformar el documento original en un documento en texto plano, quitando las etiquetas html del documento original.

El paso de *Delimitación de texto*, utilizó el componente *ANNIE English Tokenizer*, que separó el texto en palabras y las anotó con la etiqueta *token*.

El paso *Identificación basado en listas*, se basó en la utilización del componente *OntoGazetteer*. Para la configuración de este componente GATE ofrece un Recurso Visual, llamado *Ontogazetteer Editor*, que permite cargar/editar/almacenar los siguientes parámetros que necesitamos configurar:

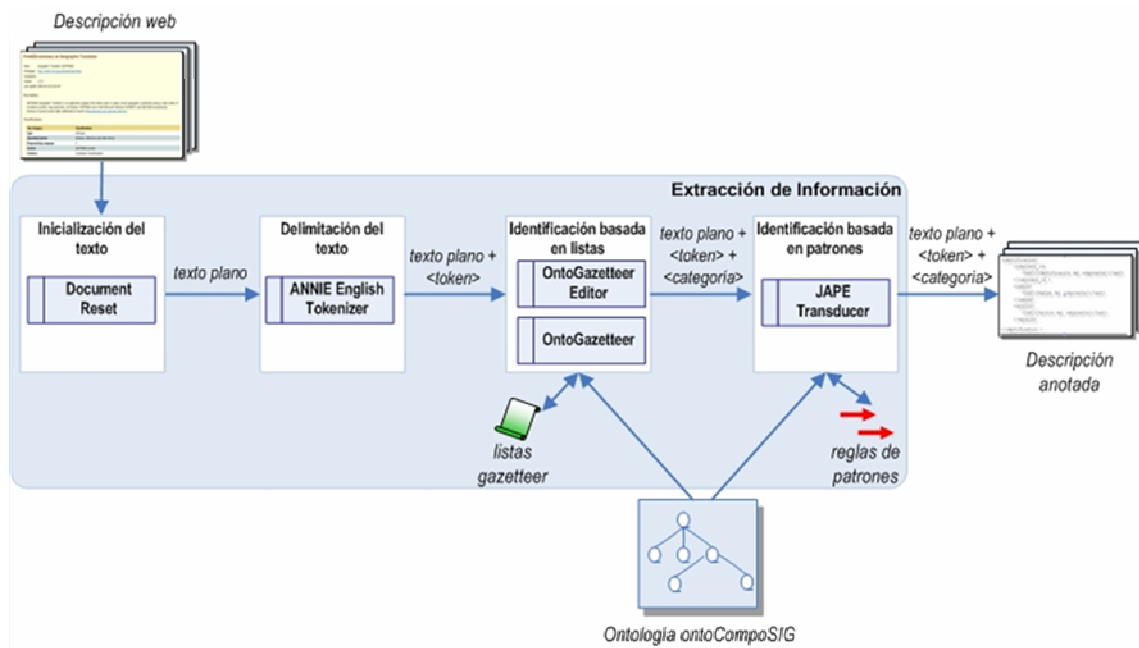


Figura 11. Implementación del módulo Extracción de Información.

- *Las listas de valores.* Se creó una lista de los valores posibles de los conceptos de la ontología *ontoCompoSIG*; cada lista es almacenada en un archivo plano con extensión *.lst*. Por ejemplo, para el concepto ‘licencia’ creamos una lista de los tipos de licencia conocidos dentro de un archivo llamado ‘licencia.lst’.
- *Las relaciones con los conceptos de la ontología.* Se genera un mapeo entre cada lista generada y la clase correspondiente de la ontología, cada mapeo se guarda como una línea en el archivo ‘mapping.def’. Por ejemplo, para asociar la lista de valores posibles (*licencia.lst*) con la clase ‘licencia’ de la ontología *ontoCompoSIG* (*OntoCompoSIG.owl:Licencia*) agregamos una línea similar a esta:
licencia.lst:http://www.semanticweb.org/ontologies/2010/10/OntoCompoSIG.owl:Licencia
- *Los tipos de anotación.* Se definen los tipos de anotación que debe generar el *Ontogazetteer* para los valores de las listas que se han creado. Cada definición se guarda como una línea del archivo ‘list.def’. Por ejemplo, para definir que el tipo de anotación correspondiente a la lista del concepto ‘licencia’ es ‘licenciaComponente’, agregamos una línea similar a esta:

El paso *Identificación basado en reglas*, se basó en la definición de un conjunto de reglas gramaticales que son ejecutadas por medio del componente *JAPE Transducer*. Estas reglas de patrones se escriben como archivos en texto plano con extensión ‘.jape’. En el siguiente ejemplo se muestra la regla que permite identificar el nombre de un componente, luego de reconocer que en el texto aparece la palabra ‘Version:’ (línea 5), y a continuación un número, seguido de un punto y de otro número (línea 6). Al texto identificado se le asigna el atributo ‘class’ para asociarlo a la clase correspondiente de la ontología (línea 8).

1. Phase:patternVersion
2. Input: Token
3. Options: control = appelt
4. Rule: Version
5. ({Token.string == "Version"} {Token.string == ":"})
6. ({Token.kind==number} {Token.string=="."} {Token.kind==number}):versionLabel
7. -->
8. :versionLabel.Lookup={rule="Version",class="http://www.semanticweb.org/ontologies/2010/10/OntoCompoSIG.owl#VersionComponente"}

La Figura 12 ilustra el resultado de ejecutar la aplicación ‘Extracción de información’ con un documento que contiene el texto original agregado con una serie de anotaciones de tipo ‘Lookup’; cada una de estas anotaciones, contiene el atributo (*feature*) ‘class’ cuyo valor indica la URI de la clase correspondiente en la ontología *ontoCompoSIG*.

Por otro lado, el módulo *Normalización de información* aprovecha otras facilidades que ofrece GATE para implementar los pasos que lo forman.

La *Ontology Annotation Tool (OAT)* es una herramienta que permite que el usuario realice anotaciones manualmente con respecto a una o más ontologías. El paso de *Evaluación de anotaciones* se realiza aprovechando las facilidades que esta herramienta ofrece al usuario para:

- Visualizar las ocurrencias de cada clase dentro del texto, identificando cada clase con un color diferente.
- Borrar o cambiar una anotación existente dentro del texto. En la Figura 13 se muestra la posibilidad de cambiar o borrar la clase con la que está relacionada una anotación; también es posible cambiar los límites del texto que fue anotado.
- Agregar nuevas anotaciones con respecto a la ontología seleccionada.

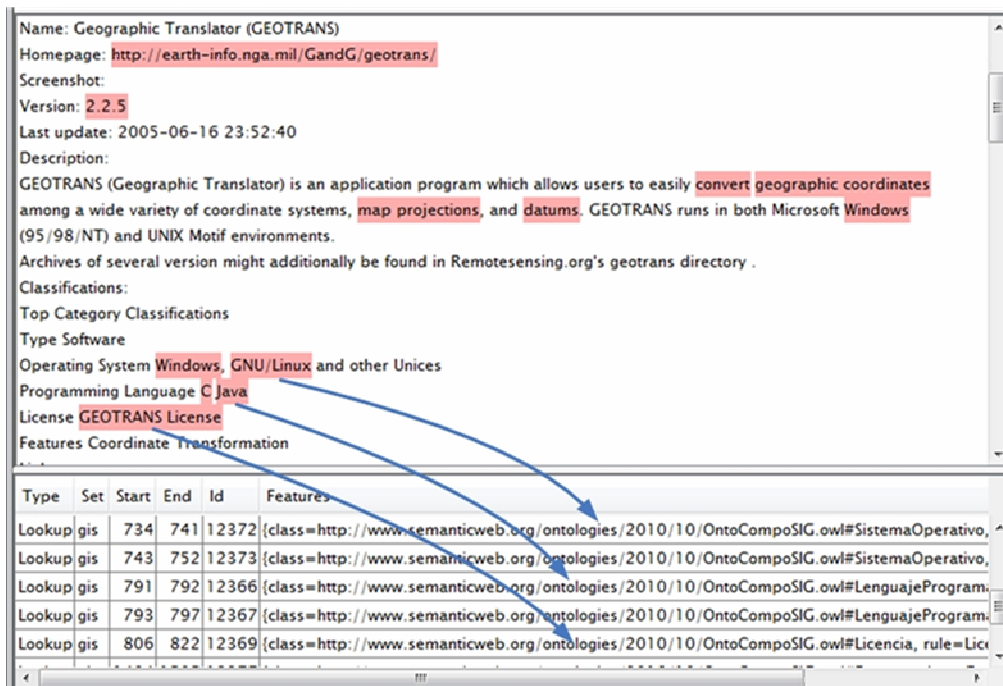


Figura 12. Detalle de las anotaciones realizadas por la aplicación Extracción de Información

El paso de Estructuración de información se implementa con un componente *JAPE Transducer*, que por medio de un conjunto de reglas gramaticales determina qué partes del documento anotado se almacenan en el *Repositorio estandarizado*. Para asegurar el uso de una sintaxis común, se almacena la *Descripción estandarizada* en formato XML/RDF.

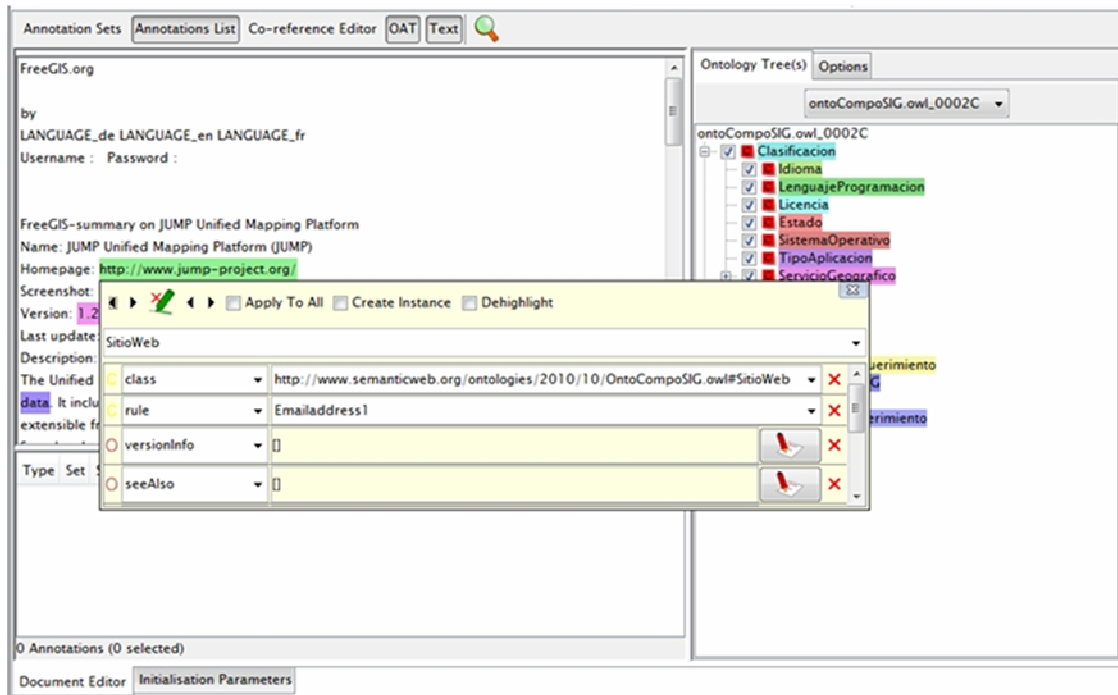


Figura 13. Cuadro de diálogo de la herramienta OAT para cambiar o eliminar anotaciones con respecto a una clase de la ontología.

3.8. Escenario de Uso

Suponemos que existe un desarrollador de aplicaciones basadas en componentes que necesita estandarizar descripciones de componentes encontrada en la web, y publicarlas en un nuevo repositorio. Estas descripciones en particular están referidas a componentes para SIG y están escritas en lenguaje natural.

Precondición: el prototipo ha sido configurado para trabajar en base a la ontología *ontoCompoSIG*.

1. El usuario selecciona un documento desde un repositorio local, que contiene un conjunto de documentos con información de los componentes SIG ofrecidos en el

catálogo FreeGIS. Para este ejemplo seleccionaremos la descripción del componente “*Geographic Translator*”²² que se muestra en la Figura 14.

FreeGIS-summary on Geographic Translator

Name: Geographic Translator (GEOTRANS)
Homepage: <http://earth-info.nga.mil/GandG/geotrans/>
Screenshot:
Version: 2.2.5
Last update: 2005-06-16 23:52:40

Description:

GEOTRANS (Geographic Translator) is an application program which allows users to easily convert geographic coordinates among a wide variety of coordinate systems, map projections, and datums. GEOTRANS runs in both Microsoft Windows (95/98/NT) and UNIX Motif environments. Archives of several version might additionally be found in [Remotesensing.org's geotrans directory](http://Remotesensing.org's_geotrans_directory).

Classifications:

Top Category	Classifications
Type	Software
Operating System	Windows, GNU/Linux and other Unices
Programming Language	C
License	GEOTRANS License
Features	Coordinate Transformation

Links:

No links have been defined.

Changes:

Figura 14. Descripción de un componente en lenguaje natural presentada en un catálogo web.

