



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

UNIVERSIDAD DE LOS ANDES
FACULTAD DE INGENIERÍA
DOCTORADO EN CIENCIAS APLICADAS

ESQUEMAS DE APRENDIZAJE ONTOLOGICO PARA SISTEMAS AUTO-ORGANIZADOS Y EMERGENTES

Autor:

Ing. MSc. Maribel Mendonça

Tutores:

Dr. José Aguilar

Dra. Niriaska Perozo

*TESIS DOCTORAL PRESENTADA ANTE LA UNIVERSIDAD DE LOS ANDES COMO
REQUISITO FINAL PARA OPTAR AL TÍTULO DE DOCTOR*

Junio, 2019

*A Dios Todopoderoso
A mis Seres Queridos*

RESUMEN

En este trabajo doctoral se presenta un “Esquema de Aprendizaje Ontológico para Sistemas Auto-Organizados y Emergentes”, para la generación y actualización dinámica de las ontologías en este tipo sistemas. Se define el concepto de “Emergencia Ontológica”, como un proceso que busca adaptar una ontología a los cambios y nuevos componentes de un dominio dinámico, emergente y auto-organizado, a través de la aplicación de un conjunto de reglas, que permiten de manera global el surgimiento de una nueva conceptualización (conceptos emergentes). Su objetivo es estructurar la información y conocimiento que se pueda ir generando en el sistema, creando modelos conceptuales que puedan representar de forma adecuada el nuevo comportamiento que vaya surgiendo. Surge de la necesidad de representar ontológicamente una conceptualización de una realidad que es dinámica, y que no se puede pre-definir o pre-determinar, para así generar modelos de conocimiento emergentes, acordes a la dinámica del sistema, que permitan enfrentar su escalabilidad y evolución. El Esquema de Emergencia Ontológica (EEO) está conformado por cuatro componentes: registro semántico, análisis de conceptos, integración ontológica y evaluación de calidad de la ontología que emerge. A través de estos componentes se registran y analizan los modelos conceptuales que interactúan en el sistema, para integrarlos en una nueva ontología, todo en base a un conjunto de reglas de procesamiento definidas. Son determinantes las meta-ontologías, que guían el proceso de emergencia ontológica a través de categorías generales o meta-conceptos, para facilitar la integración de conceptos provenientes de las diferentes fuentes de datos. El EEO representa una propuesta novedosa en el ámbito del aprendizaje ontológico para sistemas dinámicos que demandan una actualización constante de sus ontologías en la misma medida que van cambiando, para así mantener modelos conceptuales consistentes con respecto al sistema que representan.

Palabras Claves: Aprendizaje Ontológico, Emergencia Ontológica, Auto-organización, Emergencia.

AGRADECIMIENTO

Para realizar esta tesis doctoral, resultado de muchas horas de trabajo, esfuerzo y dedicación, conté el apoyo de varias personas, que de alguna u otra manera contribuyeron a su culminación, y las cuales debo mi agradecimiento:

Primeramente doy gracias a Dios por darme esta oportunidad, y por llenarme de sabiduría paciencia y fortaleza para seguir siempre adelante.

A mis tutores Niriaska Perozo y José Aguilar, gracias infinitas por todas sus orientaciones, su apoyo y sus valiosos conocimientos, que fueron base fundamental en el desarrollo del trabajo y en mi crecimiento académico, profesional y personal en esta etapa de mi vida. Que Dios los llene de bendiciones!

A mi amado esposo Roberth, sin su apoyo, amor y comprensión no hubiese podido continuar, sobre todo en estos últimos meses tan difíciles que nos ha tocado vivir. Te amo!

A mis hijos Santiago, Samuel y Sara, mis mayores tesoros, mi motivación para seguir luchando cada día, pensando que mis logros son para ustedes y que les servirá de ejemplo para que puedan alcanzar sus propias metas siempre a base de trabajo, dedicación y perseverancia. Dios los bendiga! Los amo con todo mi corazón!

A mis padres ejemplo de trabajo, humildad y honradez, gracias por todo lo que me han dado y me siguen dando cada día.

A mis familiares y amigos que siempre han estado pendientes de mis avances y me han dado sus palabras de ánimo.

A mis compañeros en el doctorado, Marxjhony y Manuel, con los que desarrolle varios proyectos y publicaciones, donde compartimos ideas y experiencias, que sirvieron para nutrir este trabajo y que me brindaron su ayuda en los momentos en que la necesite, les deseo muchos éxitos!

A la prestigiosa Universidad de Los Andes por permitirme participar en este importante programa doctoral y a su personal, sobre todo a Luisa, Heidy, Taniara con las que conté en diferente situaciones donde necesité apoyo, muchas gracias y Dios les pague.

A mi alma mater, mi querida Universidad Centroccidental “Lisandro Alvarado”, por todo lo que me ha dado y por la oportunidad que me brindó de cursar este doctorado, que me ha permitido avanzar académica y profesionalmente.

A todos los que de alguna manera me han brindado su ayuda y su apoyo, que Dios los bendiga!

INDICE DE FIGURAS

Figura 2.1. Fases del Aprendizaje Ontológico	25
Figura 3.1. Estados de un Concepto	37
Figura 3.2. Ejemplo de Meta-Ontología de Dominio	41
Figura 3.3. Arquitectura de la BCS.....	42
Figura 3.4. Esquema de Emergencia Ontológica.....	46
Figura 3.5. Espacio de soluciones para el recorrido de las hormigas.....	52
Figura 3.6 Alineaciones entre la ontología O y las ontologías O_1, O_2, O_3	52
Figura 3.7. Diferentes combinaciones de alineaciones para cada concepto	53
Figura 4.1. Aplicación EEO	63
Figura 4.2. Iteración 1.....	68
Figura 4.3. Iteración 2.....	69
Figura 4.4. Iteración 3.....	72
Figura 4.5. OE del Escenario 1	73
Figura 4.6. OE del Escenario 1 al variar el URR a 30%.....	74
Figura 4.7. Parte de la Ontología Turismo.owl.....	75
Figura 4.8. OE del Escenario 2	76
Figura 4.9. OE del Escenario 3	78
Figura 4.10. OE del Escenario 3 al variar el URR a 60%.....	78
Figura 4.11. Middleware Reflexivo para Servicios de Ontologías Emergentes (MiR-EO).	82
Figura 4.12. Marcos Ontológicos en el Gestor Autnómico del MiR-EO en el Aml.....	83
Figura 4.13. Meta-Ontología de Componentes.....	86
Figura 4.14. Meta-Ontología de Contexto	87
Figura 4.15. OE del Aml.	90
Figura 4.16. Arquitectura de MiSCi.	92
Figura 4.17. Meta-Ontología para una ciudad inteligente.....	93
Figura 4.18. Ontología de Niebla.....	96
Figura A.1. Un CE a incluir en una MOD.....	112
Figura A.2. Posibles posiciones para ubicar un CE en la MOD	113

INDICE DE TABLAS

Tabla 1.1. Tabla comparativa de los estudios previos	12
Tabla 3.1. Ejemplo de Entropía	39
Tabla 3.2. Tabla de Anotaciones Semánticas	43
Tabla 3.3. Directorio Ontológico	43
Tabla 3.4. Tabla de Meta-Conceptos	43
Tabla 3.5. Matriz de Aprendizaje Colectivo de Propiedades (MACP).	54
Tabla 3.6. Tabla de Conceptos con Propiedades Comunes (TCPC).....	55
Tabla 4.1. Eventos Semánticos.....	65
Tabla 4.2. Conceptos Candidatos.....	66
Tabla 4.3. Ontologías del Área de Turismo	77
Tabla 4.4. Conceptos Candidatos del Aml.....	89
Tabla 4.5. Tabla de Meta-Conceptos en Ciudades Inteligentes.....	94
Tabla 4.6. Tabla de Meta-Conceptos en la Ontología de Niebla.....	96

INDICE DE MACRO-ALGORITMOS

Macro-Algoritmo 3.1: Esquema de Emergencia Ontológica	47
Macro-Algoritmo 3.2 : Generación de Meta-Ontología del Dominio	55
Macro-Algoritmo 3.3: Registro Semántico.....	56
Macro-Algoritmo 3.4: Búsqueda de CC.....	57
Macro-Algoritmo 3.5: Análisis de Conceptos.....	59
Macro-Algoritmo 3.6: Integración Ontológica	60
Macro-Algoritmo 3.7: Organización de la Mod.....	60
Macro-Algoritmo 3.8 : Evaluación de Calidad	61
Macro-Algoritmo A.1: Algoritmo General de ACO.....	114
Macro-Algoritmo A.2: Búsqueda de una Solución	115

INDICE GENERAL

RESUMEN	iii
AGRADECIMIENTO.....	iv
INDICE DE FIGURAS	v
INDICE DE TABLAS	vi
INDICE DE MACRO-ALGORITMOS	vii
CAPITULO 1: INTRODUCCIÓN	1
1.1. PLANTEAMIENTO DEL PROBLEMA	3
1.2. ANTECEDENTES.....	6
1.3. OBJETIVOS.....	13
1.4. JUSTIFICACIÓN E IMPORTANCIA	13
1.5. ORGANIZACIÓN DE LA TESIS.....	14
CAPITULO 2: MARCO TEORICO.....	15
2.1. APRENDIZAJE ONTOLÓGICO	15
2.1.1. <i>Ingeniería del Conocimiento</i>	16
2.1.2. <i>Ontologías</i>	17
2.1.3. <i>Minería Ontológica</i>	19
2.1.4. <i>Características del Aprendizaje Ontológico</i>	21
2.1.5. <i>Fases del Aprendizaje Ontológico</i>	24
2.2. AUTO-ORGANIZACIÓN Y EMERGENCIA.....	26
2.2.1. <i>Sistemas Multiagente</i>	26
2.2.2. <i>Computación Emergente</i>	30
2.2.3. <i>Sistemas Auto-Organizados y Emergentes</i>	32
CAPITULO 3: ESQUEMA DE EMERGENCIA ONTOLÓGICA (EEO).....	34
3.1. LA EMERGENCIA ONTOLÓGICA	34
3.2. DESCRIPCIÓN DE LA PROPUESTA.....	34
3.3. ASPECTOS GENERALES DEL EEO	35
3.4. PROPIEDADES DEL EEO	36
3.4.1. <i>La Relevancia</i>	36
3.4.2. <i>La Entropía</i>	37
3.5. DEFINICIONES GENERALES DEL EEO	40
3.6. ARQUITECTURA GENERAL DEL EEO	45
3.7. COMPONENTES DEL EEO.....	48
3.7.1. <i>Proceso de Caracterización Semántica</i>	48
3.7.2. <i>Proceso Estructuración Ontológica</i>	59
CAPITULO 4: CASOS DE ESTUDIO.....	63
4.1. APLICACIÓN DEL EEO EN EL CONTEXTO DE DATOS ABIERTOS Y ONTOLOGÍAS.	64
4.2. APLICACIÓN DEL EEO EN AMBIENTES INTELIGENTES	79
4.3. APLICACIÓN DEL EEO EN CIUDADES INTELIGENTES	91
4.4. ANÁLISIS DE RESULTADOS.....	97

CONCLUSIONES Y RECOMENDACIONES	98
REFERENCIAS	102
APÉNDICES.....	111
5.1. APÉNDICE A: MECANISMO DE ORGANIZACIÓN DE UNA META-ONTOLOGÍA BASADO EN ACO.....	111
5.2. APÉNDICE B: HERRAMIENTA PARA LA APLICACIÓN DEL EEO.	117
5.3. APÉNDICE C: ARTÍCULOS PUBLICADOS EN EL MARCO DE LA TESIS.....	118

Capítulo 1:

INTRODUCCIÓN

Las ontologías están diseñadas para representar formalmente el conocimiento relacionado a un dominio específico o genérico, en términos de conceptos, relaciones entre los conceptos, e instancias de esos conceptos [1] [2]. La comunidad de investigadores del área de la ingeniería ontológica se ha enfocado en el desarrollo de ontologías a través de técnicas y métodos para la adquisición, fusión y representación del conocimiento, en la validación de las mismas, así como en la definición de técnicas de razonamiento en base a ellas. Uno de los resultados de esta amplia investigación en ingeniería ontológica es, por ejemplo, su uso formal para la representación del conocimiento en muchos ámbitos (por ejemplo, en la web semántica). Actualmente, existen cientos de ontologías disponibles en la web. En muchos casos, debido a los entornos dinámicos computacionales, las ontologías se encuentran distribuidas y están en constante evolución, lo que implica la necesidad de nuevos enfoques en las investigaciones en ingeniería ontológica para mantenerlas actualizadas. Las ontologías deben mantenerse y evolucionar de acuerdo a los cambios en el dominio que ellas representan, manteniendo su consistencia [3].

Por otro lado, el aprendizaje ontológico es actualmente un área de gran interés, considerando el inmenso crecimiento y evolución de las ontologías, sobre todo por el gran impulso que han tenido derivado por la web semántica. La web semántica viene demandando el uso intensivo de ontologías para hacer sus procesos más eficientes y efectivos, en lo referente, sobre todo, al procesamiento, búsqueda y almacenamiento de conocimiento. Las ontologías se presentan como una buena solución para alcanzar la interoperabilidad semántica¹, ya que permiten conceptualizar el conocimiento, generando formas comunes de representarlo y compartirlo [4]. Así, se pueden usar como un formato de intercambio de conocimiento, además, dándole a los sistemas la capacidad de razonar y tomar decisiones en base a su conocimiento [5].