2. El usuario selecciona la ontología *ontoCompoSIG*.
3. El usuario inicia la aplicación ‘Extracción de información’ para el documento seleccionado.
4. El prototipo remueve las anotaciones previas y cualquier marca (*markup*) no deseada del documento seleccionado. En este caso, se eliminan las etiquetas de marcado HTML que contiene el documento original.
5. El prototipo divide el texto en *tokens* y crea anotaciones *Token* en el documento.
6. El prototipo identifica los términos relacionados con las clases de la ontología: ‘Sistema Operativo’, ‘Licencia’, ‘Lenguaje de Programación’, ‘Tarea’ y ‘Datos’. Crea anotaciones *Lookup* con el atributo ‘class’ relacionado con la clase correspondiente de la ontología.
7. El prototipo identifica otros detalles como ‘Nombre’, ‘Version’ y ‘Sitio web’ en base a una serie de reglas gramaticales basadas en el contexto. Crea anotaciones

²² <http://www.freegis.org/database/viewobj?obj=602>

-
- Lookup* con el atributo ‘*class*’ relacionado con la clase correspondiente de la ontología.
8. El prototipo permite que el usuario (1) evalúe el resultado del proceso de extracción automático y (2) pueda hacer los cambios que considere necesarios por medio de anotación manual.
 9. Finalmente, el prototipo recolecta solamente las anotaciones importantes realizadas sobre el documento aplicando un conjunto de reglas gramaticales.
 10. El usuario almacena el documento resultante como un documento XML, en el *Repositorio estandarizado*.

Como resultado del procesamiento antes detallado, en la Tabla 9 se muestran los términos identificados para de la descripción web del componente *GeographicTranslator*. El prototipo también podría realizar algún tipo de inferencia que permita identificar la información necesaria para completar categorías del esquema de clasificación que no es posible completar por relación directa con la ontología (por ejemplo, la clase *CategoriaServicioGeográfico*); sin embargo, en esta versión del prototipo no se incorporó esta facilidad.

Tabla 9. Información de *Geographic Translator* relacionada con la ontología.

Concepto	Descripción del componente SIG
Nombre	<i>GeographicTranslator</i>
Version	2.2.5
sitioWeb	http://earth-info.nga.mil/GandG/geotrans
SistemaOperativo	Windows, GNU/Linux, other Unices
lenguajeProgramacion	C
Licencia	GEOTRANSLicense
Datos	Geographic Coordinates, Map projections
Tarea	Convert

El documento que contiene la descripción normalizada del componente *GeographicTranslator* se muestra en el Apéndice A.

4. EVALUACIÓN EXPERIMENTAL

Este capítulo describe el caso de estudio realizado para validar la propuesta de normalización de información. Adecuándonos la metodología mencionada en la sección 1.2, introduciremos el caso de estudio siguiendo sus etapas.

4.1. Definición

Este caso de estudio fue realizado para medir la performance del prototipo de Extracción de Información de Componentes, usando las métricas estándar de recuperación de información: precisión, cobertura (*recall*) y *F-Measure*. El corpus seleccionado consistió de 50 descripciones en lenguaje natural de componentes SIG.

4.2. Planificación

Los documentos sobre los que se realiza esta evaluación experimental son obtenidos a partir del catálogo web disponible en FreeGIS²³. Cada documento del catálogo contiene información semi-estructurada sobre un componente SIG, y está escrito en lenguaje natural, en idioma inglés.

Para el proceso de evaluación se seleccionaron aleatoriamente 50 documentos de aproximadamente 50 kb. de tamaño cada uno, que fueron accedidos manualmente por medio de un navegador web y almacenados con formato HTML en un repositorio local. Este conjunto de documentos conforman lo que se conoce como Corpus; a su vez, este conjunto total se divide en dos corpus distintos utilizados en las distintas iteraciones del proceso:

- Corpus-1: formado por un conjunto de 10 documentos, utilizados en la Iteración-1 e Iteración-2.

²³ <http://freegis.org/>

-
- Corpus-2: formado por los 40 documentos restantes, utilizados en la Iteración-3.

Para poder comparar la calidad de las anotaciones realizadas por nuestro prototipo necesitamos alguna métrica de evaluación que ofrezca mecanismos aplicables de una forma simple y repetible. Tradicionalmente, la mayoría de las tareas relacionadas con el Procesamiento de Lenguaje Natural, como es la Extracción de Información, son evaluadas usando Precisión, Cobertura (*Recall*) y Medida-F (*F-measure*). La larga tradición de aplicación de estas métricas en el campo de la Recuperación de Información [51], [24] se refleja por ejemplo, en las evaluaciones en gran escala realizadas en las conferencias MUC (*Message Understanding Conferences*).

La Precisión mide el número de entidades identificadas correctamente como un porcentaje de todas las entidades identificadas. Esto mide cuántas de las entidades que el sistema encontró fueron realmente correctas. Cuanto más alta es la precisión, el sistema más asegura que lo que ha identificado es correcto. Se define como:

$$\text{Precisión} = (\text{Correctas} + 1/2\text{Parciales}) / (\text{Correctas} + \text{Falsas} + \text{Parciales})$$

La Cobertura mide el número de entidades identificadas como un porcentaje de las entidades correctas. Cuanto más alto es el índice de cobertura, mas asegura el sistema que no pierde entidades correctas. Se define como:

$$\text{Cobertura} = (\text{Correctas} + 1/2\text{Parciales}) / (\text{Correctas} + \text{Perdidas} + \text{Parciales})$$

La Medida-F combina Cobertura y Precisión en una única medida. Se define como:

$$\text{F-measure} = ((\beta^2 + 1) \text{Precisión} * \text{Cobertura}) / ((\beta^2 \text{Precisión}) + \text{Cobertura})$$

donde β es un factor que indica la importancia relativa de la cobertura y la precisión. Si en la evaluación se considera que la precisión y cobertura tienen igual importancia, se tomará $\beta=1$. En este caso la Medida-F es conocida como F_1 (*media armónica*):

$$F\text{-measure} = 2 (\text{Precisión} * \text{Cobertura}) / (\text{Precisión} + \text{Cobertura})$$

En las fórmulas, Falsas representa a aquellas entidades que son anotadas como correctas, pero no lo son; Parciales a aquellas identificadas parcialmente y Perdidas a aquellas entidades que no son encontradas por medio del procesamiento de anotación automático.

4.3. Operación

Para evaluar la performance del prototipo es necesario tener un texto de referencia, generalmente conocido como “*gold standard*”, que es una anotación manualmente realizada, y se usa para hacer comparaciones con los resultados derivados del proceso de anotación automático.

Durante el proceso de anotación manual, el anotador lee el documento de principio a fin marcando todas las anotaciones que encuentra. En primer lugar, lee el documento completo para obtener un entendimiento del contenido. Luego, en una segunda lectura, agrega anotaciones para las menciones de las entidades básicas. Finalmente, se recomienda revisar nuevamente el texto para asegurarse que las anotaciones son correctas.

El diseño de este experimento, incluye un proceso de evaluación formado por tres pasos:

- **Anotación manual:** El anotador humano realiza las anotaciones manuales sobre cada documento que integra el Corpus-*n* para crear el corpus *gold standard*. Se obtiene el Corpus-*n*-manual. En la Figura 15, se muestra con un ejemplo la secuencia de pasos para realizar la anotación manual aprovechando la herramienta OAT (*Ontology Annotation Tool*) que ofrece GATE:
 1. En la ventana ‘ResourcesTree’, se elige el documento del corpus que se desea anotar, en este caso: ‘FreeGIS_org1.htm_0002F’.

2. En la ventana ‘Document Editor’, se selecciona la porción de texto que se desea anotar, por ejemplo: ‘Geographic Resources Analysis Support System (GRASS)’.
3. En el cuadro de diálogo emergente, se selecciona la entidad de la ontología *ontoCompoSIG* asociada al texto seleccionado en el paso 2. En este caso: ‘NombreComponente’.

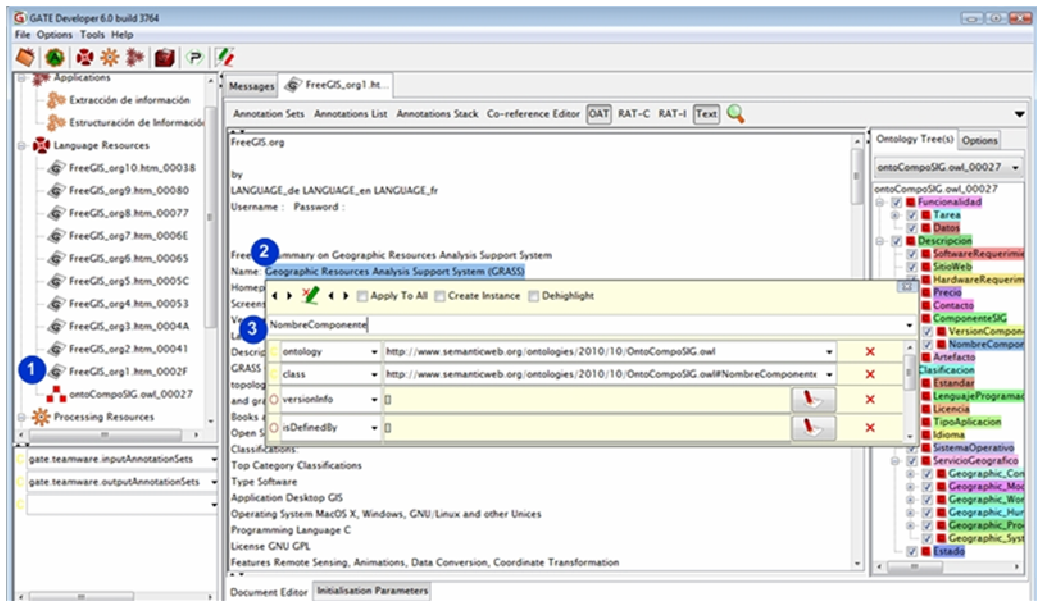


Figura 15. Ejemplo de anotación manual con GATE.

4. Los pasos 2 y 3 se repiten hasta terminar de anotar el documento completo.
 5. Se guarda el documento anotado manualmente como ‘MFreeGIS_org1.xml’.
 6. Los pasos 1, 2, 3, 4 y 5 se repiten hasta terminar de anotar todos los documentos del corpus.
- **Anotación automática:** Sobre los mismos documentos que integran el Corpus-*n* (original, sin anotaciones) se ejecuta la aplicación ‘Extracción de Información’ para anotarlo automáticamente. Se obtiene el Corpus-*n*-automático. En la Figura

16 se muestran los pasos a seguir para realizar la anotación automática por medio de una aplicación GATE:

1. Desde la ventana ‘ResourcesTree’, se selecciona la aplicación GATE desarrollada para anotar automáticamente un texto. En este ejemplo sería la aplicación ‘Extracción de información’.
2. En la ventana ‘Serial Application Editor’, se selecciona el corpus que contiene los documentos sin anotaciones. En este caso, ‘Corpus-1’.
3. Se ejecuta la aplicación (‘Extracción de información’) por medio del botón ‘Run this application’.

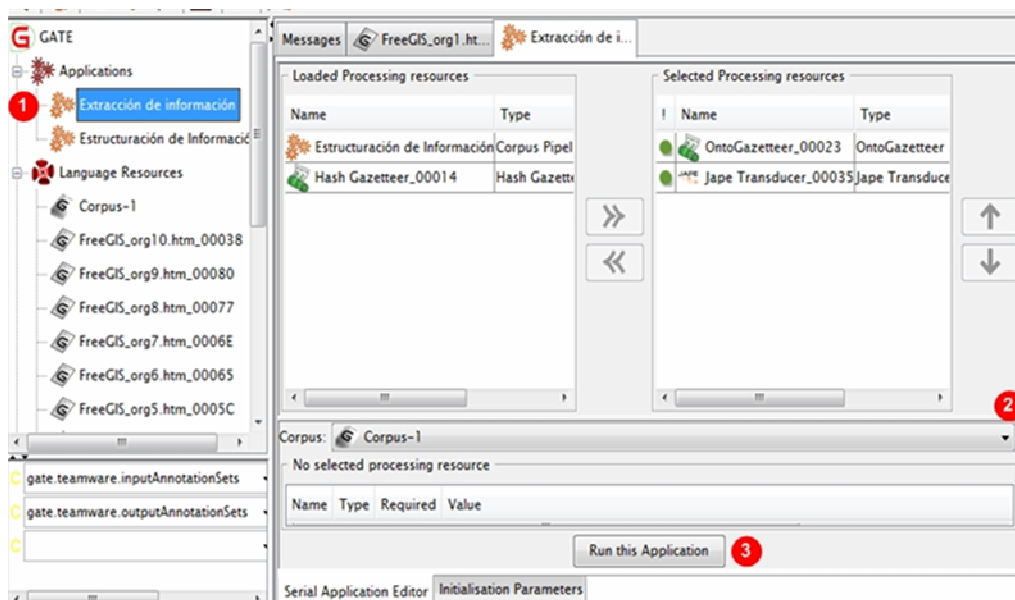


Figura 16. Anotación automática con GATE.

- **Comparación:** Cada documento del Corpus-*n*-automático es analizado con respecto a su similar del Corpus-*n*-manual para obtener las medidas Precisión y Cobertura. En la Figura 17 se muestran los resultados de ambos procesos de anotación (manual y automático) aplicados a un mismo documento.

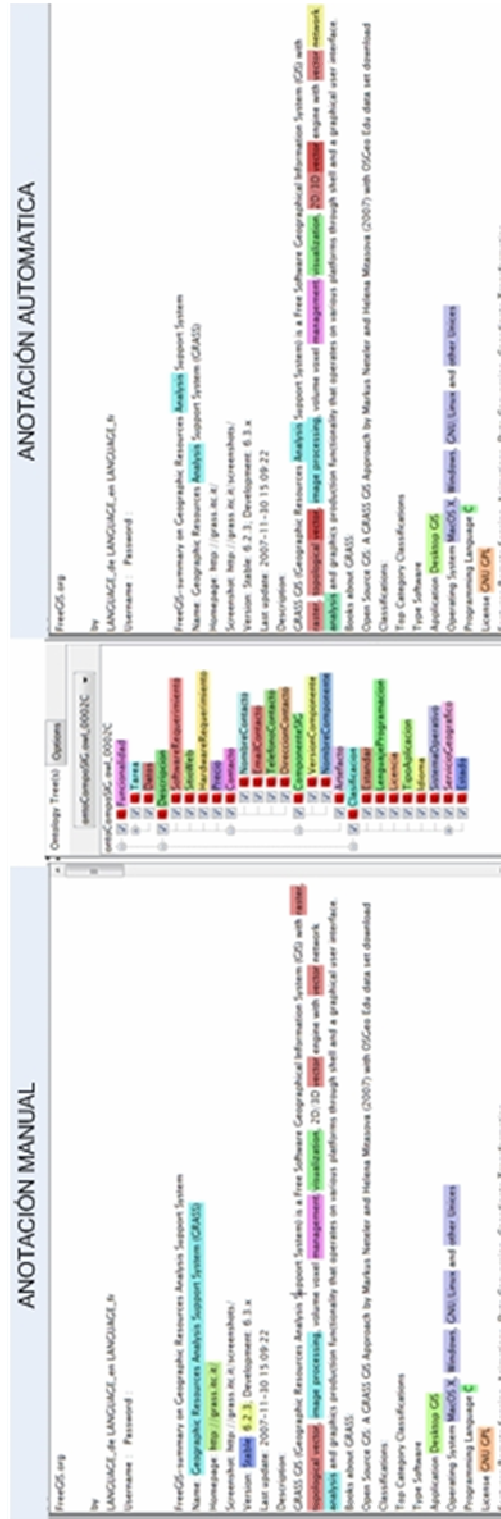


Figura 17. Anotación manual versus anotación automática.

La herramienta de evaluación *Annotation Diff* que ofrece GATE permite comparar un texto anotado automáticamente con un texto de referencia (anotado manualmente) y calcula las métricas Precisión, Cobertura y F-Measure. La Figura 18 muestra la ventana *Annotation Diff*. En esta ventana se selecciona el documento clave (*Key doc*), en este ejemplo: ‘MFreeGIS_org1.xml’, y el documento respuesta (*Resp. doc*), en este ejemplo: ‘FreeGIS_org1.htm_0002F’; y sus respectivos conjuntos de anotaciones (*Key set, Resp. set*): ‘SIG’ y ‘Default set’.

Start	End	Key	Features	=?	Start	End	Response	Features
6194	6200	raster	[class=http://www.se...oCompoSIG.ow#Datos]	=	6194	6200	raster	[ontology=file:/C:/U...os, majorType=datos]
985	992	Windows	[class=http://www.se...w#SistemaOperativo]	=	985	992	Windows	[ontology=file:/C:/U...pe=sistemaOperativo]
1882	1888	vector	[class=http://www.se...oCompoSIG.ow#Datos]	=	1882	1888	vector	[ontology=file:/C:/U...os, majorType=datos]
994	1003	GNU/Linux	[class=http://www.se...w#SistemaOperativo]	=	994	1003	GNU/Linux	[ontology=file:/C:/U...pe=sistemaOperativo]
3098	3111	visualization	[class=http://www.se...eraccionHumanaTarea]	=	3098	3111	visualization	[ontology=file:/C:/U...e=interaccionHumana]
3149	3160	vector data	[class=http://www.se...oCompoSIG.ow#Datos]	=	3149	3160	vector data	[ontology=file:/C:/U...os, majorType=datos]
578	584	vector	[class=http://www.se...oCompoSIG.ow#Datos]	=	578	584	vector	[ontology=file:/C:/U...os, majorType=datos]
6050	6061	raster data	[class=http://www.se...oCompoSIG.ow#Datos]	=	6050	6061	raster data	[ontology=file:/C:/U...os, majorType=datos]
5364	5380	network analysis	[ontology=http://gat...cionGeograficaTarea]	=	5364	5380	network analysis	[ontology=file:/C:/U...orType=comunicacion]
612	620	analysis	[class=http://www.se...entoGeograficoTarea]	=	612	620	analysis	[ontology=file:/C:/U...rType=procesamiento]
6073	6084	vector data	[class=http://www.se...oCompoSIG.ow#Datos]	=	6073	6084	vector data	[ontology=file:/C:/U...os, majorType=datos]
495	513	topological vector	[class=http://www.se...oCompoSIG.ow#Datos]	=	495	513	topological vector	[ontology=file:/C:/U...os, majorType=datos]
3138	3145	display	[class=http://www.se...eraccionHumanaTarea]	=	3138	3145	display	[ontology=file:/C:/U...rType=procesamiento]
5296	5314	topological vector	[ontology=http://gat...CompoSIG.ow#Datos]	=	5296	5314	topological vector	[ontology=file:/C:/U...os, majorType=datos]
150	202	Geographic Resources...port System: (GRASS)	[ontology=http://gat...entoGeograficoTarea]	~	171	179	Analysis	[ontology=file:/C:/U...rType=procesamiento]
213	233	http://grass.itc.it/	[class=http://www.se... rule=Emailaddress1]	-?				
296	301	6.2.3	[ontology=http://gat...IFVersionComponente]	-?				
288	294	Stable	[ontology=http://gat...CompoSIG.ow#Estado]	-?				
				7-	507	513	vector	[ontology=file:/C:/U...os, majorType=datos]
				7-	5185	5195	management	[ontology=file:/C:/U...stracionInformacion]
				7-	5408	5418	management	[ontology=file:/C:/U...stracionInformacion]
				7-	2973	2979	raster	[ontology=file:/C:/U...os, majorType=datos]

Correct: 27 Recall: 0,87 Precision: 0,35 F-measure: 0,50

Partially correct: 1 Strict: 0,87 0,35 0,50

Missing: 3 Lenient: 0,90 0,36 0,52

False positives: 49 Average: 0,89 0,36 0,51

80 pairings have been found (0 annotations are hidden)

Figura 18. Ventana *Annotation Diff*.

Todas las anotaciones del conjunto clave (*Key set*) son comparadas con las del conjunto respuesta (*Resp. Set*), y aquellas que tienen el mismo punto de comienzo y de final se muestran en la misma línea de la tabla. Luego, la

herramienta evalúa los atributos (*Features*) de cada anotación del conjunto clave y conjunto respuesta y determina si la anotación se considera: Correcta (ambas anotaciones coinciden totalmente), Parcialmente Correcta (ambas anotaciones coinciden parcialmente), Perdida (la anotación existe en el conjunto clave pero no en el conjunto respuesta), o Falsa (la anotación existe en el conjunto respuesta pero no en el conjunto clave).

Al final de la ventana de la herramienta se muestran los resultados de las métricas Precisión, Cobertura y F-Measure según tres criterios diferentes. La métrica estricta (*Strict*) considera todas las respuestas Parcialmente correctas como Incorrectas. La métrica leve (*Lenient*) considera todas las respuestas Parcialmente correctas como Correctas. La métrica promedio (*Average*) asigna un peso medio a las respuestas Parcialmente correctas.

Para evaluar los resultados, nosotros hemos considerado la métrica promedio. En el ejemplo, la herramienta *Annotation Diff* identifica 27 anotaciones Correctas, 1 Parcial, 3 Perdidas y 49 Falsas. Al aplicar las fórmulas para las métricas definidas en la Sección 4.2 , obtenemos los siguientes valores:

$$\text{Precisión} = (27 + \frac{1}{2} * 3) / (27 + 49 + 1) = 0,36$$

$$\text{Cobertura} = (27 + \frac{1}{2} * 3) / (27 + 3 + 1) = 0,89$$

$$\text{F-measure} = 2 (0,36 * 0,89) / (0,36 + 0,89) = 0,47$$

Estos pasos se ejecutan en tres iteraciones que se aplican a distintos corpus y persiguen distintos objetivos:

- **Iteración-1:** Durante esta iteración se pretende realizar una evaluación inicial de la performance del prototipo y determinar posibles ajustes para mejorar estos resultados iniciales. Se trabaja sobre el Corpus-1.
- **Iteración-2:** En base a los resultados de Precisión y Cobertura obtenidos, se examinan manualmente los documentos y se derivan e implementan nuevas

reglas para mejorar la performance del prototipo. Se trabaja sobre el mismo Corpus-1 (original, sin anotaciones).

- **Iteración-3:** Se pretende evaluar la performance del prototipo mejorado, pero aplicado a un conjunto de documentos distinto, se trabaja con el Corpus-2.

4.4. Análisis de Resultados

La primera parte de nuestro experimento incluye dos iteraciones en las que se ejecuta la aplicación ‘Extracción de Información’ sobre el mismo conjunto de pruebas (Corpus-1). Al finalizar la Iteración-1, las anotaciones automáticas se comparan con las anotaciones manuales y los resultados son expresados usando las métricas Precisión y Cobertura. La Tabla 10 muestra estos resultados, tanto para la Iteración-1 como para la Iteración-2, a nivel corpus.

Tabla 10. Mediciones de Precisión y Cobertura para la evaluación del prototipo.

	Iteración 1		Iteración 2		Diferencia	
	Cobertura	Precisión	Cobertura	Precisión	Cobertura	Precisión
<i>FreeGIS_org1.htm</i>	0,89	0,36	1	1	0,11	0,64
<i>FreeGIS_org2.htm</i>	0,42	0,64	0,9	1	0,48	0,36
<i>FreeGIS_org3.htm</i>	0,75	0,64	0,96	0,66	0,21	0,02
<i>FreeGIS_org4.htm</i>	0,26	0,76	1	0,92	0,74	0,16
<i>FreeGIS_org5.htm</i>	0,68	0,55	0,92	0,83	0,24	0,28
<i>FreeGIS_org6.htm</i>	1	0,67	1	0,88	0	0,21
<i>FreeGIS_org7.htm</i>	0,9	0,9	0,9	1	0	0,1
<i>FreeGIS_org8.htm</i>	0,81	0,74	0,92	0,8	0,11	0,06
<i>FreeGIS_org9.htm</i>	0,8	0,8	0,96	0,82	0,16	0,02
<i>FreeGIS_org10.htm</i>	0,49	0,81	0,97	0,81	0,48	0
Promedio	0,70	0,69	0,95	0,87	0,25	0,19

Si observamos los resultados de la Iteración-1, cuyos valores promedio para Cobertura y Precisión son 0,70 y 0,69 respectivamente, vemos que solo 6 documentos tienen ambos

valores mayores que 0,5. Al examinar manualmente los documentos procesados detectamos dos fuentes principales de errores:

- Errores de reglas gramaticales definidas en la *Identificación basada en patrones*: por ejemplo, en el documento ‘FreeGIS_org2.htm’ la anotación automática no reconoce al texto ‘http://www.remotesensing.org/proj’ como una entidad ‘sitioWeb’. Esto se debe que la regla original sólo reconoce texto que se ajuste a un patrón tipo: {“http://”} {palabra} {punto} {palabra} {punto} {palabra} {simbolo}, adecuado para un texto como: ‘http://www.misitio.org/’.
- Vocabulario del dominio utilizado en la *Identificación basada en listas* incompleto: por ejemplo, el texto ‘other Unices’ fue identificado en la anotación manual como una entidad ‘sistemaOperativo’, pero en la definición original del vocabulario que define la entidad no había sido considerado.

En la segunda iteración, luego de ajustar los errores identificados en la Iteración-1, comparamos los documentos del Corpus-1 anotados manualmente con los documentos anotados con el prototipo mejorado. En la Tabla 10 se observa que los valores de Cobertura y Precisión de la Iteración-2 son mayores o iguales a los de la Iteración-1, para todos los documentos.

Sin embargo, las mejoras se reflejan principalmente en la cobertura, y no tanto en la precisión. En promedio, la Cobertura de la Iteración-2 obtiene un valor de 0,95 y la Precisión de 0,87, mostrando una mejora promedio de 0,25 y 0,19 respectivamente para cada una de las métricas. Esto fue producto de que los ajustes para mejorar la Cobertura fueron pequeños cambios en las reglas definidas y en el vocabulario; pero, para mejorar la Precisión la tarea no es tan directa, requiriendo una modificación de las reglas y/o del vocabulario personalizado para que el prototipo no realice anotaciones donde no corresponde (generación de anotaciones falsas).

Después de optimizar el prototipo (en base a los resultados de la Iteración-1), y tal como se plantea en el proceso de evaluación descrito en la Sección 4.3, seleccionamos un nuevo corpus (Corpus-2) formado por 40 documentos que no se superponen con los del

Corpus-1. El experto realizó la anotación manual del Corpus-2, y luego se ejecutó el prototipo sobre este corpus para generar la anotación automática. Los resultados de ambas anotaciones se compararon usando la herramienta Corpus Quality Assurance que, a diferencia de la herramienta AnnotationDiff que calcula las métricas de precisión y cobertura a nivel del documento, calcula estas métricas a nivel del corpus. Los resultados obtenidos (Cobertura 0,94 y Precisión 0,85) se muestran en la Figura 19.

Corpus Quality Assurance

Corpus: Corpus2
 Key set: manual
 Response set: auto
 Types: [lookup]
 Features: []

Annotation	Match	Only A	Only B	Overlap	Rec.B/A	Prec.B/A	F1.0-a.
lookup	399	26	69	3	0.94	0.85	0.89
Macro summary					0.94	0.85	0.89
Micro summary	399	26	69	3	0.94	0.85	0.89

Figura 19. Cobertura y Precisión obtenidas por la aplicación ‘Extracción de Información’ sobre el Corpus-2.