El éxito de la interacción semántica entre sistemas demanda la permanente integración y actualización de las ontologías. Pero la actualización e integración de ontologías no puede ser del todo manual, por lo complicado, limitado, costoso y lento que puede ser. El aprendizaje ontológico investiga técnicas, automáticas o semiautomáticas, para la construcción, actualización e integración de ontologías, lo que permite, no sólo descubrir conocimiento ontológico a gran escala y a un ritmo rápido, sino que también, permite reducir los sesgos e inconsistencias del proceso manual [6]. El aprendizaje ontológico forma parte del área de minería semántica, la

¹ La interoperabilidad semántica es la capacidad que tienen los sistemas de intercambiar datos, información y conocimiento, entender sus significados y poder usarlos [13].

cual tiene como fin extraer conocimiento semántico desde diversas fuentes, como pueden ser: documentos, páginas web, bases de datos y ontologías, entre otros [7].

Con respecto a los Sistemas Auto-Organizados y Emergentes (SAOyE) podemos mencionar que se caracterizan por el surgimiento de nuevas propiedades y comportamientos en dichos sistemas, y por poseer una organización espontánea producida dinámicamente [8] [9]. Un SAOyE puede tener en un momento dado, gran cantidad de información proveniente de diferentes fuentes y en diferentes formatos. Son muchos los sistemas que se comportan de manera auto-organizada y emergente, tales como: las redes sociales, los ambientes inteligentes y la robótica distribuida, entre otros [9]. En este tipo de sistemas, los componentes que interactúan son dinámicos e impredecibles, pueden entrar y salir del sistema, y puede estar cambiando constantemente.

Ahora bien, las ontologías pueden contribuir a la interoperabilidad entre los componentes de un SAOyE, y hacer posible la conceptualización de sus elementos y comportamientos. Sin embargo, ya que estos comportamientos no se pueden predecir, y sus elementos no están predeterminados, esta conceptualización no puede predefinirse y debe ser completamente dinámica. Además, la gran variabilidad y heterogeneidad de sus componentes puede generar un aumento en el volumen y una diversidad de modelos conceptuales que interactúan, por lo tanto, se necesitan nuevas estrategias a nivel de ingeniería ontológica que permitan enfrentar esa heterogeneidad y dinamismo de forma eficiente. En este sentido, las ontologías que participan en un SAOyE, que conforman su marco ontológico, las cuales normalmente son distribuidas y heterogéneas, deben evolucionar, adaptándose a los cambios, necesidades y nuevas propiedades o comportamientos del sistema, manteniendo siempre su consistencia. Los algoritmos tradicionales de minería ontológica no pueden hacer frente a esta necesidad [10].

En este trabajo se introduce un nuevo concepto, denominado “emergencia ontológica”, que se puede definir como un proceso que busca adaptar una ontología, de forma iterativa, a las características cambiantes de un dominio dinámico, emergente y auto-organizado, a través de la aplicación de un conjunto de reglas, y en base a unos criterios de relevancia y entropía, previamente definidos. El objetivo de la emergencia ontológica es estructurar la información y conocimiento que se pueda ir generando en un SAOyE, creando modelos conceptuales que puedan representar de forma adecuada el nuevo comportamiento que vaya surgiendo en el sistema, generando nuevas ontologías. Estas ontologías emergentes, se deben integrar al marco ontológico, para que puedan ser usadas en los diferentes componentes del propio sistema. La “emergencia ontológica” puede entonces, dar soporte a los procesos emergentes del sistema, a nivel de sus modelos conceptuales. Surge de la necesidad de representar ontológicamente una conceptualización de una realidad que es dinámica, y que no se puede pre-definir o pre-determinar, para así generar modelos de conocimiento acordes a la dinámica del sistema, que permitan enfrentar la escalabilidad y evolución del mismo.

En esta propuesta se analiza el problema de aprendizaje ontológico en los SAOyE, en busca de un nuevo esquema para la generación y actualización dinámica de sus ontologías, que esté acorde con el comportamiento de estos sistemas dinámicos e impredecibles. Se propone para ello, lo que denominamos un Esquema de Emergencia Ontológica (EEO). Este esquema tiene un conjunto de procesos de registro, monitoreo, análisis y adaptación de los diversos modelos conceptuales que interactúan en el sistema, basados en una serie de reglas de procesamiento para favorecer la emergencia ontológica. En el esquema propuesto juegan un papel determinante las meta-ontologías, para guiar el proceso de generación de ontologías (emergencia ontológica) a través de la definición de categorías generales, a fin de facilitar la integración de conceptos provenientes de diferentes ontologías o fuentes.

De manera general entre las contribuciones logradas con este trabajo se tienen:

- El desarrollo de un Esquema de Aprendizaje Ontológico (EEO) para Sistemas Auto-Organizados y Emergentes (SAOyE)
- La definición de un proceso para la generación de meta-ontologías de forma automática en base a la “Teoría de Categorías” y al algoritmo de optimización por colonia de hormigas ACO como estrategia de computación emergente.
- Implementación una herramienta para la aplicación del EEO en los caso de estudio.
- Propuesta de un Middleware Reflexivo para la Emergencia Ontológica (MiR-EO) para el caso de Ambientes Inteligentes.
- Propuesta de una “Capa de Emergencia Ontológica” en el Middleware Reflexivo para Ciudades Inteligentes (MiSCi) y la adaptación para trabajar con Fog Computing.

Finalmente es importante resaltar la publicación de varios artículos derivados de este trabajo, lo cuales se pueden detallar en al Apéndice C.

1.1. PLANTEAMIENTO DEL PROBLEMA

El proceso relacionado con el manejo e integración de datos, requiere de una adecuada representación y formalización, que de alguna manera haga posible la interoperabilidad semántica entre sistemas. A través de las ontologías es posible crear modelos conceptuales que especifiquen de forma explícita y formal un dominio, permitiendo su caracterización semántica. La semántica (*del griego *semantikos* “lo que tiene significado”*), de acuerdo a [11], se refiere a la interpretación de signos lingüísticos como símbolos, palabras, expresiones o representaciones formales, buscando la correspondencia entre ellas y las situaciones o cosas del mundo real o abstracto. El éxito en los procesos de formalización de datos, da la posibilidad de generar, a su vez, procesos exitosos de razonamiento e interoperabilidad. Para ello, es necesario un crecimiento rápido y efectivo de ontologías que conformen la estructura semántica de los sistemas.

La semántica emergente estudia los principios que rigen la descentralización semántica en los sistemas distribuidos y su integración, para que pueda existir un

proceso efectivo de interoperabilidad, en cuanto a la generación e interpretación adecuada de conceptos [12]. La interoperabilidad semántica, vista como un fenómeno emergente que se construye incrementalmente, debe evolucionar constantemente a partir de fuentes de información que se actualizan, que aparecen y desaparecen [13]. Esta interoperabilidad depende de la eficiencia y calidad de los procesos de generación e integración entre ontologías.

Como se dijo anteriormente, los SAOyE se caracterizan por el surgimiento de propiedades y comportamientos que emergen como producto de las interacciones de sus partes, manteniendo su estructura y organización, a través de un proceso dinámico y adaptable, acorde a los requerimientos del entorno. La emergencia en estos sistemas se da en los procesos de adaptación a ambientes dinámicos, con el surgimiento de nuevos comportamientos [8]. En ese caso, es necesario tener mecanismos para detectar ese comportamiento global emergente, más allá de las interacciones locales de sus componentes. Particularmente, es necesario concebir nuevos procesos de actualización de sus ontologías, que puedan reflejar ese nuevo comportamiento, para adaptarlas a cualquier cambio.

Es interesante, entonces, indagar cómo se puede llevar a cabo el proceso de aprendizaje ontológico en un SAOyE, donde existen un conjunto de elementos (partes), que eventualmente tienen sus propias ontologías locales. Es fundamental estudiar cómo estos elementos contribuyen en la generación de una ontología general que emerge a través de la integración adecuada de los conceptos locales, que pueden representar nuevos conceptos o colaborar con el enriquecimiento de los ya existentes, todo esto derivado del proceso de interacción de las partes del sistema.

Adicionalmente, debido a que este tipo de sistemas debe gestionar una alta densidad de elementos autónomos con interacciones, y posibilitar una coordinación dinámica de sus actividades, esta propuesta de esquemas de aprendizaje ontológico debe considerar estas propiedades de auto-organización y emergencia en su modelado, a fin que pueda adaptarse a la complejidad y dinamismo propio de estos sistemas. Si se piensa en la forma en que se desarrolla el proceso de aprendizaje ontológico en este tipo de sistemas, se debe pensar entonces en un proceso igualmente auto-organizado, donde los elementos de la ontología tengan la capacidad de adaptarse continuamente a los cambios, para lograr la estabilidad y consistencia de los conceptos y sus relaciones. Este comportamiento emergente y auto-organizado, también abarca el posible surgimiento de un nuevo modelo de conocimiento, que se va construyendo de forma dinámica y emergente, con la participación e interacción entre todos sus componentes, conservando y adaptando su estructura y organización por sí mismo, sin controles externos.

Estamos hablando, entonces, de la necesidad de un nuevo enfoque para la generación de ontologías, a partir de un gran volumen y diversidad de modelos conceptuales, que se encuentran distribuidos, y que están en constante cambio. El objetivo es organizar el conocimiento que se genera en un SAOyE, para generar nuevos modelos de conocimiento que puedan dar soporte a los procesos emergentes, y que, por lo tanto, estén acordes a las dinámica y evolución del sistema.

En particular, los procesos de aprendizaje ontológico buscan diseñar estrategias para la generación automática o semi-automática de ontologías, para así generar nuevos modelos de conocimiento a gran escala [6]. Hasta ahora, muchos procesos de aprendizaje ontológico contemplan el aprendizaje a partir de documentos o estructuras estáticas: texto, xml, bases de datos, entre otras [14] [15] [16]. Algunos proponen generar ontologías a partir de procesos de integración de otras ontologías [17]. Otros estudios se han orientado al aprendizaje ontológico en la web (como por ejemplo, el etiquetado colaborativo [18] [19] [20]), dando paso a nuevos formalismos de aprendizaje ontológico donde se trata de convertir la información generada en la web en conocimiento formal. Algunos estudios más recientes consideran la necesidad de gestionar las ontologías como un proceso evolutivo y dinámico, considerando la información del entorno [21] [22] [23] [24] [25] [26] [27]. Otros proponen nuevos enfoques para el manejo de la semántica y el conocimiento en los sistemas abiertos [28] [29] [30] [31]. Ahora bien, no se han definido procesos o mecanismos formales que contemplen la posibilidad de que las ontologías se puedan ir generando automáticamente de forma incremental, a partir de fuentes de datos dinámicas.

En esta propuesta se plantea un nuevo esquema de aprendizaje ontológico para los SAOyE, considerando que, por las características propias de estos sistemas, este proceso de aprendizaje también debe tener característica de auto-organización y emergencia. De esta manera, la propuesta estará orientada especialmente a los procesos de generación, integración y mantenimiento de ontologías en SAOyE. Así, este estudio tiene como fin, ofrecer un mecanismo que pueda coadyuvar a satisfacer los requerimientos de actualización ontológica en estos sistemas, contribuyendo de alguna manera, al éxito de la interoperabilidad semántica.

Se realizan entonces los siguientes planteamientos:

¿Es posible definir un nuevo enfoque para el aprendizaje ontológico que pueda representar el nuevo conocimiento que va surgiendo en los sistemas auto-organizados y emergentes?

¿Cómo este proceso de aprendizaje ontológico puede tener características de auto-organización y emergencia?

¿En este proceso es posible definir heurísticas que permitan la adaptación dinámica y constante de las ontologías a partir de múltiples fuentes de datos, para así de forma iterativa reflejar las características cambiantes del dominio?

¿Se pueden establecer criterios para determinar qué conceptos deben formar parte de la ontología resultante y como será su organización?

¿Es posible usar estrategias de computación emergente en este proceso de aprendizaje ontológico?

¿Qué criterios se pueden definir para evaluar la calidad de la ontología resultante?

1.2. ANTECEDENTES

A continuación, se hace referencia a algunos estudios sobre aprendizaje ontológico, que nos permiten conocer las tendencias actuales en ésta área, en diversos ámbitos tales como: el aprendizaje a partir de fuentes estáticas y dinámicas, las diferentes estrategias y procesos de aprendizaje y las necesidades de nuevos enfoques en el aprendizaje.

En primer lugar, se reseñan algunos trabajos sobre aprendizaje ontológico de datos estructurados, semi-estructurados y no estructurados. Estos trabajos se enfocan en proponer sistemas y herramientas para generar ontologías a partir de fuentes de datos estáticas como bases de datos y documentos, y presentan diversas técnicas y métodos usados en el aprendizaje ontológico.