NOTA: Tiempo anotación manual: 2’ 30’’ (aprox) x cada documento. Tiempo anotación automática (para 40 docs): 10’’

4.5. Discusión

En la sección anterior evaluamos la performance de este prototipo del Sistema de Extracción de información aplicando las métricas de precisión y cobertura; métricas que miden la calidad con respecto a las anotaciones realizadas. Sin embargo, ya que la información que genera el Sistema de Extracción de información luego será utilizada en el proceso de búsqueda de componentes que satisfagan las necesidades de los desarrolladores, también es necesario evaluar la calidad de la información recuperada por este sistema de extracción.

Una forma de evaluar la calidad de la información extraída es medir el porcentaje de categorías que logra completar el Sistema de Extracción en relación a la cantidad total de categorías del Esquema de clasificación normalizado (Sección 3.1). Para esto definimos la métrica de Completitud como:

$$\text{Completitud} = \text{Categorías completadas} / \text{Total de Categorías}$$

Para evaluar la completitud de las categorías del Esquema de clasificación normalizado, en la Tabla 11. presentamos un detalle de la información extraída por el Sistema de Extracción sobre los 10 documentos que conforman el Corpus-1.

Tabla 11. Categorías completadas por el Sistema de Extracción para el Corpus-1.

Categoría	Documentos									
	1	2	3	4	5	6	7	8	9	10
Nombre	√	√	√	√	√	√	√	√	√	√
Versión	√	√	√	√	√	√	√	√	√	√
Sitio web	√	√	√	√	√	√	√	√	√	√
Organización/Desarrollado	---	---	---	---	---	---	---	---	---	---
e-mail	---	---	---	---	---	---	---	---	---	---
Teléfono	---	---	---	---	---	---	---	---	---	---
Dirección postal	---	---	---	---	---	---	---	---	---	---
Precio	---	---	---	---	---	---	---	---	---	---
Artefacto	---	---	---	---	---	---	---	---	---	---
Req de software	---	---	---	---	---	---	---	---	---	---
Req de hardware	---	---	---	---	---	---	---	---	---	---
Tipo de aplicación	√	√	√	√	√	√	√	√	√	√
Sistema operativo	√	√	√	√	√	√	√	√	√	√
Leng programación	√	√	√	√	√	√	√	√	√	√
Estado	√	---	√	√	√	√	---	√	---	√
Licencia	√	√	√	√	√	√	√	√	√	√
Estándares	---	√	---	---	√	√	---	√	---	√
Idioma	---	---	---	---	---	---	---	---	---	---
Servicio geográfico	√	√	√	√	√	√	---	√	√	√
Tareas	√	√	√	√	√	√	---	√	√	√
Datos	√	√	√	√	√	√	---	√	√	√
<i>Total de cat.completadas</i>	<i>11</i>	<i>11</i>	<i>11</i>	<i>11</i>	<i>12</i>	<i>12</i>	<i>7</i>	<i>12</i>	<i>10</i>	<i>12</i>
% Completitud	52	52	52	52	57	57	33	57	48	57

Analizando estos resultados se puede destacar que para 7 de las categorías del Esquema de clasificación normalizado es posible obtener los datos correspondientes en el 100% de los documentos. Sin embargo, hay 9 categorías para los que se obtienen valores nulos

en todos los casos; esto nos arroja un porcentaje de completitud que varía alrededor del 50%.

Este resultado podría ser mejorado ajustando el conjunto de reglas de la ontología para extraer la información; sin embargo, el proceso no es tan sencillo: en general, las descripciones asociadas con los componentes en sitios web son generalmente cortas, y es posible que la información necesaria no se encuentre en el documento o que se encuentre parcialmente.

5. CONCLUSIONES

En este capítulo se analizan los resultados del trabajo llevado a cabo en esta tesis. De esta manera, los apartados incluidos son los siguientes: análisis de la consecución de objetivos, principales aportes de esta tesis, contraste de los mismos en publicaciones científicas y líneas de trabajo abiertas.

5.1. Análisis de la Consecución de Objetivos

En el primer capítulo de esta tesis se han presentado los objetivos parciales que se pretendían cumplir para satisfacer el objetivo principal de nuestra investigación, que es el siguiente:

Desarrollar un esquema normalizado para describir componentes para Sistemas de Información Geográficos (SIG) comerciales o libres, de manera que la información pueda ser estandarizada para el proceso de búsqueda.

A continuación se presenta una valoración de la consecución de cada uno de los objetivos parciales:

- Objetivo A: Proponer un esquema de clasificación para describir información funcional y no técnica de un componente SIG.

Hemos analizado propuestas para la clasificación de información de componentes seleccionando un marco conceptual como base para la definición de una ontología liviana. A continuación, detallamos los principales elementos del proceso centrado en la normalización de la información (esquema enriquecido semánticamente) y en la recuperación de información desde catálogos existentes usando técnicas del lenguaje natural.

-
- Objetivo B: Proponer técnicas de recolección y adecuación de la información almacenada en portales Web para componentes SIG.

Desarrollamos mecanismos para transformar las descripciones de componentes SIG que se encuentran en catálogos web, en descripciones normalizadas que usan el vocabulario definido en una ontología. El objetivo es identificar de forma automática, a partir de un documento escrito en lenguaje natural, la mayor cantidad posible de las categorías definidas en el esquema normalizado. La disponibilidad de estas descripciones semánticas de los componentes permitirá facilitar el posterior proceso de búsqueda y selección de componentes SIG.

- Objetivo C: Evaluar la propuesta con niveles de confianza de la información. Hemos desarrollado un caso de estudio y evaluado la precisión, cobertura y completitud de la información encontrada y almacenada en el esquema normalizado. Los primeros resultados, aunque promisorios, indican la necesidad de mayor investigación para el tratamiento de información incompleta, que es el caso típico de los catálogos de componentes en la Web.

5.2. Principales Aportaciones

Como principales aportes de esta Tesis se pueden mencionar los siguientes:

- La normalización de información de componentes SIG basada en estándares y en frameworks para clasificación y selección.
- Una técnica para categorizar información de componentes basada en información enriquecida semánticamente y mediante el uso de técnicas de procesamiento del lenguaje natural.
- Una herramienta que soporta el proceso de categorización de manera semi-automática.

5.3. Contrastación de Resultados

A continuación se detallan las publicaciones realizadas hasta el momento de resultados parciales de esta tesis. Las mismas se listan clasificadas según su tipo:

Conferencias Internacionales

- Gabriela Gaetán, Viviana Saldaño, Agustina Buccella, Alejandra Cechich. Artículo: A Domain-Oriented Approach for GIS Component Selection. En Proceedings de ICSEA 2010, 5th International Conference on Software Engineering Advances, en Nice, Francia, Agosto 2010. Editado por: Jon Hall, H. Kaindl, L. Lavazza, G. Buchgeher. O. Takaki. ISBN 978-0-7695-4144-0. Editorial: IEEE Computer Society. Páginas: 94-100.

Conferencias Nacionales

- G. Gaetán, V. Saldaño, A. Buccella, A. Cechich, A. Martin, S. Molina. Selección de Componentes para SIG a partir de Información en Servicios y Catálogos Web Normalizados. En Anales del XIII Workshop de Investigadores en Ciencias de la Computación, Rosario, Argentina, Año: 2011.
- G. Gaetán, V. Saldaño, A. Buccella, A. Cechich, A. Martin, S. Molina. Publicación y Selección de Componentes para SIG. En Anales del XII Workshop de Investigadores en Ciencias de la Computación, El Calafate, Argentina, Año: 2010.
- G. Gaetán, A. Cechich, A. Buccella. Artículo: Aplicación de Técnicas de Procesamiento de Lenguaje Natural y Web Semántica en la Publicación de Componentes para SIG. En Actas del ASSE (Simposio Argentino de Ingeniería de Software), JAIIO Editado por: María Paula González, Andrés Flores ISSN 1850-2792 Editorial: SADIO Lugar de edición: Argentina Páginas: 110—121 Año: 2009

-
- G. Gaetán, A. Cechich, A. Buccella. Artículo: Extracción de Información a partir de Catálogos Web de Componentes para SIG. En Actas del XV Congreso Argentino en Ciencias de la Computación Editado por: Elsa Estévez, Patricia Pesado, Alejandra Cechich ISBN 978-897-24068-4-1 Editorial: Red UNCI Lugar de edición: Argentina Páginas: 891--900 Año: 2009
 - G. Gaetán, V. Saldaño, A. Buccella, A. Cechich. Clasificación de Componentes SIG. En Anales del XI Workshop de Investigadores en Ciencias de la Computación, San Juan, Argentina, Año: 2009.
 - G. Gaetán, A. Cechich, A. Buccella, Artículo: Un Esquema de Clasificación Facetado para Publicación de Catálogos de Componentes SIG En: Actas XIV Congreso Argentino de Ciencias de la Computación ISBN 978-987-24611-0-2 Editorial: Red UNCI Lugar de edición: Argentina Páginas: 781—792 Año: 2008.
 - Gaetan, G.; Saldaño, V.; Buccella, A.; Cechich, A.; Mayorga, C.; Gelman, B.; Pejcich, G.; Diaz, M. Clasificación de componentes OTS (Off-The-Shell) para sistemas de información geográfica. En Anales del X Workshop de Investigadores en Ciencias de la Computación, WICC08. Gral. Pico, La Pampa. Año: 2008.

5.4. Trabajos Futuros

El trabajo presentado en esta tesis solo trata alguno de los problemas relacionados con la selección de componentes. Futuras investigaciones podrían extender los resultados obtenidos:

- Evaluar la aplicación en otro dominio de aplicación, distinto de los SIG.
- Explorar el uso de técnicas para gestionar la evolución del esquema de clasificación.
- Explorar el uso de técnicas para manejar información incompleta y/o inconsistente como fuente proveniente de catálogos en la Web.

-
- Mejorar la herramienta de soporte incluyendo nuevos indicadores.
 - Introducir el uso de la técnica en un proceso integral de selección de componentes. En este último caso, el trabajo realizado complementa la investigación propuesta en el contexto de los proyectos que enmarcan esta tesis. Los resultados alcanzados serán integrados a un proceso de demanda de servicios para componentes SIG de manera de construir una propuesta consistente para la búsqueda y selección de componentes en este dominio. Este proceso integral ha sido publicado en la conferencia internacional anteriormente citada.

6. REFERENCIAS

- [1] Ackermann, J., Brinkop, F., Conrad, S., Fettke, P., Frick, A., Glistau, E., Jaekel, H., Kotlar, O., Loos, P., Mrech, H., Ortner, E., Overhage, S., Raape, U., Sahm, S., Schmietendorf, A., Teschke, T., Turowski, K. Standardized Specification of Business Components. German Society of Informatics. 2002.
- [2] Albrecht, J. Semantic Net of Universal Elementary GIS Functions. Proceedings AUTO-CARTO 12 (Bethesda: American Congress on Surveying and Mapping). pp. 235-244. 1995.
- [3] Ayuso D., Boisen S., Fox H., Gish H., Ingria R., and Weischedel R. BBN: Description of the PLUM system as used for MUC-4. In Proceedings of the Fourth Message Understanding Conference (MUC-4), pages 169–176, 1992.
- [4] Basili, V.B., B., COTS-Based Systems Top 10 List. IEEE Computer Mayo 2001. 34(5): p. 91-93.
- [5] Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web. Scientific American. May 2001.
- [6] Bertoa M. F., Troya J., Vallecilo A. A Survey on the Quality Information Provided by Software Component Vendors. Proc. Of the 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2003). Germany. 2003.
- [7] Borgo, S.; Guarino, N.; Masolo, C.; Vetere, G. Using a large linguistic ontology for internet-based retrieval of object-oriented components. Ninth International Conference on Software Engineering and Knowledge Engineering. Madrid, Spain. 1997.
- [8] Broekstra, J., Kampman, A. and Harmelen, F.V. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. International Semantic Web Conference. Sardinia, Italy. 2002.
- [9] Carney D., L.F., What Do You Mean by COTS? Finally a Useful Answer. IEEE Software, March/April 2000. 17 (2).

-
- [10] Cechich A., Réquilé A., Aguirre J., Luzuriaga J. Trends on COTS Component Identification. 5th International Conference on COTS-Based Software Systems. Orlando, USA. IEEE Computer Science Press. 2006.
- [11] Chaudhri VK, Farquhar A, Fikes R, Karp PD, Rice JP (1998) Open Knowledge Base Connectivity 2.0.3. Technical Report. <http://www.ai.sri.com/~okbc/okbc-2-0-3.pdf>
- [12] ComponentSource – The Definitive Source of Software Components. Available: <http://www.componentsource.com/>. [Accessed: May 24, 2010].
- [13] Corcho O., Fernández M. & Gómez A. Methodologies, tools and languages for building ontologies. Where is their meeting point? Data Knowledge Eng. Amsterdam. The Netherlands, 2003.
- [14] Crnkovic I., Larsson M., Building Component-Based Reliable Software Systems, Artech House, 2002.
- [15] Cunningham, H., Maynard, D., Bontcheva, K. y Tablan, V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Application. 40th Anniversary Meeting of the Association for Computational Linguistics. 2002.
- [16] Dill, S., Gibson, N., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.A. y Zien, J.Y. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. Twelfth International World Wide Web Conference. Budapest, Hungary. 178-186. 2003.
- [17] Dingli, A., Ciravegna, F., Wilks, Y. Automatic Semantic Annotation using Unsupervised Information Extraction and Integration. K-CAP 2003 Workshop on Knowledge Markup and Semantic Annotation. 2003.
- [18] Dong J., Alencar P. S. C., Cowan D. D. A Component Specification Template for COTS-based Software Development. First Workshop on Ensuring Successful COTS Development. 1999.
- [19] ESPRIT/ESSI Project no 21580. Guidelines for Best Practice in User Interface for GIS. Section 6 “List of key GIS operations”. European Commission.
-

-
- [20] ESRI – The GIS Software Leader – Available: <http://www.esri.com/> .
[Accessed: May 24, 2010]
- [21] Farquhar A, Fikes R, Rice J (1997) The Ontolingua Server: A Tool for Collaborative Ontology Construction. *International Journal of Human Computer Studies* 46(6):707–727
- [22] Fellbaum, C. WORDNET: An Electronic Lexical Database. The MIT Press. 1998.
- [23] Fernández-López M, Gómez-Pérez A, Juristo N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Spring Symposium on Ontological Engineering of AAAI. Stanford University, California. pp 33–40. 1997
- [24] Frakes W. and Baeza-Yates R., editors. Information retrieval, data structures and algorithms. Prentice Hall, New York, Englewood Cliffs, N.J., 1992.
- [25] FreeGIS.org – Available: <http://www.freegis.org/> . [Accessed: May 24, 2010].
- [26] Freshmeat.net – Available: <http://www.freshmeat.net/>. [Accessed: May 24, 2010].
- [27] Geographic information. Geographic Information Metadata. Draft International standard 19115. ISO/IEC. 2003.
- [28] Geographic information. Geographic Information Methodology for Feature Cataloguing. Draft International standard 19110. ISO/IEC. 2005.
- [29] Geographic information. Rules for Application Schema. Draft International Standard 19109. ISO/IEC. 2005.
- [30] Geographic Information. Services. Draft International standard 19119. ISO/IEC. 2002.
- [31] Geographic information. Spatial Schema. International standard 19107. ISO/IEC. 2003.
- [32] Girardi, M. R.; Ibrahim, B.: A software reuse system based on natural language specifications. 5th International Conference on Computing and Information. Ontario (1993)
-

-
- [33] Gruber, T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human and Computer Studies*. Vol.43. nos. 5/6. pp. 907-928. 1995.
- [34] Guarino N. *Formal Ontology and Information Systems. Formal Ontology in Information Systems*. - Amsterdam, Netherlands. : Formal Ontology in Information Systems, 1998.
- [35] Handschuh S., Staab S., y Maedche A. CREAM Creating relational metadata with a component-based, ontology driven framework. *Proceedings of K-Cap 2001*. Victoria, BC, Canada. 2001.
- [36] Handschuh, S., Staab, S. and Ciravogna, F. S-CREAM -- Semi-automatic CREATION of Metadata. *Semantic Authoring, Annotation & Knowledge Markup - Preliminary Workshop Programme*. 2002.
- [37] Heineman and W. Council. *Component-Based Software Engineering - Putting the Pieces Together*. Addison-Wesley, 2001.
- [38] Henninger, S. An Evolutionary Approach to Constructing Effective Software Reuse Repositories. *ACM Transactions on Software Engineering and Methodology*, Vol. 06, No. 02. (1997)
- [39] Hobbs J. R., Appelt D., Tyson M., Bear J., and Israel D. SRI International: Description of the FASTUS system used for MUC-4. In *Proceedings fo the 4th Message Understanding Conference (MUC-4)*, pages 268–275, 1992.
- [40] Hobbs, J. The generic information extraction system. *Proceedings of the 5th Message Understanding Conference*. Morgan Kaufmann. pp. 87-92. 1993.
- [41] Iribarne, L. *Un Modelo de Mediación para el Desarrollo de Software Basado en Componentes COTS*. Phd thesis, Univerdad de Almería. (2003)
- [42] Iribarne, L., Troya, J. M., Vallecillo, A.: *Trading for COTS Components in Open Environments*. 27th Euromicro Conference. IEEE Computer Society Press. Warsaw (2001)

-
- [43] ISO. Information Technology – Software Product evaluation – Quality Characteristics and Guidelines for their Use. Int. Standard ISO/IEC 9126, ISO, 1991.
- [44] Kahan J., Koivunen M., Prud'Hommeaux E., Swick R. Annotea: An Open RDF Infrastructure for Shared Web Annotations. The WWW10 Conference. Hong Kong. pp. 623-632. 2001.
- [45] Kaiser, K., and Silvia M. “Information Extraction A Survey.” Technology May 2005.
- [46] Kallio, P., Niemelä, E.: Documented Quality of COTS and OCM Components. 4th ICSE Workshop on Component-Based Software Engineering (2001)
- [47] Kiryakov A., Popov B., Kirilov A., Manov D., Ognyanoff D., Goranov M.: Semantic Annotation, Indexing, and Retrieval. 2nd International Semantic Web Conference. Florida, USA. (2003)
- [48] Kogut, P.; Holmes, W. AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. First International Conference on Knowledge Capture. 2001.
- [49] Lucena, V. F. Flexible Web-based Management of Components for Industrial Automation. Phd thesis, Stuttgart University. (2002)
- [50] Maguire D. J., Dangermond J. The functionality of GIS, in Geographical Information Systems : Principles and Application. edited by D. J. Maguire, M. F. Goodchild and D. Rhind, (Harlows : Longmans), vol1. 1991.
- [51] Maynard, D., Li, Y., Peters, W.: Metrics for evaluation of ontology-based information extraction. In WWW 2006 Workshop on “Evaluation of Ontologies for the Web” (EON), Edinburgh, Scotland, 2006.
- [52] Nirenburg, S., Raskin, V. Ontological Semantics, Formal Ontology, and Ambiguity. Formal Ontology in Information Systems (FOIS). pp. 151-161. New York, NY: ACM. 2001.

-
- [53] Oberndorf, P., Facilitating Component-Based Software Engineering: COTS and Open Systems. Proceedings of the Fifth International Symposium on Assessment of Software Tools- SAST.97, 1997.
- [54] Overhage S. Towards a Standardized Specification Framework for Component Development, Discovery, and Configuration. Proceedings of the Eighth International Workshop on Component-Oriented Programming (WCOP). Bosch, J., Szyperski, C., Weck, W. (Eds.). 2003.
- [55] OWL W3C submission, <http://www.w3.org/Submission/OWL/>.
- [56] Palmer D. and Day D. A statistical profile of the named entity task. In ANLP, pages 190–193, 1997.
- [57] Papazoglou, M.; Van den Heuvel, W. Service Oriented Architectures: Approaches, Technologies and Research Issues. VLDB Journal. 2007.
- [58] Papazoglou, M.; Van den Heuvel, W. Service-Oriented Design and Development Methodology. International Journal of Web Engineering and Technology. 2006.
- [59] Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. y Goranov, M. KIM – Semantic Annotation Platform. 2nd International Semantic Web Conference. Florida, USA. 834- 849. 2003.
- [60] Prieto-Díaz, R., Freeman, P.: Classifying Software for Reusability. IEEE Software. 4(1):6-16 (1987)
- [61] Santana de Almeida, E., Lucrédio, D., F. d. Prado, A. A Survey on Software Components Search and Retrieval. In 30th IEEE EUROMICRO Conference, Component-Based Software Engineering Track. Rennes, France. (2004)
- [62] Seacord, R. C., Hissam, S. A., Wallnau, and K. C. Agora: A Search Engine for Software Components. Technical report, CMU/SEI - Carnegie Mellon University/Software Engineering Institute. (1998)
- [63] Sugumaran, V.; Storey, V.C.: A semantic-based approach to component retrieval. SIGMIS Database. 34(3): p. 8-24 (2003)
-

-
- [64] Szyperski, C. Component Software – Beyond Object-Oriented Programming. Addison-Wesley. 2002.
- [65] Torchiano, M., Jaccheri, L., Sørensen, C., Wang, I.: COTS Products Characterization. 14th international conference on Software engineering and knowledge engineering. Ischia (2002)
- [66] Torchiano, M., Morisio M., Overlooked Aspects of COTS-based Development. IEEE Software, March/April 2004.
- [67] Traas, V.; Hillegersberg, J. V. The software component market on the internet current status and conditions for growth. ACM SIGSOFT Software Engineering Notes, Vol. 25 No. 01. January, 2000.
- [68] UDDI Organization (ed.). UDDI Executive White Paper. UDDI Standards Organization Public Draft. 2001. Disponible en: http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf
- [69] Varadarajan, S.; Kumar, A.; Gupta, D.; Jalote, P. ComponentXchange: An E-Exchange For Software Components. Proc. IADIS Conf. WWW/Internet. Portugal. 2001.
- [70] Vargas-Vera M., Motta E., Domingue J., Lanzoni M., Stutt A. y Ciravegna F. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. Proc. Of EKAW 2002. ed. Gomez-Perez, A., Springer Verlag. 2002.
- [71] Vigder, M., Gentleman, M. , Dean, J., COTS Software Integration: State of the Art. 1996.
- [72] Wallnau S., Hissam R., Seacord. Building Systems from Commercial Components. Addison-Wesley. 2002.
- [73] Wimalasuriya, D. C. and Dou, D. Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science, 36(3):306–323, 2010.
- [74] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., y Wesslén, A. Experimentation in Software Engineering: An Introduction. The Kluwer International Series in Software Engineering. Kluwer Academic Publishers. 2000.
-

-
- [75] Yaguinuma, C.; Santos, M.; Vieira, M. Ontology-Based Meta-model for Storage and Retrieval of Software Components.
- [76] Yangarber, R., Grishman R., Tapanainen P., and Hutunen S., 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In 18th COLING Conference. Germany.
- [77] Ye, Y.; Fischer, G. Supporting Reuse By Delivering Task-Relevant and Personalized Information. In: ICSE 2002 - 24th International Conference on Software Engineering, 2002, Orlando, Florida, USA.
- [78] Zelkowitz, M. V., Wallace, D., y Binkley, D. Experimental validation of new software technology. En N. Juristo y A. Moreno (Eds.), Lecture Notes on Empirical Software Engineering, volume 12 of Series on Software Engineering and Knowledge Engineering pp. 229-263. World Scientific. 2003.

Apéndice A

Descripción anotada semánticamente del componente *GeographicTranslator*

```
<?xml version="1.0" encoding="WINDOWS-1252"?>
-<GateDocument>
  <!-- The document's Features-->
  +<GateDocumentFeatures>--
    <!-- The document content area with serialized nodes -->
  -<TextWithNodes>
    FreeGIS.org by LANGUAGE_de LANGUAGE_en LANGUAGE_fr UserName :
    Password : FreeGIS-summary on Geographic Translator Name:
    <Node Id="127"/>
    Geographic Translator
    <Node Id="148"/>
    (GEOTRANS) Homepage:
    <Node Id="170"/>
    http://
    <Node Id="177"/>
    earth
    <Node Id="182"/>-
    info.nga.mil/GandG/geotrans/
    <Node Id="211"/>
    Screenshot: Version:
    <Node Id="233"/>
    2.2.5
    <Node Id="238"/>
    Last update: 2005-06-16 23:52:40 Description: GEOTRANS (Geographic
    Translator) is an application program which allows users to easily
    <Node Id="373"/>
    convert
    <Node Id="380"/>
    <Node Id="381"/>
    geographic coordinates
    <Node Id="403"/>
    among a wIldE variety of coordinate systems,
    <Node Id="448"/>
    map
    <Node Id="451"/>
    <Node Id="452"/>
    projections
    <Node Id="463"/>
    , and
    <Node Id="469"/>
    Datums
    <Node Id="475"/>
    . GEOTRANS runs in both Microsoft
    <Node Id="509"/>
    Windows
    <Node Id="516"/>
    (95/98/NT) and UNIX Motif environments. Archives of several version might
    additionally be found in Remotesensing.org's geotrans directory . Classifications:
    Top Category Classifications Type Software Operating System
    <Node Id="734"/>
    Windows
    <Node Id="741"/>
```

, GNU/Linux and other Unices Programming Language
 <Node Id="791"/>
C
 <Node Id="792"/>
 <Node Id="793"/>
Java
 <Node Id="797"/>
License
 <Node Id="806"/>
GEOTRANS License
 <Node Id="822"/>
Features Coordinate Transformation Links: No links have been defined. Changes: 2005-06-16 23:52:40 (jan) Fixed Homepage. 2004-10-20 21:17:55 (jan) Update from 2.2.4 to 2.2.5. 2004-08-19 10:27:38 (jan) Updated from 2.2.3 to 2.2.4. New homepage. Previous was: <http://164.214.2.59/GandG/geotrans/geotrans.html> 2003-02-28 17:13:10 (jan) Updates from 2.2.1 to 2.2.3. 2002-04-01 15:21:05 (bernhard) NIMA seems to have released a new version last year. Link changed to NIMA page. Indicated that remotesensing.org might have a copy or older versions archives. v2.0.1->2.2.1, as gathered from the Changelog : Many bugfixes Support for GNU g++. Shared library support. Seven new
 <Node Id="1494"/>
projections
 <Node Id="1505"/>
 . (Azimuthal Equidistant, (Oblique) Gnomonic,
 <Node Id="1551"/>
Oblique
 <Node Id="1558"/>
 Mercator, Ney's (Modified Lambert Conformal Conic), Stereographic, British National Grid, and New Zealand Map Grid) Addition of a
 <Node Id="1689"/>
Java
 <Node Id="1693"/>
 GUI. Inverse flattening added as primary ellipsoid parameter. Homepage Hall of Fame FreeGIS
 <Node Id="1790"/>
Database
 <Node Id="1798"/>
 Software (351) Geo-Data (25) Documents (19) Projects (10) General Mailing List: Archive | subscribe Core-Team Mailing List: Archive | subscribe Updates Mailing List: Archive | subscribe About FreeGIS project site design next steps Sponsorship Public honoring? Yes No Community Contribution Proposals FreeGIS CD FreeGIS Tutorial Exits Other <http://remotesensing.org/> <http://maptools.org/> <http://opensourcegis.org/> only for contributors: Administration categories objects FreeGIS is a project of FOSSGIS e.V. Contact: Björn Schilberg <mailto:bjoern.schilberg@uni-osnabrueck.de> These pages are hosted by Intevation GmbH

```

</TextWithNodes>
  <!-- The default Annotation Set -->
  -<AnnotationSet>
    -<Annotation EndNode="1693" StartNode="1689" Type="LenguajeProgramacion"
      Id="5248">
      -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
          className="java.lang.String">http://www.semanticweb.org/ontologies/2010/10/OntoCompoSIG.owl#LenguajeProgramacion</Value>
        </Feature>
      -<Feature>