En [32] los autores presentan una propuesta de aprendizaje ontológico donde se integran varias técnicas de aprendizaje de información ontológica, para adquirir conceptos, relaciones, axiomas, combinar ontologías, etc. En ese trabajo se propone una arquitectura de aprendizaje llamada “Sistema para el Aprendizaje Automático de Ontologías e Información Léxica” (SALOX), donde a través de la combinación de diversas fuentes de aprendizaje (como bases de datos, documentos html, xml ó texto) con técnicas de aprendizaje automático, se actualiza el conocimiento usado en el “Framework Dinámico Semántico Ontológico para la Web Semántica” (DSOFSW), que interpreta consultas en lenguaje natural (Español) en la web. Ese trabajo continuó en [14], con la definición de un esquema de adquisición automática de conocimiento léxico y semántico. Ese esquema permite la adquisición de conocimiento sobre términos (palabras), conceptos (taxonómicos, no taxonómicos), relaciones entre ellos, y axiomas. Además, ese conocimiento adquirido es después almacenado en ontologías, usando técnicas de mezcla y enriquecimiento ontológico-

En el trabajo presentado en [15] se muestran técnicas de aprendizaje ontológico para obtener automáticamente términos, conceptos y relaciones a partir de documentos de texto. Las técnicas de aprendizaje ontológico explotan un amplio conocimiento que va desde datos no estructurados en cuerpos de texto, hasta datos estructurados en vocabularios semánticos. En esa investigación se usa la web como la fuente de conocimiento dinámico a través de todas las fases del aprendizaje ontológico, para la construcción de nubes de términos y ontologías ligeras de documentos. Las ontologías ligeras son un tipo de ontologías que hacen poco o ningún uso de axiomas (proposiciones sobre el dominio), como los glosarios, diccionarios web, tesauros, documentos XML² (“Extensible Markup Language”, por sus siglas en inglés), DTD³s (“Document Type Definitions”, por sus siglas en inglés), entre otros. Ellos presentan un enfoque original de aprendizaje ontológico aplicable a una amplia gama de dominios, que comprende las siguientes fases: pre-procesamiento de texto,

² XML es un meta-lenguaje que permite definir lenguajes de marcas para almacenar datos en forma legible [110].

³ DTD es una descripción de estructura y sintaxis de un documento XML [109].

procesamiento de texto, reconocimiento de términos, agrupación de términos y reconocimiento de relaciones. El sistema toma como entrada un conjunto de términos y texto en lenguaje natural, y produce como salida nubes de términos y ontologías ligeras.

En [16] se investigan técnicas semiautomáticas, para la construcción de ontologías. La técnica propuesta consiste en aplicar ciertas heurísticas a un conjunto de Documentos XSD⁴ ("*XML Schema Definition*", por sus siglas en inglés), que son documentos para describir el esquema y la estructura que pueden contener los documentos XML. La idea de la propuesta se basa en dos aspectos fundamentales: primero, el hecho que XML es actualmente el lenguaje más utilizado para intercambiar información en la Web, y segundo, que en el proceso de generar los documentos XSD se utilizan términos del dominio para nombrar las etiquetas de sus documentos, y por lo general, las organizaciones que generan dichos documentos se basan en estándares y normativas globales o propias de la organización, por lo que se espera encontrar en estos documentos información importante representativa del dominio. Las heurísticas propuestas para la identificación de elementos de la ontología se dividen en: heurísticas para extraer clases, extraer propiedades, determinar las relaciones específicas del dominio, determinar las relaciones de especialización/generalización que se aplican a documentos XSD del dominio. Con ello se generan los elementos que conformarán la ontología en construcción. Los elementos que se identificaron con las heurísticas propuestas, deben ser validados por expertos del dominio, para luego formalizar la ontología en un lenguaje apropiado.

Entre otros sistemas o herramientas de aprendizaje ontológico se pueden mencionar: WEBKB [33] que es un sistema de aprendizaje ontológico, cuyo objetivo es el aprendizaje de instancias y reglas desde documentos de la Web, combinando métodos estadísticos y lógicos; TEXT-TO-ONTO [34], que es un entorno para el aprendizaje semi-automático de ontologías, a partir de textos de dominios específicos, aplicando técnicas de aprendizaje automático; ONTOLEARN [35] que es un sistema de aprendizaje de ontologías, donde se busca ampliar una ontología genérica ya existente en un dominio, con otros conceptos y relaciones; ONTOLEARN RELOADED [36], que es un algoritmo basado en grafos para el aprendizaje de taxonomías desde cero; DL-LEARNER [37], que es un sistema de aprendizaje supervisado de sentencias en OWL⁵ ("*Web Ontology Language*", por sus siglas en Inglés), y Lógica Descriptiva, a partir de instancias; TEXT-2-ONTO [38], que es un sistema de aprendizaje ontológico (sucesor de Text-to-Onto) para la adquisición de ontologías desde documentos de texto, el cual hace uso de estrategias como: representar el conocimiento adquirido en un modelo de ontología probabilístico independiente del lenguaje objetivo, incorporar estrategias para el descubrimiento de cambios en los datos evitando el procesamiento de todo el texto desde cero cada vez que hay un cambio, y permitir al usuario trazar la evolución de la ontología.

⁴ XSD (XML Schema) es un lenguaje que sirve para definir la estructura de un documento XML [111].

⁵ OWL es un lenguaje de marcado para publicar y compartir datos usando ontologías [51].

En segundo lugar, se muestra cómo la integración de ontologías es usada como estrategia de aprendizaje ontológico. En el trabajo presentado en [17], se realiza una propuesta de una ontología emergente partiendo de ontologías disponibles en ambientes distribuidos. Se considera que en cada punto del ambiente distribuido se tiene una o más ontologías que representan la información que allí manejan. A medida que estos puntos cruzan información, va surgiendo el mapeo de estas ontologías. Ellos manifiestan que la construcción de una ontología a nivel organizacional es un proceso costoso y complicado, ya que se necesita un equipo de consultores especializados. Aunado a esto, el carácter dinámico de las organizaciones puede traer desactualizaciones rápidas de las ontologías, por lo que se hace necesario desarrollar mecanismos que permitan una generación y actualización de ontologías de forma ágil y descentralizada, que puedan reflejar el dinamismo de la organización. Se indica en ese estudio, que existen mecanismos para la interoperabilidad de ontologías que pueden ser provistos por el proceso de integración, los cuales ellos denominan: *Combinación (fusión ó mezcla)*, donde se tiene como resultado una versión de las ontologías originales combinadas en una ontología única con todos sus elementos juntos; y *Mapeo o enlace*, que mantiene las dos ontologías originales, agregándoles enlaces que indican cuales elementos son equivalentes, de manera de permitir la interacción entre ellas. Considerando que cada punto de la organización representa su información o conocimiento a través de una ontología local, y que es necesario la integración, se proponen allí mecanismos para extraer información valiosa sobre conceptos compartidos a través de los mecanismos de interoperabilidad ya descritos, que determinan los procesos de integración, los atributos de los conceptos a integrar y sus relaciones, a fin de facilitar el surgimiento de conceptos y propiedades que conformarán una ontología organizacional, que denominan ontología emergente.

En tercer lugar, se mencionan estudios recientes orientados al aprendizaje ontológico de fuentes dinámicas, como la web. En el etiquetado colaborativo, las etiquetas son generadas libremente por los usuarios, en diversos lugares y momentos, de forma dinámica y constante, y en estos trabajos se muestran las estrategias empleadas para entender la semántica de estas folksonomías⁶, capturando de alguna manera su estructura en ontologías. Entre algunos estudios de este tipo se pueden señalar los siguientes:

En [18] los autores proponen un modelo de aprendizaje ontológico que captura la estructura jerárquica semántica de una folksonomía. Esta práctica también se conoce como etiquetado colaborativo, donde a través de un conjunto de etiquetas definidas por los usuarios, se describen los contenidos publicados en la web. Así, en ese estudio se propone un enfoque de aprendizaje para crear una ontología que capture la estructura jerárquica semántica de las folksonomías. El enfoque propuesto consiste en tres etapas: primero, a través de un método probabilístico se genera un modelo de correspondencia entre las etiquetas y los recursos que ellas identifican,

⁶Una "Folksonomía" es un sistema de clasificación de información derivada de la práctica de gestión colaborativa de etiquetas, para anotar y categorizar contenidos en la Web [18].

luego, se estiman las posibles relaciones entre etiquetas caracterizando cuantitativamente las relaciones con medidas de divergencia, y finalmente, se determina el tipo de relación definitiva para construir la estructura jerárquica. Según este modelo de aprendizaje planteado, no se requiere tener un conocimiento previo de un dominio específico, y se puede aprender una jerarquía ontológica de cualquier aplicación de etiquetado social.

En [19] se presenta una metodología para convertir la información generada por los usuarios durante los procesos de etiquetado, en conocimiento formal. El proceso se inicia a partir del proceso de etiquetado colaborativo, donde los usuarios buscan identificar recursos en la web, para clasificarlos de acuerdo a lo que son o la información que contienen, favoreciendo de alguna manera el proceso de organización global de la información. Luego se procede a extraer la información vinculada al etiquetado de los recursos, para finalmente, una vez extraídos los datos, construir el contexto formal de conceptos, considerando los recursos como objetos y las etiquetas como atributos. En ese trabajo se propone que el proceso de evolución de una información etiquetada hacia su transformación en conocimiento, se aborde haciendo uso del Análisis Formal de Conceptos (AFC) [39]. AFC es una teoría matemática que formaliza la noción de "concepto", y permite calcular jerarquías de conceptos a partir de una tabla de relaciones. Es muy adecuada, entre otras aplicaciones, para la minería ontológica en folksonomías. En [20] los autores, considerando que las folksonomías son cada vez más adoptadas en los sistemas web, ya que pueden desempeñar un rol complementario en el proceso evolutivo de las ontologías, proponen un enfoque para el manejo de la evolución ontológica apoyado en folksonomías. De esta manera, consideran que el conocimiento sistemático formalizado en las ontologías por parte de un grupo restringido, puede enriquecerse con el conocimiento implícito producido colaborativamente por un grupo mucho más amplio.

En cuarto lugar, se presentan algunos trabajos donde se considera la gestión de ontologías como un proceso evolutivo y dinámico, y donde se muestran mecanismos para adaptarlas a los cambios, considerando la información del entorno.

En [21] se propone un enfoque que busca brindarle autonomía a las ontologías, para que crezcan dinámicamente según las necesidades y propósitos de su creación. Allí se brindan alternativas para la generación y actualización de estas ontologías de manera dinámica y autónoma, aprendiendo del ambiente y de estímulos de entrada, mediante el uso de una red neuronal no supervisada, conocida como mapas auto-organizados. En esta propuesta, la red neuronal es capaz de encontrar patrones sobre datos de entrada. La red operará sobre las distintas entidades (objetos reales) con las que tiene contacto, y el conjunto de señales de entrada serán filtradas de las interfaces con las que cuente el sistema físico que aloje la ontología autónoma. Estas señales de entrada serán procesadas y clasificadas por la red, arrojando como resultado diferentes patrones, que se convertirán en nuevos conceptos. Cada una de las características en común que encontró la red neuronal sobre los conceptos, son posibles relaciones.

En [22] los autores implementan un *framework* para la gestión de la evolución de ontologías, el cual trata de adaptar una ontología a los cambios y necesidades del sistema y sus usuarios de forma semi-automática, es decir, reduciendo lo más posible la intervención del usuario en el proceso, a través de la exploración y análisis de diversas fuentes de conocimiento externas. Este *framework* está formado por una serie de componentes que realizarán las siguientes tareas: i) Descubrir nueva información, para detectar nuevos posibles conceptos a agregar a la ontología base, contrastando su información con la contenida en dominios y aplicaciones externas, como corpus⁷ de textos, bases de datos u otras ontologías; ii) Validar datos para chequear la calidad y confianza de los conceptos; iii) Verificar los cambios ontológicos para tratar de descubrir relaciones entre conceptos, usando diversas técnicas de correspondencia, para determinar si es un nuevo concepto, cómo se puede relacionar, o si debe ser descartado; iv) Validar la evolución de la ontología, verificando la consistencia de los datos, conflictos o duplicaciones; v) Gestionar la evolución, registrándolos cambios en la ontología y propagando los cambios en otras ontologías o aplicaciones relacionadas.

En [23] se muestra un enfoque para evaluar de forma automática la evolución de una ontología a través de información del contexto. El objetivo es evaluar la pertinencia de las nuevas declaraciones o conceptos que se agregan a una ontología, es decir, cuánto se ajusta o acopla el concepto a la ontología. Para ello se usa información del contexto del nuevo concepto, por medio de la identificación de ontologías en la web donde esté presente ese concepto, a las que se llaman ontologías de contexto. En principio, se buscaba determinar el número de los conceptos compartidos entre las ontologías como un indicador de pertinencia, a más solapamientos de conceptos más pertinente. Este enfoque se consideró muy sencillo, por lo que luego se presenta un enfoque mejorado a través de la identificación no sólo de entidades compartidas, sino de la estructura que la rodea, es decir, la coincidencia de conceptos relacionados. Entonces, se establecen un conjunto de patrones de coincidencia como un indicador de pertinencia del concepto. Al ocurrir alguno de los patrones en la zona de intersección de la ontología en estudio y la ontología contextual, se determina un indicador de pertinencia del nuevo concepto en la ontología.

En el trabajo realizado en [24], considerando que en un entorno dinámico es necesario realizar cambios en una ontología, de acuerdo con los nuevos conocimientos y las necesidades de los usuarios, plantean que la evolución de la ontología es una tarea compleja y lenta, y presentan una herramienta de retroalimentación para los ingenieros ontológicos, para la gestión de la evolución ontológica. En esta herramienta se define un comportamiento adaptativo basado en los ajustes realizados por el ontólogo, tal que de forma interactiva e iterativa, se auto-organiza la ontología hasta alcanzar un estado satisfactorio. En [25] se resaltan, debido a que las ontologías no son entidades estáticas, la necesidad de herramientas para apoyar el proceso de evolución de las ontologías. En ese trabajo se proporcionan un conjunto de requisitos para los

⁷ Conjunto de datos, textos u otros materiales sobre determinada materia o dominio [45].

sistemas de evolución ontológica. En [26] se propone un *framework* para la evolución de ontologías, donde seleccionan de un conjunto de conceptos candidatos, un grupo de conceptos capaces de producir una nueva versión de una ontología dada, considerando que las ontologías deben evolucionar regularmente, en respuesta a los cambios en el dominio y a la aparición de nuevos requisitos. En [27], considerando la necesidad de entender y explorar la evolución de las ontologías, se presenta un *framework* para hacer seguimiento a la evolución de las ontologías. Resaltan que las ontologías evolucionan constantemente, a medida que surgen nuevos requisitos, y las opciones de modelado del pasado deben actualizarse o adaptarse. La exploración de esta evolución mejorará la comprensión, y aumentará el potencial de explotación, de las ontologías disponibles.

Finalmente, podemos mencionar algunos trabajos que muestran el uso de ontologías en sistemas emergentes, y la necesidad de definir nuevos esquemas para el manejo de la semántica y el conocimiento en los sistemas abiertos⁸:

En [28] se presenta una propuesta donde se hace uso de ontologías para analizar y detectar el comportamiento emergente de sistemas distribuidos. Allí proponen un sistema para la detección de comportamientos emergentes, para lo que hacen uso de una propiedad que llaman causalidad semántica, que definen como una secuencia de mensajes (eventos) que se tienen que dar para realizar una operación. Esta causalidad semántica se considera fundamental para la detección del comportamiento emergente. La propuesta consta de dos pasos: primero se construye un modelo del comportamiento de cada componente del sistema, a través de máquinas de estado finito que muestran el diagrama de secuencia de los procesos de cada componente. Se debe determinar los valores para cada estado, para lo cual usan la causalidad semántica. Luego, utilizando cada modelo de comportamiento de los componentes, se analizan los requerimientos para detectar posibles comportamientos emergentes usando ontologías. La ontología se construye para reflejar las relaciones y el flujo de información entre los componentes del sistema, de manera que puedan reflejar la causalidad semántica de forma general para todos los posibles escenarios.

En [29] los autores consideran que en los sistemas abiertos, donde hay un alto grado de flexibilidad y de adaptación a los cambios de su entorno, y donde no se puede asegurar consistencia de las fuentes de conocimiento, es necesario darle un enfoque social al manejo del conocimiento a través de ontologías abiertas. Este enfoque de ontologías abiertas considera que la ontología debe surgir y evolucionar en el proceso de comunicación entre múltiples fuentes de información y usuarios, de donde surgen afirmaciones y negaciones en el proceso de especificaciones de conceptos y relaciones, dándole un carácter dinámico. En [30] los autores resaltan que las ontologías tradicionales pueden perder la validez de sus conocimientos a medida que estos sistemas evolucionan, con sus cambios constantes, y con la entrada y salida de componente. La ontología debe convertirse en un producto emergente de este tipo de sistemas auto-organizados. Allí proponen un modelo abstracto genérico de red semántica social, que llaman modelo instancia-actor-concepto de ontología, para

⁸ Los sistemas abiertos son aquellos que interactúan con el entorno y con otros sistemas [29].

mostrar cómo emergen con este modelo, ontologías ligeras de conceptos y redes sociales de personas.

En [31] se presenta un *framework* para capturar dinámicamente en ontologías, nuevo conocimiento a partir de patrones de actividad en ambientes inteligentes, para reflejar la evolución en los cambios de comportamiento en un Modelo de Reconocimiento de Actividad (MRA). Este modelo, basado en ontologías, juega un papel vital en el ámbito de “Internet de las Cosas”; sin embargo, sufre de deficiencias por su naturaleza estática, que deriva en la incapacidad de evolucionar, y en la falta de adaptabilidad. Además, no es lo suficientemente completo para cubrir todas las actividades en un ambiente inteligente. Así, un MRA puede no reconocer o inferir correctamente nuevas actividades. De esta manera, proponen adaptar las ontologías aprendiendo de las actividades en el ambiente, en base a patrones de actividades realizados por los usuarios. Por otro lado, también buscan identificar nuevos patrones de actividades previamente desconocidos por el modelo, y adaptar nuevas propiedades en los modelos de actividad existentes. Todo lo anterior, permite enriquecer el modelo ontológico de un MRA.

Los trabajos presentados muestran diversos aspectos del aprendizaje ontológico, y la tendencia actual a generar nuevos mecanismos y estrategias de aprendizaje para ambientes dinámicos y sistemas abiertos, reflejando la necesidad de crear ontologías que puedan adaptarse a medida que evolucionan los sistemas. En la Tabla [xx] se muestra un cuadro comparativo entre algunos de los trabajos más importantes y esta propuesta, considerando varios criterios como son: las fuentes de datos, la activación del proceso de aprendizaje, la participación del usuario y las técnicas usadas en el proceso. Se puede observar en los trabajos previos no existe un enfoque que pueda abarcar de forma integral todos los requerimientos para el aprendizaje ontológico en los sistemas auto-organizados y emergentes. No existen mecanismos formales que contemplen la posibilidad de que las ontologías se puedan ir generando automáticamente de forma incremental, a partir de múltiples fuentes de datos dinámicas. Tampoco se han definido procesos que permita la auto-organización de la ontología de acuerdo a los requerimientos conceptuales del sistema con criterios para la inclusión y organización de conceptos. Es por ello que se plantea un nuevo enfoque, donde el proceso de aprendizaje ontológico pueda de forma dinámica y a través de criterios previamente definidos (como la relevancia de los conceptos y la entropía) actualizar una ontología de forma iterativa a partir de múltiples fuentes de datos y organizar su estructura de acuerdo a los requerimientos del sistema para adaptarse a las características cambiantes de un dominio emergente y auto-organizado.

TABLA I.1. TABLA COMPARATIVA DE LOS ESTUDIOS PREVIOS

	Fuentes de datos	Activación del proceso	Participación Usuario (Experto Ontológico)	Proceso de Aprendizaje

Estudios	Dinámicos	Estáticos	Única	Múltiple	Automático	Por el usuario	Baja	Media	Alta	Pre- Procesamiento	Computación Emergente.	Integración Ontológica	Auto-adaptación Ontológica	Iterativo	Relevancia de Conceptos
[14]		X	X			X		X		X					
[15]		X	X			X			X	X					
[16]		X		X		X			X	X					
[17]		X		X	X			X				X			
[18]	X		X			X		X							
[19]	X		X			X			X	X					
[20]	X		X			X			X						
[21]	X			X	X		X				X				
[22]	X			X		X		X		X					
[23]		X		X		X			X	X					
[24]	X		X			X			X						
[26]		X		X		X			X	X					
[27]		X	X			X		X		X					
EEO	X			X	X		X			X	X	X	X	X	X

1.3. OBJETIVOS

Objetivo General

Desarrollar un esquema de aprendizaje ontológico para sistemas auto-organizados y emergentes.

Objetivos Específicos

- 1) Estudiar las metodologías, métodos, técnicas y herramientas para el aprendizaje ontológico disponibles actualmente.
- 2) Estudiar los aspectos teóricos relacionados a los sistemas auto-organizados y emergentes.
- 3) Definir el esquema de aprendizaje ontológico, considerando las características de los sistemas auto-organizados y emergentes.
- 4) Verificar los esquemas que se propongan, a través de su aplicación en casos de estudios.

1.4. JUSTIFICACIÓN E IMPORTANCIA

Considerando que actualmente muchos sistemas requieren trabajar de manera auto-organizada y emergente, tales como: las redes sociales, los sistemas de gestión de redes, los ambientes inteligentes y la robótica distribuida, entre otros,

se presenta esta propuesta de esquemas de aprendizaje ontológico para sistemas emergentes y auto-organizados, con la idea que el proceso de aprendizaje pueda abordarse tomando en cuenta las características propias de estos sistemas. Para ello, se hace necesaria una actualización dinámica de sus ontologías en la misma medida que los sistemas van cambiando, para así mantener una conceptualización consistente de sus componentes. Para el desarrollo de la propuesta se consideraron los aspectos más importantes del aprendizaje ontológico, sus enfoques, metodologías y sistemas de aprendizaje, que permiten realizar procesos automáticos o semiautomáticos de aprendizaje, y su aplicación considerando los elementos característicos de la auto-organización y la emergencia.

En resumen, este estudio tiene como fin proporcionar esquemas de aprendizaje ontológico que puedan coadyuvar a satisfacer los requerimientos de generación, mantenimiento e integración de ontologías en los SAOyE. Así, se propone la generación de esquemas de aprendizaje ontológico, que deberían funcionar de forma dinámica, emergente y auto-organizada, en las tareas requeridas para la gestión de las ontologías. Todo esto se desarrolla en base al análisis y estudio de las metodologías, métodos, técnicas y herramientas que se pueden emplear para la interpretación, descubrimiento, definición y actualización de conceptos y relaciones que conformen las ontologías que representen el dominio de estos sistemas.

1.5. ORGANIZACIÓN DE LA TESIS

En este capítulo se describió el planteamiento del problema, los antecedentes donde se muestran diferentes estrategias de aprendizaje ontológico y el uso de ontologías en sistemas abiertos y emergentes, los objetivos de la investigación y su importancia. En el capítulo 2 se establecen algunos aspectos teóricos relacionados a las ontologías, al aprendizaje ontológico y a los SAOyE. En el capítulo 3 se propone el esquema de aprendizaje ontológico para sistemas auto-organizados y emergentes llamado “Esquema de Emergencia Ontológica”, sus propiedades, arquitectura y componentes. En el capítulo 4 se desarrollan algunos casos de estudio, a fin de mostrar la aplicación del esquema propuesto en diferentes escenarios, así como la integración con otros sistemas. Por último, se presentan las conclusiones del trabajo, los trabajos futuros y una discusión de los resultados obtenidos.

Capítulo 2:

MARCO TEORICO

En este capítulo se presenta una visión general sobre los aspectos teóricos más relevantes para este trabajo, agrupados en dos áreas de interés: el Aprendizaje Ontológico y la Auto-Organización y Emergencia.

2.1. APRENDIZAJE ONTOLÓGICO

El éxito de la interoperabilidad semántica entre sistemas, depende en gran medida de la proliferación de ontologías que permitan el enriquecimiento semántico de Internet. Para ello, el proceso de gestión ontológica se debe apoyar en procesos de aprendizaje automático. Aunque las herramientas de ingeniería ontológica han madurado últimamente, la adquisición manual de ontologías sigue siendo una tarea engorrosa, de alto costo y consumo de tiempo, convirtiéndola en el cuello de botella de la adquisición de conocimiento [40], por eso, la adquisición automática y semiautomática de ontologías toma fuerza e importancia.

El Aprendizaje Ontológico *“investiga técnicas semiautomáticas para la construcción de ontologías. De esta manera, el aprendizaje ontológico se refiere al descubrimiento y creación de conocimiento ontológico usando técnicas de aprendizaje automático, tales como: aprendizaje de máquina, procesamiento del lenguaje natural, programación inductiva y análisis formal de conceptos, entre otros”* [6].

Actualmente es ampliamente aceptada la concepción que *“el aprendizaje ontológico se refiere al aprendizaje de componentes de una ontología: conceptos (clases), taxonomías, relaciones conceptuales, atributos, axiomas...”* [41].

En [42] se define el aprendizaje ontológico como *“un conjunto de métodos y técnicas usadas para construir una ontología desde cero, enriquecer o adaptar ontologías existentes, de una forma semi-automática, usando fuentes de información y conocimiento heterogénea y distribuida, permitiendo la reducción de tiempo y esfuerzo en el proceso de desarrollo de ontologías”*.

De las definiciones, extraemos dos aspectos del proceso de gestión ontológica de interés en este trabajo, creación y actualización (mantenimiento) de ontologías.

El objetivo del aprendizaje ontológico es extraer conceptos y relaciones relevantes de una fuente de datos, que puede ser estructurada, semi-estructurada, o no estructurada. Combina diferentes técnicas de aprendizaje y recuperación de la información, para descubrir semántica en los datos y hacerla explícita. El resultado debe integrarse a un formato comprensible por la máquina [43].

Antes de mostrar en detalle las características y fases del aprendizaje ontológico, se presentan algunos aspectos de interés sobre la ingeniería del conocimiento, las ontologías y la minería ontológica.

2.1.1. Ingeniería del Conocimiento

La ingeniería del conocimiento es una rama de la Inteligencia Artificial, que se empezó a desarrollar en la década de los 70, para el diseño y desarrollo de Sistemas Basados en Conocimiento y Sistemas Expertos. Estos sistemas intentan representar el conocimiento y el razonamiento humano sobre un determinado dominio, computacionalmente [44]. En [45] se define la ingeniería del conocimiento como *“una disciplina que estudia el proceso de construcción, mantenimiento, así como las herramientas y lenguajes, para el desarrollo de sistemas basados en conocimiento”*.

El trabajo de los ingenieros del conocimiento consiste en extraer el conocimiento de los expertos humanos en una determinada área, y en codificar dicho conocimiento de manera que pueda ser procesado por un computador. Los procesos principales en la ingeniería del conocimiento son [44] [46]:

- *Adquisición del Conocimiento*: se refiere a la transferencia de conocimiento y experiencia en la resolución de problemas desde una fuente de conocimiento. La adquisición del conocimiento se puede realizar a partir de fuentes documentales (conocimiento estático o explícito), o a partir del diálogo, la observación y el análisis, para comprender la forma en que el experto llega a manejar su conocimiento, y precisar los procesos mentales que el experto realiza con su conocimiento. Dentro de los métodos de adquisición de conocimiento se pueden citar los métodos basados en interacción humana, tales como entrevistas, y los basados en técnicas de aprendizaje automático.
- *Representación y Modelado del Conocimiento*: Luego de adquirir el conocimiento, éste debe ser llevado a una forma entendible por las personas y sistemas que vayan a utilizarlo. El modelado del conocimiento tiene como propósito dar forma manipulable a los distintos tipos de conocimientos del dominio que maneja el experto. El conocimiento se debe estructurar y formalizar, para que el sistema que vaya a procesarlo lo haga bajo el mismo formato que lo haría el experto del dominio. El conocimiento puede representarse mediante diversos paradigmas, como es el caso de las redes semánticas, los marcos, las ontologías, etc. Este proceso es el que permite la construcción de la base de conocimientos del sistema.
- *Manipulación y Validación del Conocimiento*: Lo más importante de este proceso es que el conocimiento que se adquirió del experto, de las diversas fuentes, y la representación de éste, sea igual a la realidad. Por lo tanto, es

importante antes de empezar a utilizarlo, hacer las pruebas y validaciones pertinentes.