```

```

        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Lenguaje</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="822" StartNode="806" Type="Licencia" Id="5249">
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Licencia</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="463" StartNode="452" Type="Lookup" Id="5230">
    -<Feature>
        <Name className="java.lang.String">ontology</Name>
        <Value
        className="java.lang.String">file:/C:/Users/Gaby/Desktop/ontoCo
        mpoSIG.owl</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
        className="java.lang.String">ProcesamientoGeograficoTarea</Val
        ue>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">majorType</Name>
        <Value className="java.lang.String">procesamiento</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="148" StartNode="127" Type="nombreComponente"
Id="5250">
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">nombreId</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="516" StartNode="509" Type="SistemaOperativo" Id="5251">
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
        className="java.lang.String">http://www.semanticweb.org/ontolog
        ies/2010/10/OntoCompoSIG.owl#SistemaOperativo</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Sistema</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="451" StartNode="448" Type="Lookup" Id="5228">
    -<Feature>
        <Name className="java.lang.String">ontology</Name>
        <Value
        className="java.lang.String">file:/C:/Users/Gaby/Desktop/ontoCo
        mpoSIG.owl</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value className="java.lang.String">Datos</Value>
    </Feature>

```

```

    -<Feature>
      <Name className="java.lang.String">majorType</Name>
      <Value className="java.lang.String">datos</Value>
    </Feature>
  </Annotation>
  -<Annotation EndNode="741" StartNode="734" Type="SistemaOperativo" Id="5252">
    -<Feature>
      <Name className="java.lang.String">class</Name>
      <Value
        className="java.lang.String">http://www.semanticweb.org/ontolog
        ies/2010/10/OntoCompoSIG.owl#SistemaOperativo</Value>
      </Feature>
    -<Feature>
      <Name className="java.lang.String">rule</Name>
      <Value className="java.lang.String">Sistema</Value>
    </Feature>
  </Annotation>
  -<Annotation EndNode="211" StartNode="170" Type="sitioWeb" Id="5254">
    -<Feature>
      <Name className="java.lang.String">class</Name>
      <Value
        className="java.lang.String">http://www.semanticweb.org/ontolog
        ies/2010/10/OntoCompoSIG.owl#SitioWeb</Value>
      </Feature>
    -<Feature>
      <Name className="java.lang.String">rule</Name>
      <Value className="java.lang.String">Emailaddress1</Value>
    </Feature>
  </Annotation>
  -<Annotation EndNode="380" StartNode="373" Type="TareaProcesamiento"
  Id="5255">
    -<Feature>
      <Name className="java.lang.String">class</Name>
      <Value
        className="java.lang.String">http://www.semanticweb.org/ontolog
        ies/2010/10/OntoCompoSIG.owl#ProcesamientoGeograficoTarea</
        Value>
      </Feature>
    -<Feature>
      <Name className="java.lang.String">rule</Name>
      <Value className="java.lang.String">Tarea</Value>
    </Feature>
  </Annotation>
  -<Annotation EndNode="463" StartNode="452" Type="TareaProcesamiento"
  Id="5256">
    -<Feature>
      <Name className="java.lang.String">class</Name>
      <Value
        className="java.lang.String">http://www.semanticweb.org/ontolog
        ies/2010/10/OntoCompoSIG.owl#ProcesamientoGeograficoTarea</
        Value>
      </Feature>
    -<Feature>
      <Name className="java.lang.String">rule</Name>
      <Value className="java.lang.String">Tarea</Value>
    </Feature>
  </Annotation>

```

```

    -<Annotation EndNode="1505" StartNode="1494" Type="TareaProcesamiento"
    Id="5257">
      -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
          className="java.lang.String">http://www.semanticweb.org/ontolog
          ies/2010/10/OntoCompoSIG.owl#ProcesamientoGeograficoTarea</
          Value>
        </Feature>
      -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Tarea</Value>
      </Feature>
    </Annotation>
    -<Annotation EndNode="238" StartNode="233" Type="versionComponente"
    Id="5258">
      -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Version</Value>
      </Feature>
    </Annotation>
    <Annotation EndNode="1558" StartNode="1551" Type="SistemaOperativo"
    Id="5259">
    </Annotation>
    -<Annotation EndNode="792" StartNode="791" Type="LenguajeProgramacion"
    Id="5246">
      -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
          className="java.lang.String">http://www.semanticweb.org/ontolog
          ies/2010/10/OntoCompoSIG.owl#LenguajeProgramacion</Value>
        </Feature>
      -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Lenguaje</Value>
      </Feature>
    </Annotation>
    -<Annotation EndNode="797" StartNode="793" Type="LenguajeProgramacion"
    Id="5247">
      -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
          className="java.lang.String">http://www.semanticweb.org/ontolog
          ies/2010/10/OntoCompoSIG.owl#LenguajeProgramacion</Value>
        </Feature>
      -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Lenguaje</Value>
      </Feature>
    </Annotation>
    -<Annotation EndNode="463" StartNode="448" Type="Dato" Id="5244">
      -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
          className="java.lang.String">http://www.semanticweb.org/ontolog
          ies/2010/10/OntoCompoSIG.owl#Datos</Value>
        </Feature>
      -<Feature>

```

```

        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Dato</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="475" StartNode="469" Type="Dato" Id="5245">
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
            className="java.lang.String">http://www.semanticweb.org/ontolog
            ies/2010/10/OntoCompoSIG.owl#Datos</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Dato</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="182" StartNode="177" Type="Dato" Id="5242">
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
            className="java.lang.String">http://www.semanticweb.org/ontolog
            ies/2010/10/OntoCompoSIG.owl#Datos</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Dato</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="403" StartNode="381" Type="Dato" Id="5243">
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
            className="java.lang.String">http://www.semanticweb.org/ontolog
            ies/2010/10/OntoCompoSIG.owl#Datos</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Dato</Value>
    </Feature>
</Annotation>
-<Annotation EndNode="1798" StartNode="1790" Type="TipoAplicacion" Id="5241">
    -<Feature>
        <Name className="java.lang.String">class</Name>
        <Value
            className="java.lang.String">http://www.semanticweb.org/ontolog
            ies/2010/10/OntoCompoSIG.owl#TipoAplicacion</Value>
    </Feature>
    -<Feature>
        <Name className="java.lang.String">rule</Name>
        <Value className="java.lang.String">Aplicacion</Value>
    </Feature>
</Annotation>
</AnnotationSet>
</GateDocument>

```

Apéndice B

Descripción de la ontología que describe la estructura del esquema normalizado de clasificación de componentes SIG.

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY OntologyComopoSIG "http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#" >
  <!ENTITY ISO_geographic_service "http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#" >
]>

<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.w3.org/2002/07/owl"
  xmlns:ISO_geographic_service="http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#"
  xmlns:OntologyComopoSIG="http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <Ontology rdf:about="http://www.semanticweb.org/ontologies/2010/10/OntoCompoSIG.owl"/>
  <!--
  //
  //
  // Annotation properties
  //
  -->
  <AnnotationProperty rdf:about="&rdfs:label"/>
  <AnnotationProperty rdf:about="&rdfs:comment"/>
  <!--
  //
  // Object Properties
  //
  //
  -->
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#esRealizadaPor -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;esRealizadaPor">
    <rdfs:range rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:domain rdf:resource="&OntologyComopoSIG;Tarea"/>
  </ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#realizaTarea -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;realizaTarea">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&OntologyComopoSIG;Tarea"/>
    <inverseOf rdf:resource="&OntologyComopoSIG;esRealizadaPor"/>
  </ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneEntrada -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;tieneEntrada">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&OntologyComopoSIG;Datos"/>
  </ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneInformacion -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;tieneInformacion">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&OntologyComopoSIG;Contacto"/>
  </ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneLicencia -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;tieneLicencia">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&OntologyComopoSIG;Licencia"/>
  </ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneRequerimiento -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;tieneRequerimiento">
```

```

    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&OntologyComopoSIG;Requerimiento"/>
  </ObjectProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneSalida -->
  <ObjectProperty rdf:about="&OntologyComopoSIG;tieneSalida">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&OntologyComopoSIG;Datos"/>
    <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  </ObjectProperty>
  <!-- http://www.w3.org/2002/07/owl#topObjectProperty -->
  <ObjectProperty rdf:about="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  <!--
  ////////////////////////////////////////////////////////////////////
  //
  // Data properties
  //
  ////////////////////////////////////////////////////////////////////
  -->
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#escritoEnLenguajeProgramacion -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;escritoEnLenguajeProgramacion">
    <rdfs:comment rdf:datatype="&xsd:string">C, Java, C++, Python, Tcl/Tk, PHP, Perl, (Visual) Basic, C#, Fortran, Delphi,
    Ruby, SWF, Javascript, Otros</rdfs:comment>
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#requiereCapacidadDisco -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;requiereCapacidadDisco">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;HardwareRequerimiento"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#requiereCapacidadMemoria -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;requiereCapacidadMemoria">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;HardwareRequerimiento"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#requiereProcesador -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;requiereProcesador">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;HardwareRequerimiento"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#soportaEstandar -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;soportaEstandar">
    <rdfs:comment rdf:datatype="&xsd:string">OGC WMS, OGC WFS</rdfs:comment>
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#soportaSistemaOperativo -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;soportaSistemaOperativo">
    <rdfs:comment rdf:datatype="&xsd:string">GNU/Linux, otros Unices, Windows, MacOS X, Otros</rdfs:comment>
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneAplicacion -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;tieneAplicacion">
    <rdfs:comment rdf:datatype="&xsd:string">Escritorio, MÃ³vil, Web, Base de Datos, LibrerÃa</rdfs:comment>
    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneDireccionPostal -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;tieneDireccionPostal">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;Contacto"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneEmail -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;tieneEmail">
    <rdfs:domain rdf:resource="&OntologyComopoSIG;Contacto"/>
    <rdfs:range rdf:resource="&xsd:string"/>
  </DatatypeProperty>
  <!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneEstadoDesarrollo -->
  <DatatypeProperty rdf:about="&OntologyComopoSIG;tieneEstadoDesarrollo">
    <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
    <rdfs:comment rdf:datatype="&xsd:string">estable, alfa, beta, archivado</rdfs:comment>

```

```

    <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</DatatypeProperty>
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneIdioma -->
<DatatypeProperty rdf:about="&OntologyComopoSIG;tieneIdioma">
  <rdfs:comment rdf:datatype="&xsd:string">espaÃ±ol, alemÃ¡n, francÃ©s</rdfs:comment>
  <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</DatatypeProperty>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneNombre -->
<DatatypeProperty rdf:about="&OntologyComopoSIG;tieneNombre">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="&xsd:string"/>
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="&OntologyComopoSIG;ComponenteSIG"/>
        <rdf:Description rdf:about="&OntologyComopoSIG;Contacto"/>
        <rdf:Description rdf:about="&OntologyComopoSIG;SoftwareRequerimiento"/>
      </unionOf>
    </Class>
  </rdfs:domain>
</DatatypeProperty>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tienePrecio -->
<DatatypeProperty rdf:about="&OntologyComopoSIG;tienePrecio">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="&OntologyComopoSIG;ComercialLicencia"/>
  <rdfs:range rdf:resource="&xsd:decimal"/>
</DatatypeProperty>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneSitioWeb -->
<DatatypeProperty rdf:about="&OntologyComopoSIG;tieneSitioWeb">
  <rdfs:domain rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</DatatypeProperty>
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneTelefono -->

<DatatypeProperty rdf:about="&OntologyComopoSIG;tieneTelefono">
  <rdfs:domain rdf:resource="&OntologyComopoSIG;Contacto"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</DatatypeProperty>
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneTipoLicencia -->
<DatatypeProperty rdf:about="&OntologyComopoSIG;tieneTipoLicencia">
  <rdfs:comment rdf:datatype="&xsd:string">GNU GPL, GNU LGPL, MIT, Public Domain, Creative
Commons</rdfs:comment>
  <rdfs:domain rdf:resource="&OntologyComopoSIG;NoComercialLicencia"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</DatatypeProperty>
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#tieneVersion -->

<DatatypeProperty rdf:about="&OntologyComopoSIG;tieneVersion">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="&xsd:string"/>
  <rdfs:domain>
    <Class>
      <unionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="&OntologyComopoSIG;ComponenteSIG"/>
        <rdf:Description rdf:about="&OntologyComopoSIG;SoftwareRequerimiento"/>
      </unionOf>
    </Class>
  </rdfs:domain>
</DatatypeProperty>

<!--
////////////////////////////////////
//
// Classes
//

```

```

////////////////////////////////////
-->

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Catalogue_Service -->

<Class rdf:about="&ISO_geographic_service;Catalogue_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides discovery and management services on a store of metadata about
instances.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Catalogue_Viewer -->