2.1.2. Ontologías

En las últimas décadas se empezó a utilizar el concepto de ontología en el área de Inteligencia Artificial, específicamente para la representación de conocimiento y la lingüística computacional [6]. Hoy día ha tenido un nuevo impulso por el deseo de transformar la red no sólo en un espacio de información, sino también en un espacio de conocimiento, con el desarrollo de la Web Semántica⁹. Se puede afirmar que para que los sistemas computacionales puedan razonar, deben apoyarse en ontologías (la formalización de la conceptualización asociado al dominio) [19]. Una máquina no puede comprender la información que procesa; sin embargo, si los datos que procesa están caracterizados semánticamente mediante ontologías, la Inteligencia Artificial puede simular la forma de razonar y la inteligencia humana [47].

Para la Ingeniería de Conocimiento, una ontología es *“Un vocabulario de términos y relaciones con las que se puede modelar un dominio”*. Según [43], la definición más aceptada en el área de inteligencia artificial fue propuesta por Gruber, que define una ontología como *“una representación explícita de una conceptualización”* [1]. Esta definición fue complementada por Borst, que la define como *“una especificación explícita y formal de una conceptualización compartida”* [2].

Una *“conceptualización”* es un modelo abstracto de algún fenómeno del mundo, construido mediante la identificación de los conceptos relevantes a ese fenómeno. *“Explícito”* significa que los conceptos utilizados en la ontología, y las restricciones para su uso, están claramente definidos. *“Formal”* se refiere al hecho de que debe ser comprensible para las máquinas, es decir, estar expresada mediante una sintaxis que permita a un computador operar sobre ella. *“Compartida”* refleja la noción de que contendrá conocimiento consensuado y aceptado por un grupo de personas [42] [48].

De acuerdo a [43] [49], tenemos que el uso de ontologías permite:

- Una mejor comprensión de algún área del conocimiento.
- Compartir un entendimiento común de una estructura de información entre personas, aplicaciones o agentes de software.
- Posibilitar la reutilización de conocimiento de un dominio.
- Hacer explícitas suposiciones de un dominio.
- Analizar el conocimiento de un dominio.
- Permitir que el computador utilice el conocimiento en alguna aplicación.

⁹ La Web Semántica es una ampliación de la web que crea tecnologías para añadir significado a los datos a través de metadatos semánticos y ontologías, de manera que los datos sean entendibles por las aplicaciones informáticas [45].

- Lograr la interoperabilidad entre sistemas informáticos.
- Razonar automáticamente.

Los componentes básicos de una ontología, según [5] [4], son:

- *Clases*: son la base de la descripción del conocimiento en las ontologías, ya que describen los conceptos del Dominio.
- *Relaciones*: definen las relaciones entre los conceptos, algunos ejemplos son: parte-de, subclase-de, conectado-a, compuesto-por.
- *Axiomas*: afirmaciones que son siempre ciertas para un dominio.
- *Instancias*: para representar elementos/objetos/individuos específicos en una ontología.
- *Propiedades*: establecen características o atributos de las clases y de las relaciones.

Las ontologías se presentan como una buena solución para alcanzar interoperabilidad semántica, ya que permiten conceptualizar el conocimiento, generando formas comunes y compartidas de ver el mundo. Su importancia es tal, que ha aparecido recientemente una rama de la ingeniería dedicada al estudio de las actividades concernientes al proceso de desarrollo de las ontologías, a su ciclo de vida, a los métodos y metodologías para construirlas, y a las herramientas y lenguajes que las soportan, denominada *Ingeniería Ontológica* [4].

Para construir ontologías se necesitan lenguajes lógicos y formales para representarlas [48], por lo que se han desarrollado numerosos lenguajes para este fin, dentro de los cuales se destacan: XML, XML Schema, RDF¹⁰ (“*Resource Description Framework*”, por sus siglas en Inglés), y OWL estos últimos propuestos por el World Wide Web Consortium (W3C) [50] [51].

En cuanto a herramientas para la construcción de ontologías, en los últimos años han aparecido muchos entornos para la construcción de ontologías, entre los que podemos mencionar: ONTOLINGUA [52] y PROTÉGÉ [53].

Finalmente, existen varias metodologías que describen las actividades propias de la planificación, desarrollo y evaluación de ontologías, tales como: METHONTOLOGY [54], TOVE [55] y KACTUS [56]. Una de las más comúnmente usadas es METHONTOLOGY, la cual consisten en los siguientes pasos para el desarrollo de las ontologías:

- *Especificación*: Se desarrolla un documento que contenga la meta de la ontología, nivel de granularidad, alcance, propósito. Se identifican los términos a ser representados, sus características y relaciones.
- *Conceptualización*: Se organiza el conjunto de términos y sus características en una representación intermedia, que el desarrollador de la ontología y los expertos puedan entender. En este paso se construye un glosario de

¹⁰ RDF es una familia de especificaciones como un modelo de datos para metadatos. Usado como un método general para la descripción conceptual o modelado de la información en la web [50].

términos, diagramas de relaciones binarias, diccionario de conceptos, tablas de atributos, tabla de axiomas lógicos, tablas de constantes, tablas de instancias.

- *Adquisición de conocimiento*: Normalmente se realiza en tres etapas: reuniones preliminares con los expertos, análisis y revisión de la bibliografía asociada al dominio, y una vez que se tiene un conocimiento base, se refina y detalla el conocimiento hasta completar la ontología.
- *Integración*: Se identifican las ontologías candidatas a ser reutilizadas en la ontología que se está construyendo, incorporando aquellas piezas de conocimiento que sean de utilidad.
- *Implantación*: Consiste en la codificación del modelo conceptual en un modelo codificado, usando alguna herramienta, como ONTOLINGUA, y un lenguaje como OWL.
- *Evaluación*: Se realiza un juicio técnico a la ontología, al ambiente de software asociado, y a la documentación, con respecto a un esquema de referencia.
- *Documentación*: Se detallar clara y exhaustivamente cada paso completado y los productos generados.

2.1.3. Minería Ontológica

La minería ontológica se refiere a la extracción de patrones de comportamiento ontológico y a la generación de modelos de conocimiento, desde un conjunto de ontologías, para lograr un dominio de conocimiento más amplio [10]. Actualmente, debido a la gran cantidad de ontologías disponibles, es necesaria la minería ontológica. Algunas de las técnicas que se han venido desarrollando son de enlazado, mezcla y alineamiento entre ontologías [57]. A través de los procesos de minería ontológica se descubre conocimiento a partir de estructuras y conceptos que conforman ontologías, para actualizar y enriquecer ontologías o construir nuevas ontologías [10].

La minería ontológica forma parte del área de minería semántica, cuyo objetivo es extraer conocimiento semántico desde diferentes fuentes como: documentos, bases de datos, contenido estructurado y no estructurado en la web, grafos anotados, ontologías, entre otros. La minería semántica se divide en tres grandes grupos: minería de datos semántica, minería web semántica y minería ontológica. La minería de datos semánticos consiste en extraer información con significado de un conjunto de datos semánticos, estructurados o no; y la minería de la web semántica consiste en extraer información con significado de la web, documentos html, xml, entre otros [7].

Una de las tareas fundamentales de la minería ontológica es el proceso de integración. La Integración de Ontologías puede ser vista como el proceso de encontrar puntos en común entre dos ontologías diferentes A y B, para la obtención de una nueva ontología C, que facilite la interoperabilidad entre los sistemas que se basan en las ontologías A y B. La nueva ontología C puede sustituir a A o B, o puede

ser utilizada como intermediario entre un sistema que se basa en A y un sistema que se basa en B [58]. Se pueden distinguir diferentes procesos en la integración de ontologías:

- *Alineación de Ontologías*: consiste en el análisis de correspondencias entre dos o más ontologías, realizando comparaciones entre los conceptos de las ontologías. El objetivo de alinear ontologías es encontrar relaciones entre las entidades expresadas en diferentes ontologías. Estas relaciones a descubrir son de equivalencia, y se deben determinar a través de medidas de similitud entre estas entidades. Estas medidas pueden ser de similitud léxica, que se basa en la similitud de dos expresiones en base a su escritura; similitud semántica, que se refiere a la similitud en cuanto a su significado; o similitud estructural, donde se determina la similitud en base a la taxonomía de la ontología (similitud de elementos vecinos: padres, hermanos e hijos). La alineación es un proceso previo para la mezcla y el enlazado [59], y la similitud se calcula normalmente con respecto a los conceptos y propiedades.
- *Mezcla de Ontologías*: Es el proceso donde varias ontologías dentro de un mismo dominio se unen, para estandarizar el conocimiento, hacer crecer el conocimiento, o tener el conocimiento total de manera local, entre otras cosas. A través de la mezcla o fusión, se unen ontologías que manejan el mismo conocimiento, pero con diferentes representaciones, o que poseen representaciones parciales de dicho conocimiento, donde las ontologías pueden coincidir en ciertos conceptos y en otros no [13]. La unión debe ser cuidadosa, ya que se pueden presentar repeticiones, expresiones disjuntas, distinto nivel de detalle, y contradicciones. La mezcla puede ser “fuerte”, si no deja conocimiento por fuera sin ser mezclado, o puede ser mezcla “débil”, si se deja conocimiento por fuera sin ser mezclado [57]. Para realizar el proceso de mezcla débil, se debe alinear cada concepto de una ontología A con otra ontología B. Si el concepto es encontrado en B, se completa su definición agregando en A todas las propiedades y relaciones que tenga este concepto en B [60].
- *Enlazado de Ontologías (Mapeo)*: Es el proceso de encontrar relaciones entre entidades que pertenecen a diferentes ontologías, para crear una conexión entre estas ontologías sin necesidad de mezclarlas. Esto puede realizarse con la creación de una ontología intermedia, que permite la navegación entre las ontologías que se están enlazando, o identificando las entidades comunes en ellas que sirvan de enlace. En general, la alineación de ontologías permite crear los enlaces directos, usando las equivalencias entre los conceptos y sus propiedades, encontradas en las distintas ontologías. Los resultados pueden ser usados para visualizar correspondencias, transformar una fuente en otra, crear un conjunto de relaciones o reglas entre las ontologías, o generar consultas para extraer

información desde las dos ontologías [40]. El proceso de enlazado de ontologías puede verse como un proceso de intersección de conjuntos.

Dentro del área de minería ontológica, los procesos dedicados a la integración de ontologías (enlace-fusión), donde se busca la incorporación de nuevos conceptos a una ontología, son usados como estrategia de aprendizaje ontológico. A continuación se explican las características y fases del aprendizaje ontológico.

2.1.4. Características del Aprendizaje Ontológico

El proceso de aprendizaje ontológico puede ser analizado de acuerdo a una variedad de enfoques, clasificado en diferentes dimensiones, conforme a los factores que intervienen en el proceso de aprendizaje. Los diferentes aspectos a considerar en un mecanismo de Aprendizaje Ontológico, de acuerdo a [61] [6], son:

- A. *Elementos Aprendidos*: Se refiere al tipo de estructura conceptual que es adquirida: conceptos, relaciones (taxonómicas ó no taxonómicas), axiomas, reglas, instancias.
- B. *Punto de Partida*: Hace referencia al punto de donde comienza la adquisición ontológica. Los sistemas de aprendizaje ontológico utilizan conocimiento previo, que varían en cantidad y calidad, para adquirir nuevos elementos de conocimiento. Los diferentes puntos de partida son:
 - a. *Antecedentes o Conocimiento previo*: pueden ser recursos lingüísticos u ontológicos.
 - b. *Entradas*:
 - i. *De acuerdo al Tipo*: las entradas de las cuales se adquiere el conocimiento ontológico, pueden ser:
 1. *Estructuradas*: Bases de Datos, ontologías, bases de conocimiento.
 2. *Semi-estructuradas*: Diccionarios, documentos HTML, XML, DTDs.
 3. *No-estructuradas*: es la entrada más difícil para extraer conocimiento, por las restricciones y características propias del lenguaje. Pueden ser textos en lenguaje natural o textos web.
 - ii. *De acuerdo al Lenguaje*: Se refiere al idioma. Existen sistemas de aprendizaje en español, inglés, alemán y francés, entre otros.
- C. *Pre-procesamiento*: Esta dimensión hace referencia a si existe algún pre-procesamiento para convertir la entrada en una estructura adecuada para aprender. Por ejemplo, en el aprendizaje de texto es muy usado el pre-procesamiento lingüístico¹¹, que elimina siglas, abreviaturas, etc.

¹¹ Aplicación de técnicas de lingüística computacional, a fin de depurar los datos y de conformar un conjunto de características enriquecidas que constituyan una representación adecuada del texto para que pueda ser procesado [108]

D. *Método de Aprendizaje*: esta dimensión se refiere al método usado para extraer conocimiento, el cual, a su vez, se clasifica de acuerdo las siguientes dimensiones:

a. *Categoría del Aprendizaje*: aquí se consideran dos categorías: supervisado y no supervisado, y en línea y fuera de línea.

b. *Técnica de Aprendizaje*: se puede clasificar en:

- i. *Estadístico*: estas técnicas se basan en la realización de análisis estadísticos a los datos de entrada, como por ejemplo, análisis de probabilidad, frecuencias, distribución de términos, entre otros.
- ii. *Lógicos*: se refieren a métodos de aprendizaje basados en lógica, para descubrir nuevo conocimiento mediante deducción o inducción, tales como aprendizaje inductivo, aprendizaje proposicional, entre otros.
- iii. *Lingüísticos*: los enfoques lingüísticos, tales como análisis sintáctico, análisis morfosintáctico, análisis de patrones léxico sintácticos, procesamiento semántico y comprensión de texto, podrían ser utilizados para extraer conocimiento ontológico a partir de textos en lenguaje natural.
- iv. *Basado en patrones/plantillas*: este enfoque es muy usado en la extracción de información de texto, donde por ejemplo, se puede realizar búsquedas con palabras claves pre-definidas. Hay varios tipos de plantilla o patrones: sintácticos, semánticos, léxicos-sintácticos, o de propósito general, para extraer elementos ontológicos: relaciones taxonómicas y no taxonómicas, axiomas.
- v. *Basados en Heurísticas*: este método puede ser usado como soporte a otros enfoques, y consiste en definir un conjunto de reglas para evaluar condiciones y ejecutar acciones.
- vi. *Híbridos o Múltiples estrategias*: muchos sistemas de aprendizaje que aprenden varios tipos de elementos, deben usar varias técnicas de aprendizaje, como por ejemplo, combinar métodos lógicos y lingüísticos, ó heurísticos con estadísticos, dependiendo de los elementos a aprender.

c. *Tareas de Aprendizaje*: los métodos de aprendizaje también se pueden clasificar de acuerdo a las tareas que realizan dentro del proceso de aprendizaje. Estas pueden ser de: Clasificación, Agrupamiento, Aprendizaje de reglas, Análisis de Conceptos.

E. *Dirección del Aprendizaje*: el proceso de aprendizaje puede llevarse a cabo de varias maneras:

a. *Ascendente (de abajo hacia arriba)*: comienza el aprendizaje desde los niveles bajos, por ejemplo documentos de texto, para gradualmente formar la ontología en un nivel superior. Se identifican las clases más

específicas, y a continuación, la agrupación de éstas en otras clases más generales.

- b. *Descendente (de arriba hacia abajo)*: el proceso de aprendizaje va desde ontologías superiores, muy generalizadas, para formar ontologías más especializadas. Se puede recurrir a una aproximación, comenzando con la definición de los conceptos más generales en el dominio, para a continuación, extraer conceptos más específicos.
- c. *Híbridos*: una combinación de los métodos anteriores.

F. *Grado de Automatización*: se refiere a si el proceso de aprendizaje es manual, automático o semi-automático.

G. *Resultado*: esta dimensión tiene que ver con el resultado del proceso de aprendizaje, es decir, lo que se construye. Algunos sistemas de aprendizaje construyen ontologías, otros solamente construyen estructuras intermedias que sirven de apoyo a los usuarios, expertos, u otros sistemas de aprendizaje.

H. *Evaluación*: aunque no está definido un método formal y estándar para evaluar los sistemas de aprendizaje ontológico, en esta dimensión se consideran dos aspectos, la evaluación del proceso de aprendizaje, o la evaluación del resultado del proceso. Es más común evaluar la ontología resultante, comparando dos o más ontologías modeladas dentro de un mismo dominio, o a través de la aplicación en la que se emplean.

I. *Dominio*: considerar el tipo de dominio donde se desarrolla el proceso de aprendizaje es importante, para determinar el tipo de estrategia a aplicar. La selección del enfoque de aprendizaje ontológico debe tomar en cuenta las características del dominio en consideración. Las perspectivas para caracterizar los dominios son:

- a. *Desarrollo del Dominio*: si la ontología en el dominio ya ha sido creada, los enfoques “Alto-Conocimiento” y “Descendente” son adecuados para el aprendizaje ontológico, ya que pueden existir ontologías superiores ya definidas que se pueden especializar. Sin embargo, si el dominio no se ha desarrollado y carece de recursos específicos, los enfoques “Bajo-Conocimiento” y “Ascendente” deben ser preferidos en el aprendizaje ontológico, para comenzar el proceso desde abajo. También están los dominios Emergentes, que son los nuevos dominios que emergen constantemente, impulsados por los avances científicos y tecnológicos o por las necesidades. Tales dominios generalmente aún no tienen ontologías. En estos casos, es probable que el acceso a fuentes de información fidedignas sea limitado. De esta manera, un enfoque

ascendente puede ser una buena alternativa para este aprendizaje ontológico.

- b. *Tecnología en el Dominio*: en los dominios con tecnología avanzada, el conocimiento se actualiza a un ritmo mucho más rápido que en los dominios de tecnología baja. En estos casos, un enfoque híbrido (Ascendente y Descendente), que permite una actualización incremental del conocimiento, es el más adecuado en los dominios de tecnología avanzada.
- c. *Dependencia del Dominio*: a diferencia de los dominios autónomos, los dominios interdisciplinarios son propensos a ambigüedades en los términos empleados. Por ejemplo, “ontología” en filosofía tiene un significado diferente que en inteligencia artificial. Como resultado de esto, un enfoque ascendente no puede converger a ontologías de alto nivel en ámbitos interdisciplinarios. Una mejor alternativa sería comenzar con la definición de la ontología de nivel superior, y luego, continuar construyendo la ontología completa.

2.1.5. Fases del Aprendizaje Ontológico

En el aprendizaje ontológico se integran diversas disciplinas, especialmente de aprendizaje automático, para facilitar la construcción de ontologías. En general, la adquisición de conocimiento actualmente no se realiza de forma completamente automática, tal que el aprendizaje ontológico es habitualmente un proceso semi-automático [48].

En [49] se proponen unas fases del proceso del aprendizaje ontológico, considerándola como un proceso semi-automático que requiere la intervención humana, basada en un modelo cooperativo balanceado¹². En la Fig. 2.1 se muestran las fases del proceso de aprendizaje, las cuales se explican a continuación [62] [49] [48]:

- *Importación y reutilización*: Tiene por objetivo desarrollar mecanismos y estrategias para importar y reutilizar conceptualizaciones existentes de un dominio (esquemas de bases de datos, definiciones de tipos de documentos ó DTD, ontologías).
- *Extracción*: En esta fase se definen los términos que provienen de recursos seleccionados de un dominio dado. Dichos términos pueden venir de diferentes fuentes (textos, documentos html, etc.), y pueden ser propuestas por herramientas de forma automática, o de forma manual por

¹² El modelo Cooperativo Balanceado se refiere a que cada tarea puede ser realizada por el usuario o por alguna herramienta de aprendizaje del sistema [107].

el ingeniero ontológico¹³. Luego se realiza su clasificación taxonómica (por ejemplo, mediante técnicas automáticas de clasificación o agrupamiento, que establecen la similitud entre los elementos en base a sus propiedades comunes). Como resultado, se tiene la propuesta de una taxonomía del dominio, que puede ser modificada por el ingeniero ontológico.

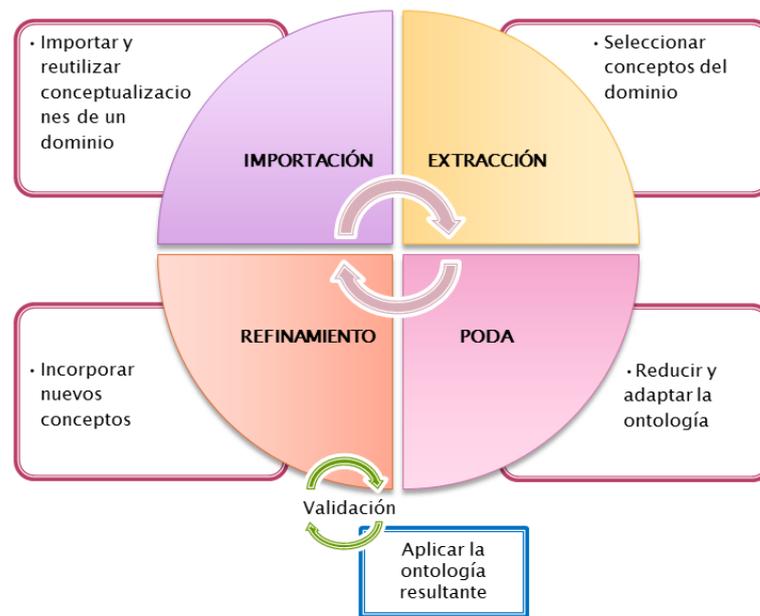


Figura 2.1. Fases del Aprendizaje Ontológico

- *Poda*: el boceto de la ontología final, resultante de la importación, reutilización y extracción, es reducida para adaptar la ontología a su propósito principal. El ingeniero ontológico puede eliminar o mantener conceptos, cuando sea conveniente. La poda o reducción de la ontología se basa en la determinación de la frecuencia de aparición de los conceptos en los documentos del dominio. Las entidades que son frecuentes en las fuentes se consideran elementos fundamentales del dominio, en contraste con los términos que sólo aparecen una vez, y que han sido considerados en la fase de extracción. El ingeniero ontológico puede determinar la medida de relevancia para la determinación de la frecuencia. El algoritmo de poda de la ontología utiliza las frecuencias calculadas para determinar la importancia de cada concepto contenido en la ontología. Todos los conceptos y las relaciones existentes, que son más frecuentes en los documentos del dominio específico, permanecen en la ontología.

¹³El Ingeniero Ontológico realiza actividades inherentes al proceso de desarrollo de una ontología, el ciclo de vida de una ontología, así como a las metodologías, herramientas y lenguajes necesarios para la construcción de ontologías [2].

- *Refinamiento*: esta fase se basa en los ajustes a la ontología final. El refinamiento trata de incorporar nuevas entradas léxicas o conceptos, de acuerdo a las necesidades específicas de los usuarios del dominio, o a nuevas actualizaciones del dominio.
- *Validación*: Finalmente, la aplicación destino sirve como medida para validar la ontología resultante. Sólo a través del seguimiento y uso, se puede evaluar realmente la ontología, para así determinar si es necesario actualizarla para incluir nuevos aspectos del dominio. Entonces, el ciclo puede iniciarse nuevamente.

2.2. AUTO-ORGANIZACIÓN Y EMERGENCIA

Algunos autores confunden o asocian la emergencia con la auto-organización, incluso tratándolos a veces como sinónimos, sin embargo, ambos conceptos se complementan. Como se afirma en [9], la esencia de la emergencia es la existencia de un comportamiento global nuevo, diferente del de los elementos constitutivos del sistema, mientras que la esencia de la auto-organización es un comportamiento adaptable, que de forma autónoma adquiere para mantener un orden. Al diseñar e implementar SAOyE, se indica en [63] que una herramienta idónea para modelarlos son los sistemas multiagente (SMAs), que describiremos a continuación.

2.2.1. Sistemas Multiagente

En los SAOyE un número de agentes se agrupan y forman sociedades de agentes, tal que estos últimos trabajan juntos para resolver problemas que van más allá de sus capacidades individuales. Allí, las dinámicas de interacción entre los agentes y/o con el entorno no son predefinidas, y favorecen la aparición de un comportamiento emergente a nivel colectivo en el SMA.

Un SMA se puede definir como una red de solucionadores de problemas (agentes) con un nivel muy bajo de acoplamiento, que trabajan conjuntamente, lo que posibilita que se enfrenten a problemas más complejos que los abordables de forma individual [64].

Un “Agente” es un sistema computacional que está situado en un ambiente, y que es capaz de tomar acciones autónomas en ese ambiente, con el fin de cumplir sus objetivos de diseño [65] [64]. La autonomía es la característica principal que describe a un agente, entendiendo como autonomía la capacidad del agente de actuar sin la intervención de un usuario o de otro sistema. De esta manera, los Agentes Inteligentes, además de atributos y métodos (propiedades del paradigma de orientación por objetos), poseen creencias, deseos e intenciones que los vinculan con su entorno, y estados mentales de los cuales depende su comportamiento.

Algunas propiedades de los agentes son las siguientes [64] [66] [65]:

- *Autonomía*: es la capacidad de tener motivaciones propias a partir de las cuales generan autónomamente sus objetivos. La autonomía le da la capacidad de actuar y reaccionar a los estímulos externos basándose en su estado interno. Los agentes son autónomos en la medida en que actúan sin la intervención humana ni de otros sistemas externos. La proactividad está estrechamente ligada con la autonomía de los agentes.
- *Comunicación*: La capacidad de cada agente de conversar utilizando un lenguaje basado en ontologías y realizar intervenciones asíncronas, constituye un paso adelante en llevar el concepto real de conversación al ámbito computacional. La comunicación entre agentes puede ser directa o indirecta. Para comunicarse directamente, los agentes realizan intervenciones, traducidas en mensajes, donde se especifican los datos del agente emisor, datos de los agentes receptores, datos de la intervención, tales como el lenguaje, la ontología y el protocolo utilizado, y el contenido. El contenido incluye una performativa que indica la acción general del mensaje, y una frase compuesta por elementos pertenecientes a la ontología usada, que indica sobre qué aspectos habla la performativa. Cada agente comprende esta intervención, en consecuencia, afecta su estado interno, y eventualmente reaccionará con un conjunto de intervenciones. La comunicación indirecta hace uso del ambiente donde los agentes se desenvuelven, y está basada en variables compartidas por todos los agentes.
- *Racionalidad*. Los agentes tienen un conjunto de objetivos predefinidos, y emprenden acciones para conseguirlos. La decisión de cual acción seguir y en qué momento hacerlo, es definida según un principio de racionalidad, basada en la idea de que prefieren la acción más prometedora o eficiente para conseguir sus metas.
- *Inteligencia*. Es la capacidad de acumular conocimiento (aprender) y de usarlo adecuadamente (razonar a partir de él). Esta propiedad es necesaria, para que un agente analice y ordene las ideas y el conocimiento que posee sobre el entorno, para llegar a una conclusión, y tomar una acción de forma autónoma. Para implementar esta característica se pueden utilizar técnicas computacionales inteligentes, como los sistemas expertos, redes neuronales artificiales, lógica difusa, etc.
- *Razonamiento*. Es la capacidad de explotar el conocimiento para inferir conclusiones, hipótesis, nuevo conocimiento, entre otras cosas. Las formas clásicas de razonamiento son la inducción, la deducción y la abducción.