<Class rdf:about="&ISO_geographic_service;Catalogue_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Human_Interaction_Service"/>
  <rdfs:comment xml:lang="en">Allows a user to interact with a catalogue to locate, browse, and manage metadata about
geographic data or geographic services.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Chain_Definition_Editor -->

<Class rdf:about="&ISO_geographic_service;Chain_Definition_Editor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Human_Interaction_Service"/>
  <rdfs:comment xml:lang="en">Provides user interaction with a chain definition service</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Chain_Definition_Service -->

<Class rdf:about="&ISO_geographic_service;Chain_Definition_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Workflow-Task_Management_Service"/>
  <rdfs:comment xml:lang="en">Service to define a chain and to enable it to be executed by the workflow enactment
service.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Change_Detection_Service -->

<Class rdf:about="&ISO_geographic_service;Change_Detection_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to find differences between two data sets that represent the same geographical area at
different times.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Coordinate_Conversion_Service -->

<Class rdf:about="&ISO_geographic_service;Coordinate_Conversion_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to change coordinates from one coordinate system to another coordinate system that is
related to the same datum.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Coordinate_Transformation_Service -->

<Class rdf:about="&ISO_geographic_service;Coordinate_Transformation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to change coordinates from a coordinate reference system based on datum to a
coordinate reference system based on a second datum.</rdfs:comment>
</Class>

```

```

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Coverage-Vector_Conversion_Service -->

<Class rdf:about="&ISO_geographic_service;Coverage-Vector_Conversion_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to change the spatial representation from a coverage schema to a vector
schema.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Coverage_Access_Service -->

<Class rdf:about="&ISO_geographic_service;Coverage_Access_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides a client access to and management of a coverage store.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Coverage_Access_Service-Sensor -->

<Class rdf:about="&ISO_geographic_service;Coverage_Access_Service-Sensor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Coverage_Access_Service"/>
  <rdfs:comment xml:lang="en">Provide access to a coverage where the source of the coverage data is a real-time
sensor.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Dimension_Measurement_Service -->

<Class rdf:about="&ISO_geographic_service;Dimension_Measurement_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to compute dimension of objects visible in an image or other geodata.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Encoding_Service -->

<Class rdf:about="&ISO_geographic_service;Encoding_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Communication_Service"/>
  <rdfs:comment xml:lang="en">Service that provides implementation of an encoding rule and provides an interface to encoding
and decoding functionality.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Feature_Access_Service -->

<Class rdf:about="&ISO_geographic_service;Feature_Access_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides a client access to and management of a feature store.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Feature_Generalization_Service -->

<Class rdf:about="&ISO_geographic_service;Feature_Generalization_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that generalizes feature types in a feature collection to increase the effectiveness of
communication by counteracting the undesirable effects of data reduction.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Feature_Gengeralization_Editor -->

```

```

<Class rdf:about="&ISO_geographic_service;Feature_Generalization_Editor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Editor"/>
  <rdfs:comment xml:lang="en">Allows a user to modify the cartographic characteristics of a feature or feature collection by
simplifying its visualization, while maintaining its salient elements - the spatial equivalent of simplification.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Feature_Manipulation_Service -->

<Class rdf:about="&ISO_geographic_service;Feature_Manipulation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Register one feature to another, an image, or another data set or coordinateset: correcting for
relative translation shifts, rotational differences, scale differences, and perspective differences. Verify that all features in the Feature
Collection are topologically consistent according to the topology rules of the Feature Collection, and identifies and/or corrects any
inconsistencies that are discovered.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Feature_Type_Service -->

<Class rdf:about="&ISO_geographic_service;Feature_Type_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides a client to access to and management of a store of feature type.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Gazetteer_Service -->

<Class rdf:about="&ISO_geographic_service;Gazetteer_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides access to a directory of instances of a class or classes of real-world phenomena
containing some information regarding position.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geocoding_Service -->

<Class rdf:about="&ISO_geographic_service;Geocoding_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to augment location-based text references with geographic
coordinates.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Animation_Viewer -->

<Class rdf:about="&ISO_geographic_service;Geographic_Animation_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Viewer"/>
  <rdfs:comment xml:lang="en">Allows a human to sequence views of the same geographic location at different
times</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Annotation_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Annotation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Metadata_Processing_Service"/>
  <rdfs:comment xml:lang="en">Services to add ancillary information to an image or a feature in a Feature
Collection.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Communication_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Communication_Service">

```

```

    <equivalentClass>
      <Restriction>
        <onProperty rdf:resource="&OntologyComopoSIG;realizaTarea"/>
        <someValuesFrom rdf:resource="&OntologyComopoSIG;ComunicacionGeograficaTarea"/>
      </Restriction>
    </equivalentClass>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;ServicioGeografico"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Compression_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Compression_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Communication_Service"/>
  <rdfs:comment xml:lang="en">Service that converts spatial portions of a feature collection to and from compressed
form.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Data-structure_Viewer -->

<Class rdf:about="&ISO_geographic_service;Geographic_Data-structure_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Viewer"/>
  <rdfs:comment xml:lang="en">Allows a user to access part of dataset to see its internal structure.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Editor -->

<Class rdf:about="&ISO_geographic_service;Geographic_Editor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Human_Interaction_Service"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Feature_Editor -->

<Class rdf:about="&ISO_geographic_service;Geographic_Feature_Editor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Editor"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Format_Conversion_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Format_Conversion_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Communication_Service"/>
  <rdfs:comment xml:lang="en">Service that converts from one geographic data format to another.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Human_Interaction_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Human_Interaction_Service">
  <rdfs:label xml:lang="en">Geographic Human Interaction Service</rdfs:label>
  <equivalentClass>
    <Restriction>
      <onProperty rdf:resource="&OntologyComopoSIG;realizaTarea"/>
      <someValuesFrom rdf:resource="&OntologyComopoSIG;InteraccionHumanaTarea"/>
    </Restriction>
  </equivalentClass>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;ServicioGeografico"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Imagery_Viewer -->

```

```

<Class rdf:about="&ISO_geographic_service;Geographic_Imagery_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Viewer"/>
  <rdfs:comment xml:lang="en">Visualizes coverage data including the mapping of sample dimensions in the coverage to
colours in the display.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Information_Extraction_Service -
->

<Class rdf:about="&ISO_geographic_service;Geographic_Information_Extraction_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service supporting the extraction of feature and terrain information from remotely sensed and
scanned images.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Metadata_Processing_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Metadata_Processing_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Processing_Service"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Model-
Information_Management_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Model-Information_Management_Service">
  <rdfs:label xml:lang="en">Geographic Model/Information Management Service</rdfs:label>
  <equivalentClass>
    <Restriction>
      <onProperty rdf:resource="&OntologyComopoSIG;realizaTarea"/>
      <someValuesFrom rdf:resource="&OntologyComopoSIG;AdministracionInformacionTarea"/>
    </Restriction>
  </equivalentClass>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;ServicioGeografico"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Mosaicing_Viewer -->

<Class rdf:about="&ISO_geographic_service;Geographic_Mosaicing_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Viewer"/>
  <rdfs:comment xml:lang="en">Allows combination of views of geographic data for adjacent areas into a single
view.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Perspective_Viewer -->

<Class rdf:about="&ISO_geographic_service;Geographic_Perspective_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Viewer"/>
  <rdfs:comment xml:lang="en">Allows the view point to be changed</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Processing_Service -->

<Class rdf:about="&ISO_geographic_service;Geographic_Processing_Service">
  <equivalentClass>
    <Restriction>
      <onProperty rdf:resource="&OntologyComopoSIG;realizaTarea"/>
      <someValuesFrom rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
    </Restriction>
  </equivalentClass>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;ServicioGeografico"/>

```

```

</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Service_Editor -->
<Class rdf:about="&ISO_geographic_service;Geographic_Service_Editor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Editor"/>
  <rdfs:comment xml:lang="en">Allows a user to control geographic processing services</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Spatial_Processing_Service -->
<Class rdf:about="&ISO_geographic_service;Geographic_Spatial_Processing_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Processing_Service"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Spreadsheet_Viewer -->
<Class rdf:about="&ISO_geographic_service;Geographic_Spreadsheet_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Viewer"/>
  <rdfs:comment xml:lang="en">Allows user to interact with multiple data objects and to request calculations similar to an
arithmetic spreadsheet but extended to geographic data.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Symbol_Editor -->
<Class rdf:about="&ISO_geographic_service;Geographic_Symbol_Editor">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Editor"/>
  <rdfs:comment xml:lang="en">Allows a human to select and manage symbol libraries.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_System_Management_Service -->
<Class rdf:about="&ISO_geographic_service;Geographic_System_Management_Service">
  <equivalentClass>
    <Restriction>
      <onProperty rdf:resource="&OntologyComopoSIG;realizaTarea"/>
      <someValuesFrom rdf:resource="&OntologyComopoSIG;AdministracionSistemasTarea"/>
    </Restriction>
  </equivalentClass>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;ServicioGeografico"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Temporal_Processing_Service -->
<Class rdf:about="&ISO_geographic_service;Geographic_Temporal_Processing_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Temporal_Reference_System_Transformation_Service</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Thematic_Processing_Service -->
<Class rdf:about="&ISO_geographic_service;Geographic_Thematic_Processing_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Processing_Service"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Viewer -->

```

```
<Class rdf:about="&ISO_geographic_service;Geographic_Viewer">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Human_Interaction_Service"/>
  <rdfs:comment xml:lang="en">Allows a user to view on or more feature collections or coverages, and interact with map data,
  e.g., displaying, overlaying and querying.</rdfs:comment>
</Class>
```

```
<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geographic_Workflow-
Task_Management_Service -->
```

```
<Class rdf:about="&ISO_geographic_service;Geographic_Workflow-Task_Management_Service">
  <equivalentClass>
    <Restriction>
      <onProperty rdf:resource="&OntologyComopoSIG;realizaTarea"/>
      <someValuesFrom rdf:resource="&OntologyComopoSIG;GestionTareasTarea"/>
    </Restriction>
  </equivalentClass>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;ServicioGeografico"/>
</Class>
```

```
<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geoparameter_Calculation_Service -->
```

```
<Class rdf:about="&ISO_geographic_service;Geoparameter_Calculation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to derive application-oriented quantitative results that are not available from the raw
  data itself.</rdfs:comment>
</Class>
```

```
<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Geoparsing_Service -->
```

```
<Class rdf:about="&ISO_geographic_service;Geoparsing_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to scan text documents for location-based references.</rdfs:comment>
</Class>
```

```
<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Image_Coordinate_Conversion_Service -->
```

```
<Class rdf:about="&ISO_geographic_service;Image_Coordinate_Conversion_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">A coordinate transformation or coordinate conversion service to change the coordinate
  reference system for an image.</rdfs:comment>
</Class>
```

```
<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Image_Geometry_Model_Conversion_Service
-->
```

```
<Class rdf:about="&ISO_geographic_service;Image_Geometry_Model_Conversion_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Sensor_Geometry_Model_Adjustment_Service"/>
  <rdfs:comment xml:lang="en">Service that converts sensor geometry models into a different but equivalent sensor geometry
  model.</rdfs:comment>
</Class>
```

```
<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Image_Manipulation_Service -->
```

```
<Class rdf:about="&ISO_geographic_service;Image_Manipulation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Services for manipulating data values in images: changing colour and contrast values, applying
  various filters, manipulating image resolution, noise removal, &quot;striping&quot;, systematic-radiometric corrections,
  atmospheric attenuation, changes in scene illumination, ets.</rdfs:comment>
</Class>
```

```

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Image_Processing_Service -->

<Class rdf:about="&ISO_geographic_service;Image_Processing_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to change the values of thematic attributes of an image using a mathematical
function.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Image_Synthesis_Service -->

<Class rdf:about="&ISO_geographic_service;Image_Synthesis_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Services for creating or transforming images using computer-based spatial models, perspective
transformations, and manipulaitons of image characteristics to improve visibility, sharpen resolution, and/or reduce the effects of
cloud cover or haze.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Image_Understanding_Service -->

<Class rdf:about="&ISO_geographic_service;Image_Understanding_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Services that provide automated image change detection, registered image differencing,
significance-of-difference analysis and display, and area-based and model-based differencing.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Map_Access_Service -->

<Class rdf:about="&ISO_geographic_service;Map_Access_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides a client access to a geographic graphics, i.e., pictures of geographic
data.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Messaging_Service -->

<Class rdf:about="&ISO_geographic_service;Messaging_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Communication_Service"/>
  <rdfs:comment xml:lang="en">Service that allows multiple users to simultaneously view, comment about, and request edits of
feature collections.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Multi-band_Image_Manipulaiton_Service -->

<Class rdf:about="&ISO_geographic_service;Multi-band_Image_Manipulaiton_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Image_Manipulation_Service"/>
  <rdfs:comment xml:lang="en">Services that modify an image using the multiple bands of the image.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Object_Detection_Service -->

<Class rdf:about="&ISO_geographic_service;Object_Detection_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to detect real-world objects in an image.</rdfs:comment>
</Class>

```