- *Reactividad.* Los agentes perciben su entorno respondiendo a los cambios que ocurren en él. La capacidad de emitir una acción inmediata al recibir una señal o percibir un estado en el ambiente, es lo que caracteriza a los agentes reactivos. La reactividad en los agentes posibilita acciones rápidas, cruciales en sistemas de tiempo real, que no ameritan aplicar reglas complejas.
- *Sociabilidad.* Esta capacidad está muy relacionada con el aspecto comunicacional. Los agentes interactúan con otros agentes mediante algún tipo de comunicación y convenios colectivos, y es determinante en la eficiencia de las sociedades de agentes. Una sociedad de agentes es un grupo de agentes que interactúan, se comunican, conversan, “piensan” y actúan en conjunto, para lograr un objetivo común. La colaboración y la competencia son esquemas de la sociabilidad de los agentes.

Los SMA son sistemas que describen a los agentes en un entorno social, en el cual dichos agentes cooperan para lograr tanto sus metas individuales como las metas colectivas de la comunidad multiagente [64]. Un SMA está formado por un grupo (comunidad) de agentes que interactúan entre sí, utilizando protocolos y lenguajes de comunicación de alto nivel, para resolver problemas que van más allá de las capacidades o del conocimiento de cada uno [65].

Entre los aspectos más resaltantes de los SMA se tienen que [67] [65]:

- Cada agente tiene información incompleta, o no es capaz de resolver los problemas completos, es decir, tiene conocimiento limitado del problema.
- No existe un sistema de control global.
- Los datos están descentralizados.
- El procesamiento de la información es asíncrono.
- La coordinación es necesaria en la comunidad de agentes, para que puedan actuar colectivamente.

En particular, la coordinación es la articulación de las acciones individuales realizadas por cada uno de los agentes, de manera que en conjunto tengan éxito. La coordinación de acciones es necesaria, ya que los agentes necesitan información y resultados que otros agentes pueden suministrar. Además, los recursos son limitados, por lo tanto, tienen que ser compartidos. Todo lo anterior, permite que los agentes tengan objetivos diferentes, pero interdependientes, y así realizar trabajos en conjunto.

Algunas características representativas de los SMA son las siguientes [68]:

- *Modularidad.* Los módulos (entidades, subsistemas) de un sistema son representados por los agentes. Dependiendo de la granularidad del modelo, cada uno de esos módulos puede estar compuesto de módulos más pequeños. El punto de vista modular permite la reconfiguración y extensibilidad del SMA de una manera fácil.

- *Redundancia.* Un SMA consiste generalmente de un gran número de agentes, donde en principio cada uno hace su labor, pero si es necesario, pueden existir varios con características que los hacen similares en función y diseño. Esto significa, por un lado, que las instancias críticas no son representadas por un solo agente, y por otro lado, que el sistema no se cae si un agente falla de alguna manera, brindándole robustez al sistema.
- *Descentralización.* Un SMA no es regido por un control centralizado. En lugar de eso, las competencias y capacidades, entre otras cosas, son distribuidas entre los diversos agentes.
- *Comportamiento Emergente.* En un SMA, la interacción entre los agentes puede producir un comportamiento nuevo (y estable) en el nivel global de todo el sistema. Esta cualidad resulta del comportamiento agregado de los agentes, y no puede ser reducido a los agentes individuales.
- *Funcionalidad.* Aunque cada agente puede tener sus propias funciones o comportamientos, la funcionalidad del sistema como un todo, por ejemplo, la resolución de un problema, no es asignado a agentes específicos, sino que resulta de la interacción de los diferentes agentes.
- *Adaptación.* La modularidad, la descentralización y la funcionalidad emergente, son las bases para que el SMA se adapte a situaciones cambiantes. La adaptación (algunas veces llamada aprendizaje colectivo) también necesita que el sistema pueda olvidar/desaprender sus viejos estados, interacciones, entre otras cosas, para adaptarse a nuevas situaciones. Un ejemplo de lo anterior, es a través del refuerzo y evaporación de la feromona, utilizado en los sistemas basados en el comportamiento de las colonias de hormigas, o en los mecanismos de recompensa y castigo en los sistemas colaborativos, entre otros.

Los SMA deben ser flexibles, robustos y adaptables a las circunstancias, para poder trabajar en ambientes heterogéneos y dinámicos, y ser capaces de enfrentar los cambios requeridos. Muchas aplicaciones que requieren cierto grado de auto-organización y un comportamiento emergente, son diseñadas bajo un enfoque de SMA. En este enfoque, el SMA se diseña en base a agentes reactivos que utilizan intensamente el ambiente. Este enfoque se basa en los siguientes principios [63]:

- Los agentes son simples, reactivos, e inconscientes de otros agentes, o de las actividades complejas que emergen de la sociedad de agentes.
- El ambiente es un mecanismo importante para dirigir las actividades de estos agentes, y para acumular información sobre las actividades en curso de la sociedad entera de agentes.
- La coordinación de actividades a través de la comunicación directa, es remplazada por interacciones indirectas que permiten tener agentes más simples y reducir los requerimientos de comunicación entre agentes.

Estos principios son base de la computación emergente, la cual tiene como producto global una inteligencia a nivel colectivo. Estos aspectos son presentados en la siguiente sección.

2.2.2. Computación Emergente

La Computación Emergente se enfoca en la aplicación de técnicas computacionales no convencionales, para la resolución de problemas que debido a la complejidad que presentan, no pueden ser abordados utilizando métodos tradicionales. Actualmente se incluye en la computación emergente a los autómatas celulares, las redes neuronales, los algoritmos genéticos, los algoritmos bio-inspirados, y los SMAs [69].

Una disciplina clave del área de computación emergente es la “Inteligencia Colectiva”, una forma de inteligencia que surge de la colaboración de muchos individuos, generalmente de una misma especie. La inteligencia de enjambre (“Swarm Intelligence” en inglés), o también llamada “Inteligencia Colectiva”, es un campo de investigación científica, multidisciplinaria, que se interesa en los procesos distribuidos (no supervisados) de una organización, y como este modelo está presente en un cierto número de sociedades animales, para resolver problemas complejos en ellas [70]. Se han propuesto un gran número de sistemas artificiales bio-inspirados para diferentes cosas: resolver problemas de optimización, coordinación de robots, auto-ensamblaje, organización de bases de datos, protección de virus, entre otras [63].

Los sistemas realizados bajo el enfoque de la inteligencia colectiva se caracterizan por exhibir características que hacen a las sociedades de insectos exitosas en su entorno, tales como: flexibilidad, robustez, control descentralizado y auto-organización. Particularmente, cuando se habla de inteligencia colectiva, se tratan dos aspectos claves: la auto-organización y la interacción indirecta [70].

- *Auto-organización.* La característica principal de estos sistemas es su capacidad de lograr tareas colectivas complejas sin un control central. Requiere de cuatro insumos [70]:
 - a. *Retroalimentación Positiva:* promueve la creación de estructuras a través de reglas de comportamiento simples. Por ejemplo: el rastro dejado por las hormigas a través de la feromona, o por la danza en el caso de las abejas.
 - b. *Retroalimentación Negativa:* permite el equilibrio con la retroalimentación positiva, y ayuda a estabilizar el patrón colectivo, puede tomar la forma de saturación, agotamiento o competición. Por ejemplo: la evaporación de la feromona.

- c. *Amplificación de las Fluctuaciones (Aleatoriedad)*: La aleatoriedad es frecuentemente crucial, ya que permite el descubrimiento de nuevas soluciones, que pueden actuar como semillas, para que nuevas estructuras se desarrollen y se fortalezcan (emerjan). Por ejemplo, una hormiga que se pierda siguiendo el rastro de sus compañeras. Aunque este fenómeno puede parecer ineficiente, esto favorece la exploración de nuevas áreas, para encontrar fuentes de alimento no explotadas.
 - d. *Múltiples Interacciones*: La auto-organización generalmente requiere de una densidad mínima de individuos tolerantes mutuamente. Estos individuos deben ser capaces de hacer uso de los resultados de sus propias actividades, como también de las actividades de los otros. Por ejemplo, redes de rastros pueden auto-organizarse y ser usadas colectivamente, si los individuos usan la feromona de los otros.
- *Interacción Indirecta ó Estigmergia*. Es un mecanismo usado para reflejar las interacciones entre los agentes y el entorno. Este mecanismo es un poderoso principio de cooperación en las sociedades de insectos. Ha sido observado dentro de muchas sociedades de insectos, como las avispas, abejas y hormigas. Está basado en el uso del entorno como medio de inscripción de los efectos de comportamientos pasados, para influenciar el futuro. Por ejemplo, las hormigas permanentemente interactúan con el entorno, siendo dirigidas por éste para hacer ciertas actividades y producir cambios en él, y esparcen feromona como un reflejo de sus actividades.

Algunos modelos y aplicaciones se han desarrollado explotando los beneficios de la inteligencia colectiva, como por ejemplo, el enrutamiento del tráfico de la red en sistemas de telecomunicaciones, problemas de transporte multi-robot, análisis y clasificación de datos, entre otros. Uno de los más utilizados es el Modelo de Optimización Basada en Colonia de Hormigas (ACO, “Ant Colony Optimization”, en inglés) [70]. Los algoritmos ACO se inspiran directamente en el comportamiento de las colonias reales de hormigas, para solucionar problemas de optimización combinatoria. Se basan en una colonia de hormigas artificiales, esto es, agentes computacionales simples, que trabajan de manera cooperativa y se comunican mediante rastros de feromona artificiales. Los algoritmos ACO son algoritmos constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema, recorriendo un grafo que contiene soluciones parciales. Cada arista del grafo, que representa los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga: la información heurística, que mide la preferencia heurística de moverse desde un nodo hasta otro, y la información de los rastros de feromona artificiales, que mide la “deseabilidad aprendida” de ese movimiento, que imita la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo, dependiendo de la calidad de las soluciones encontradas por las hormigas [70] [71].

2.2.3. Sistemas Auto-Organizados y Emergentes

Un sistema Auto-organizado y Emergente, de acuerdo a [8] [9], es aquel sistema que se basa en los principios de:

- *“Emergencia”*: se habla de emergencia cuando hay nuevas propiedades, comportamiento, estructura y patrones coherentes a nivel macro, que surgen dinámicamente de las interacciones entre las partes a nivel micro [72]. Estas propiedades, comportamiento, estructura, patrones, etc., son nuevas con respecto a las partes individuales del sistema [73]. La emergencia se asocia al comportamiento de una sistema que surge de las interacciones de sus componentes, difíciles o imposibles de predecir. Un clásico indicador de emergencia son los patrones observables en un nivel superior, con características temporales y espaciales específicas [9].
- *“Auto-Organización”*: La auto-organización se refiere a una organización espontánea producida dinámicamente. Es un proceso dinámico y adaptable donde los sistemas adquieren y mantienen la estructura por sí mismos, sin control externo. La "estructura" puede ser una estructura espacial, temporal o funcional. "No hay control externo" se refiere a la ausencia de dirección, manipulación, interferencia, presiones, o de participación desde fuera del sistema. Un sistema auto-organizado no sólo regula o adapta su comportamiento, sino que crea su propia organización [9].

En [8], se señala la importancia e interés reciente de la auto-organización y emergencia para gestionar recursos distribuidos. La Web es un ejemplo, ya que muchas aplicaciones sobre la Web obtienen contribuciones pequeñas, a muy bajo costo, de un gran y diverso grupo de colaboradores, para producir productos y servicios de información. Este es un modelo de producción auto-organizada y emergente, a diferencia de la manera centralizada de los sistemas de control. Como ejemplos de estos sistemas, tenemos en la vida real, la organización de las ciudades y las sociedades de insectos. En el área de computación, algunos ejemplos de sistemas con tendencia a trabajar de manera auto-organizada y emergente, los tenemos en: las redes sociales y Wikipedia. Los ambientes inteligentes también pueden presentar comportamientos emergentes y auto-organizados, al proporcionar servicios inteligentes a los usuarios del contexto, de una manera adaptativa y adecuada a sus necesidades [74]. Otros ejemplos de aplicaciones auto-organizativas emergentes son también la robótica distribuida, las aplicaciones para seguridad y gestión de redes de comunicación, sistemas de control de manufactura, infraestructuras para procesos de negocios, entre otras [63].