```

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Order_Handling_Service -->
<Class rdf:about="&ISO_geographic_service;Order_Handling_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides a client with the ability to order products from a provider.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Orthorectification_Service -->
<Class rdf:about="&ISO_geographic_service;Orthorectification_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Rectified_Service"/>
  <rdfs:comment xml:lang="en">A rectification service that removes image tilt and displacement due to terrain
  evevation.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Positioning_Service -->
<Class rdf:about="&ISO_geographic_service;Positioning_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service provided by a position-providing device to use, obtain and unambiguously interpret
  position information, and determine whether the results meet the requirements of the use.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Product_Access_Service -->
<Class rdf:about="&ISO_geographic_service;Product_Access_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides access to and management of a geographic product store.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Proximity_Analysis_Service -->
<Class rdf:about="&ISO_geographic_service;Proximity_Analysis_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Given a position or geographic feature, finds all objects with a given set of attributes that are
  located within an user-specified distance of the position or feature.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Rectified_Service -->
<Class rdf:about="&ISO_geographic_service;Rectified_Service">
  <rdfs:label xml:lang="en">Service for transforming an image into a perpendicular parallel projection and therefore a constant
  scale.</rdfs:label>
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Reduced_Resolution_Generation_Service -->
<Class rdf:about="&ISO_geographic_service;Reduced_Resolution_Generation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that reduces the resolution of an image.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Registry_Service -->
<Class rdf:about="&ISO_geographic_service;Registry_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides access to store of metadata about types.</rdfs:comment>

```

```

</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Remote_File_Executable_Management -->
<Class rdf:about="&ISO_geographic_service;Remote_File_Executable_Management">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Communication_Service"/>
  <rdfs:comment xml:lang="en">Service that provides access to secondary storage of geographic features as if it were local to
the client.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Route_Determination_Service -->
<Class rdf:about="&ISO_geographic_service;Route_Determination_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to determine the optimal path between two specified points based on the input
parameters and properties contained in the Feature Collection.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Sampling_Service -->
<Class rdf:about="&ISO_geographic_service;Sampling_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that extracts data from an input using a consistent sampling scheme either by location
or by grid coordinates.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Sensor_Description_Service -->
<Class rdf:about="&ISO_geographic_service;Sensor_Description_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Provides the description of a coverage sensor, including sensor locaiton and orientation, as well
as the sensor's geometric, dynamic, and radiometric characteristics for geoprocessing purposes.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Sensor_Geometry_Model_Adjustment_Service
-->
<Class rdf:about="&ISO_geographic_service;Sensor_Geometry_Model_Adjustment_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that adjusts sensor geometry models to improve the match of the image with other
images and /or known ground positions.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#ServicioGeografico -->
<Class rdf:about="&ISO_geographic_service;ServicioGeografico">
  <rdfs:label xml:lang="en">Geographic Service</rdfs:label>
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Spatial_Counting_Service -->
<Class rdf:about="&ISO_geographic_service;Spatial_Counting_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that counts geographic features of a given type within a specified area.</rdfs:comment>
</Class>

```

```

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Spatial_Subsetting_Service -->

<Class rdf:about="&ISO_geographic_service;Spatial_Subsetting_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that extracts data from an input in a continuous spatial region either by geographic
location or by grid coordinates.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Standing_Order_Service -->

<Class rdf:about="&ISO_geographic_service;Standing_Order_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Model-Information_Management_Service"/>
  <rdfs:comment xml:lang="en">Order handling service that allows a user to request that a product over a geographic area be
disseminated when it becomes available. Such dissemination includes receive, prepare (i.e., reformat, compress, decompress, etc.),
prioritize, and transmit geographic information requested through standing queries or profiles.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Statistical_Calculation_Service -->

<Class rdf:about="&ISO_geographic_service;Statistical_Calculation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Metadata_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to calculate the statistics of data set.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Subscription_Service -->

<Class rdf:about="&ISO_geographic_service;Subscription_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Workflow-Task_Management_Service"/>
  <rdfs:comment xml:lang="en">Allows clients to register for notification about events.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Temporal_Proximity_Analysis_Service -->

<Class rdf:about="&ISO_geographic_service;Temporal_Proximity_Analysis_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Temporal_Processing_Service"/>
  <rdfs:comment xml:lang="en">Given a temporal interval or event, find all objects with a given set of attributes that are located
within an user-specified interval from the interval or event.</rdfs:comment>
</Class>

<!--
http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Temporal_Reference_System_Transformation_Service -->

<Class rdf:about="&ISO_geographic_service;Temporal_Reference_System_Transformation_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Temporal_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to change the values of temporal instances from one temporal reference system to
another temporal reference system.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Temporal_Sampling_Service -->

<Class rdf:about="&ISO_geographic_service;Temporal_Sampling_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Temporal_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that extracts data from an input using a consistent sampling scheme based on temporal
position values.</rdfs:comment>
</Class>

```

```

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Temporal_Subsetting_Service -->
<Class rdf:about="&ISO_geographic_service;Temporal_Subsetting_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Temporal_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that extracts data from an input in a continuous interval based on temporal position
values.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Thematic_Classification_Service -->
<Class rdf:about="&ISO_geographic_service;Thematic_Classification_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service to classify regions of geographic data based on thematic attributes.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Thematic_Subsetting_Service -->
<Class rdf:about="&ISO_geographic_service;Thematic_Subsetting_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Thematic_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that extracts data from an input based on parameter values.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Tiling_Change_Service -->
<Class rdf:about="&ISO_geographic_service;Tiling_Change_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Spatial_Processing_Service"/>
  <rdfs:comment xml:lang="en">Service that changes the tiling of geographic data.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Transfer_Service -->
<Class rdf:about="&ISO_geographic_service;Transfer_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Communication_Service"/>
  <rdfs:comment xml:lang="en">Service that provides implementation of one or more transfer protocols, which allows data
transfer between distributed information systems over off-line or on-line communication media.</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Workflow_Enactment_Management -->
<Class rdf:about="&ISO_geographic_service;Workflow_Enactment_Management">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Human_Interaction_Service"/>
  <rdfs:comment xml:lang="en">Provides user interaction with a workflow enactment service</rdfs:comment>
</Class>

<!-- http://geobrain.laits.gmu.edu/ontology/2004/12/ISO_geographic_service.owl#Workflow_Enactment_Service -->
<Class rdf:about="&ISO_geographic_service;Workflow_Enactment_Service">
  <rdfs:subClassOf rdf:resource="&ISO_geographic_service;Geographic_Workflow-Task_Management_Service"/>
  <rdfs:comment xml:lang="en">Interprets a chain and controls the instantiation of services and sequencing of
activities.</rdfs:comment>
</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#AdministracionInformacionTarea -->
<Class rdf:about="&OntologyComopoSIG;AdministracionInformacionTarea">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Tarea"/>
</Class>

```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#AdministracionSistemasTarea -->

```
<Class rdf:about="&OntologyComopoSIG;AdministracionSistemasTarea">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Tarea"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#ComercialLicencia -->

```
<Class rdf:about="&OntologyComopoSIG;ComercialLicencia">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Licencia"/>
  <disjointWith rdf:resource="&OntologyComopoSIG;NoComercialLicencia"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#ComponenteSIG -->

```
<Class rdf:about="&OntologyComopoSIG;ComponenteSIG">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#ComunicacionGeograficaTarea -->

```
<Class rdf:about="&OntologyComopoSIG;ComunicacionGeograficaTarea">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Tarea"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#Contacto -->

```
<Class rdf:about="&OntologyComopoSIG;Contacto">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#Datos -->

```
<Class rdf:about="&OntologyComopoSIG;Datos">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#GestionTareasTarea -->

```
<Class rdf:about="&OntologyComopoSIG;GestionTareasTarea">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Tarea"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#HardwareRequerimiento -->

```
<Class rdf:about="&OntologyComopoSIG;HardwareRequerimiento">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Requerimiento"/>
  <disjointWith rdf:resource="&OntologyComopoSIG;SoftwareRequerimiento"/>
</Class>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#InteraccionHumanaTarea -->

```
<Class rdf:about="&OntologyComopoSIG;InteraccionHumanaTarea">
  <rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Tarea"/>
```

</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#Licencia -->

<Class rdf:about="&OntologyComopoSIG;Licencia">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#NoComercialLicencia -->

<Class rdf:about="&OntologyComopoSIG;NoComercialLicencia">
<rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Licencia"/>
</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#ProcesamientoGeograficoTarea -->

<Class rdf:about="&OntologyComopoSIG;ProcesamientoGeograficoTarea">
<rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Tarea"/>
</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#Requerimiento -->

<Class rdf:about="&OntologyComopoSIG;Requerimiento">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#SoftwareRequerimiento -->

<Class rdf:about="&OntologyComopoSIG;SoftwareRequerimiento">
<rdfs:subClassOf rdf:resource="&OntologyComopoSIG;Requerimiento"/>
</Class>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#Tarea -->

<Class rdf:about="&OntologyComopoSIG;Tarea">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>

<!-- http://www.w3.org/2002/07/owl#Thing -->

<Class rdf:about="http://www.w3.org/2002/07/owl#Thing">
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</Class>

<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#GEOTRANSLicense -->

```
<NamedIndividual rdf:about="&OntologyComopoSIG;GEOTRANSLicense">
  <rdf:type rdf:resource="&OntologyComopoSIG;NoComercialLicencia"/>
</NamedIndividual>
```

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#GeographicTranslator -->
```

```
<NamedIndividual rdf:about="&OntologyComopoSIG;GeographicTranslator">
  <rdf:type rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
  <OntologyComopoSIG:tieneVersion rdf:datatype="&xsd:string">2.2.5</OntologyComopoSIG:tieneVersion>
  <OntologyComopoSIG:escritoEnLenguajeProgramacion
rdf:datatype="&xsd:string">C</OntologyComopoSIG:escritoEnLenguajeProgramacion>
  <OntologyComopoSIG:soportaSistemaOperativo
rdf:datatype="&xsd:string">GNU/Linux</OntologyComopoSIG:soportaSistemaOperativo>
  <OntologyComopoSIG:soportaSistemaOperativo
rdf:datatype="&xsd:string">Windows</OntologyComopoSIG:soportaSistemaOperativo>
  <OntologyComopoSIG:tieneSitioWeb rdf:datatype="&xsd:string">http://earth-
info.nga.mil/GandG/geotrans/</OntologyComopoSIG:tieneSitioWeb>
  <OntologyComopoSIG:soportaSistemaOperativo rdf:datatype="&xsd:string">other
Unices</OntologyComopoSIG:soportaSistemaOperativo>
  <OntologyComopoSIG:tieneLicencia rdf:resource="&OntologyComopoSIG;GEOTRANSLicense"/>
  <OntologyComopoSIG:realizaTarea rdf:resource="&OntologyComopoSIG;convert"/>
  <OntologyComopoSIG:tieneEntrada rdf:resource="&OntologyComopoSIG;geographicCoordinates"/>
  <OntologyComopoSIG:tieneSalida rdf:resource="&OntologyComopoSIG;mapProjections"/>
</NamedIndividual>
```

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#OpenLayers -->
```

```
<NamedIndividual rdf:about="&OntologyComopoSIG;OpenLayers">
  <rdf:type rdf:resource="&OntologyComopoSIG;ComponenteSIG"/>
  <OntologyComopoSIG:tieneNombre rdf:datatype="&xsd:string">OpenLayers</OntologyComopoSIG:tieneNombre>
  <OntologyComopoSIG:realizaTarea rdf:resource="&OntologyComopoSIG;display"/>
  <OntologyComopoSIG:tieneSalida rdf:resource="&OntologyComopoSIG;map"/>
</NamedIndividual>
```

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#annotate -->
```

```
<NamedIndividual rdf:about="&OntologyComopoSIG;annotate">
  <rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>
```

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#calculate -->
```

```
<NamedIndividual rdf:about="&OntologyComopoSIG;calculate">
  <rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>
```

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#convert -->
```

```
<NamedIndividual rdf:about="&OntologyComopoSIG;convert">
  <rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>
```

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#coordinateSystems -->
```

```
<NamedIndividual rdf:about="&OntologyComopoSIG;coordinateSystems">
  <rdf:type rdf:resource="&OntologyComopoSIG;Datos"/>
</NamedIndividual>
```

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#datums -->

<NamedIndividual rdf:about="&OntologyComopoSIG;datums">
<rdf:type rdf:resource="&OntologyComopoSIG;Datos"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#display -->

<NamedIndividual rdf:about="&OntologyComopoSIG;display">
<rdf:type rdf:resource="&OntologyComopoSIG;InteraccionHumanaTarea"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#geographicCoordinates -->

<NamedIndividual rdf:about="&OntologyComopoSIG;geographicCoordinates">
<rdf:type rdf:resource="&OntologyComopoSIG;Datos"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#manipulate -->

<NamedIndividual rdf:about="&OntologyComopoSIG;manipulate">
<rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#map -->

<NamedIndividual rdf:about="&OntologyComopoSIG;map">
<rdf:type rdf:resource="&OntologyComopoSIG;Datos"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#mapProjections -->

<NamedIndividual rdf:about="&OntologyComopoSIG;mapProjections">
<rdf:type rdf:resource="&OntologyComopoSIG;Datos"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#measure -->

<NamedIndividual rdf:about="&OntologyComopoSIG;measure">
<rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#positioning -->

<NamedIndividual rdf:about="&OntologyComopoSIG;positioning">
<rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>

<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#proximity_analysis -->

<NamedIndividual rdf:about="&OntologyComopoSIG;proximity_analysis">
<rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>

```
<!-- http://www.semanticweb.org/ontologies/2010/10/OntologyComopoSIG.owl#transformate -->
<NamedIndividual rdf:about="&OntologyComopoSIG;transformate">
  <rdf:type rdf:resource="&OntologyComopoSIG;ProcesamientoGeograficoTarea"/>
</NamedIndividual>
</rdf:RDF>
```

```
<!-- Generated by the OWL API (version 3.1.0.20069) http://owlapi.sourceforge.net -->
```