En [73] se señalan las siguientes características de la emergencia:

- *Efecto Micro-Macro*: El comportamiento global del sistema (es decir, las propiedades emergentes) es el resultado de las interacciones entre las distintas entidades del sistema.
- *Novedad Radical*: El comportamiento global es nuevo con respecto a los comportamientos individuales en el nivel micro, es decir, los individuos en el nivel micro no tienen una representación explícita del comportamiento global.
- *Coherencia*: se refiere a una correlación lógica y coherente de las partes.
- *Interacción de las Partes*: Las propiedades emergentes surgen de las interacciones entre las partes.
- *Dinamismo*: las propiedades emergentes surgen como parte de la evolución del sistema en el tiempo. Tal propiedad emergente es un nuevo tipo de comportamiento, que se hace posible en un momento determinado en el tiempo.
- *Control Descentralizado*: No hay control central, es decir, ninguna parte del sistema orienta el comportamiento a nivel macro.
- *Relación Bidireccional*: Del nivel micro al nivel macro, las partes dan lugar a una estructura emergente. En la otra dirección, la estructura emergente influencia sus partes, es decir, las propiedades a alto nivel tienen efectos causales en el nivel inferior.
- *Robustez y Flexibilidad*: La falla o reemplazo de una entidad no causará una falla general a nivel de las propiedades emergentes. Esta flexibilidad hace que las entidades individuales puedan ser reemplazadas, y que la estructura emergente pueda permanecer.

En [73] se señalan las siguientes características de la auto-organización:

- *Aumento del Orden*: la organización puede ser vista como un aumento en el orden del comportamiento del sistema, que permite al sistema adquirir una estructura espacial, temporal o funcional.
- *Autonomía*: es la ausencia de control externo. Un sistema debe organizarse sin interferencia del exterior.
- *Adaptabilidad o Robustez con respecto a los Cambios*: la robustez es usada en términos de la adaptabilidad en presencia de perturbaciones y cambio. Un sistema auto-organizado se espera que haga frente a ese cambio, y mantenga su organización autónomamente.
- *Dinamismo*: la auto-organización se trata de un proceso. Con el tiempo hay un aumento en orden, es decir, una dinámica hacia más orden. En relación a la adaptabilidad requerida en un contexto en rápida evolución, un comportamiento auto-organizado necesita ser dinámico.

Capítulo 3:

ESQUEMA DE EMERGENCIA ONTOLÓGICA (EEO)

El esquema de aprendizaje ontológico propuesto, lo llamaremos “Esquema de Emergencia Ontológica” (EEO). Antes de describir en detalle las características del esquema propuesto, se define el concepto de “Emergencia Ontológica”(EO).

3.1. La Emergencia Ontológica

Las “Emergencia Ontológica” se puede definir como un proceso que busca adaptar una ontología a los cambios y nuevos componentes de un dominio dinámico, emergente y auto-organizado, a través de la aplicación de un conjunto de reglas, que permiten de manera global el surgimiento de una nueva conceptualización (conceptos emergentes). La generación de esta nueva ontología se realiza de forma iterativa, realizando las adaptaciones de nuevos conceptos y de su estructura en base a unos criterios de relevancia y entropía, previamente definidos.

Una “Ontología Emergente”, puede surgir a partir de procesos de análisis e integración de diversas fuentes de datos: tablas, documentos de texto y xml entre otros; o incluso, a través del análisis y procesamiento de datos captados por sensores. También puede surgir del enriquecimiento, integración o extensión de ontologías, a través de diversas tareas de minería ontológica (para más detalles, ver sección 2.1.3 del marco teórico). En general, estos procesos de extracción de información son los que alimentan el proceso de emergencia ontológica a través del procesamiento de las fuentes de datos.

3.2. Descripción de la Propuesta

El EEO trata de representar en una ontología una conceptualización de una realidad, que es cambiante de acuerdo al contexto, que no se puede pre-definir o pre-determinar. Una ontología emergente, por lo tanto, tendrá una validez determinada, es decir, el modelo conceptual resultante es válido para un contexto determinado y concreto, y surge en un momento dado para un dominio específico, a partir de la integración de anotaciones semánticas, ontologías y meta-ontologías, reflejando un modelo conceptual acorde a las actuales necesidades y dinámica del sistema. El conocimiento emergente resultante reflejado en la OE vuelve a ser usado por el sistema, y modificado

continuamente, lo que permite su adecuación a la realidad cambiante, tal que este conocimiento no es estático, fijo, absoluto, ni completo.

La emergencia ontológica no puede ser vista como un proceso aislado y mecánico. Para que sea posible la emergencia ontológica, es necesario llevar a cabo una serie de tareas que se deben ir realizando de acuerdo a la dinámica del sistema, siguiendo un conjunto de reglas definidas, enmarcadas en un proceso general, donde se integran y complementan, para dar como resultado una nueva conceptualización emergente.

En la literatura se pueden encontrar algunas propuestas para estructurar procesos de evolución ontológica, de forma integral y controlada, posibilitando las actualizaciones o la generación de nuevas versiones de una ontología [21] [22] [24] [26] [27]. Sin embargo, no contemplan la posibilidad de que los conceptos puedan surgir de forma dinámica, y que las ontologías puedan ser adaptadas de acuerdo a los cambios del sistema, en este caso, del SAOyE. La propuesta del EEO va orientada a una visión más amplia, donde se considera la evolución de una ontología influenciada por diferentes elementos dinámicos en el entorno (datos, eventos, aplicaciones, servicios, etc.), que aportan nuevos datos, que deben ser caracterizados y clasificados para contribuir al enriquecimiento de la Ontología Emergente (OE). Una conceptualización emerge a partir del análisis de las estructuras existentes, de los nuevos requerimientos e información de contexto, que necesitan ser representados en el marco ontológico del sistema. Los procesos clásicos de actualización, integración, validación y verificación de ontologías, pueden ser reutilizados e integrados en este proceso dinámico, para complementar el proceso de emergencia ontológica, y así modelar y representar ontológicamente el conocimiento y la dinámica de los sistemas con características de auto-organización y emergencia, que son los que se quieren abordar en esta propuesta. Las meta-ontologías van a ser determinantes en la propuesta, ya que permitirán la integración de los conceptos emergentes. Estas meta-ontologías son requeridas en el proceso, y se van a ir estructurando, a medida que van ingresando conceptos al proceso. También pueden ser generadas a partir del procesamiento de diversas ontologías de un mismo dominio, o ser previamente definidas por expertos.

3.3. Aspectos Generales del EEO

De acuerdo a los enfoques y factores que intervienen en el proceso de aprendizaje ontológico (Ver sección 2.1.4 del marco teórico), el EEO propuesto presenta las siguientes características:

- En cuanto a los elementos aprendidos, el esquema contempla el aprendizaje de conceptos, propiedades y relaciones taxonómicas.
- Las fuentes de datos usadas durante la adquisición ontológica, pueden ser estructuradas o semi-estructuradas.

- El idioma no es determinante, pero es importante que todas las entradas (fuentes) en un mismo proceso estén en el mismo idioma.
- Las entradas (fuentes) pueden pasar por un pre-procesamiento, para unificar criterios en cuanto a propiedades, formatos, etc. Puede consistir en eliminar propiedades menos relevantes, unificar formatos en la identificación de atributos y propiedades. Es realizado por el experto ontológico.
- Se emplean varias técnicas de aprendizaje como las estadísticas, el análisis de similitud y la definición de un conjunto de reglas. Entre las tareas que se realizan durante el aprendizaje están la clasificación y el agrupamiento.
- En cuanto a la dirección del aprendizaje, se realiza de forma híbrida: *ascendente* porque se realiza a partir de fuentes de datos desde donde se descubren meta-datos para definir las meta-ontologías, y *descendente* ya que se especializan ontologías a partir de meta-ontologías que contiene los conceptos generales de un dominio.
- El grado de automatización es semi-automático.
- Como resultado del proceso de aprendizaje, se construyen estructuras finales e intermedias que sirven de apoyo a los expertos ontológicos en el desarrollo de otras estructuras y aplicaciones.
- Las ontologías resultantes son evaluadas a través de métricas de calidad, las cuales son: Completitud, Robustez y Precisión.

3.4. Propiedades del EEO

En el proceso de emergencia ontológica propuesto son determinantes 2 propiedades: La “Relevancia” y la “Entropía” las cuales se describen a continuación.

3.4.1. La Relevancia

La relevancia de un concepto hace referencia a la importancia del concepto en un ámbito determinado y en un momento dado. Esta relevancia será el aspecto determinante, al momento de calificar un concepto como emergente. En [26], se resalta que los métodos actuales de evolución ontológica se enfocan, entre otras cosas, en la clasificación de los conceptos de acuerdo a su relevancia.

En el proceso de emergencia se considera que un concepto puede pasar por 3 estados: “Obsoleto”, “Candidato”, “Emergente” (Fig. 3.1). El pase por esos estados va a depender de la relevancia del concepto, es decir, que tan importante es el concepto para el sistema en un momento dado, en base a la frecuencia de aparición y del conjunto de instancias asociadas al concepto.

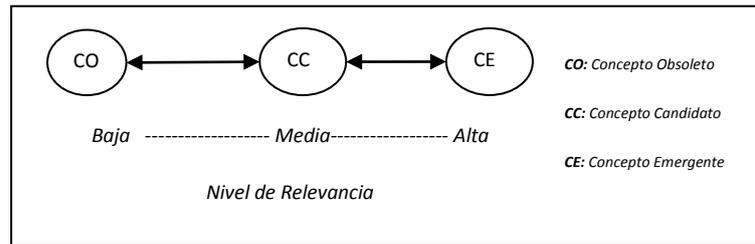


Figura 3.1. Estados de un Concepto

Entonces, la “Relevancia” de un concepto se refiere a su importancia, y se determina en base a la “Frecuencia” de uso y la cantidad de “Instancias” asociadas (para más detalles, ver Definiciones 10, 11 y 12 en la Sección 3.5).

3.4.2. La Entropía

Un criterio importante que se usa en la propuesta es la entropía. La Entropía hace referencia a la tendencia al desorden, al desgaste, al aumento de la aleatoriedad, derivada de los continuos ajustes que sufre un sistema [75]. En los sistemas abiertos, gracias a los recursos que se obtienen del ambiente, esta tendencia puede ser corregida y transformada en una organización más completa y estructurada. En base a eso, en [75] se afirma que la entropía alcanzará un máximo cuando el sistema se acerca al equilibrio.

La entropía es una medida natural de los Sistemas Auto-Organizados y Emergentes (SAOyE), para determinar su nivel de incertidumbre. Los SAOyE se caracterizan por el surgimiento de nuevos patrones, como consecuencia del dinamismo del sistema y de un proceso de adaptación al entorno. La auto-organización permite a los sistemas alcanzar de forma autónoma un orden, a través de una estructura o patrón emergente, como una manera de reducir espontáneamente la entropía de un sistema dinámico [76].

La entropía en el EEO, se refiere al desorden conceptual del sistema, que en este caso viene dado por meta-conceptos que agrupan conceptos que no son del todo homogéneos, es decir, entre los cuales hay muchas diferencias de propiedades, lo que requiere una reorganización de los conceptos. Para entender un poco más el concepto de entropía en nuestra propuesta, se explicará brevemente como es el uso de la entropía en la teoría de información y en los procesos de *clustering*¹⁴.

¹⁴ Clustering o Algoritmo de Agrupamiento, es un procedimiento de agrupación de un conjunto de objetos de tal manera que los miembros del mismo grupo sean más similares, en algún sentido u otro. Es la tarea principal de la minería de datos exploratoria y es una técnica común en el análisis de datos estadísticos [78].

En el área de la “Teoría de la Información”, la entropía se define como la medida de incertidumbre de una fuente de información, o la cantidad de información que contiene una variable aleatoria [77]. Shannon define la siguiente fórmula para determinar esta incertidumbre:

$$H(X) = - \sum_{i=1}^K P(x_i) \log_2 P(x_i) \quad (3.1)$$

Donde:

- X es el mensaje o variable a evaluar.
- k es el total de los posibles estados o escenarios.
- x_i estado o escenario i.
- $P(x_i)$ es la probabilidad del estado i.

Si un evento (variable) tiene un grado de determinación inicial k (k estados posibles), la probabilidad de que se dé uno de los estados es $p=1/k$. Para esos k estados se tiene que $\log_2(k)$ sería el máximo de iteraciones en una búsqueda binaria, entonces: $\log_2(k) = \log_2(1/1/k) = \log_2(1/p) = \log_2(1) - \log_2(p) = -\log_2(p)$. Si cada uno de los k estados tiene una probabilidad de p_i , entonces la entropía vendrá dada por la suma ponderada de la cantidad de información.

$$H = -p_1 \log_2(p_1) - p_2 \log_2(p_2) - \dots - p_k \log_2(p_k)$$

Entonces, la entropía de un mensaje X, denotado por $H(X)$, es el valor medio ponderado de la cantidad de información de los diversos estados del mensaje (Ec. 3.1). Por ejemplo, si se tiene un mensaje X que puede tener 3 estados posibles M_1 , M_2 y M_3 . Considerando que la probabilidad de cada estado es fija, se realiza el cálculo de la entropía para: $M_1=50%$, $M_2=25%$ y $M_3=25%$

$$H(X) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 1.5$$

Si las probabilidades son, $M_1: 100%$, $M_2:0%$ y $M_3:0%$, la Entropía entonces sería:

$$H(X) = -1 \log_2 1 - 0 \log_2 0 - 0 \log_2 0 = 0$$

En este caso no hay incertidumbre, existe certeza absoluta de que el valor de X siempre será M_1 .

En el caso de los procesos de clustering, la entropía también es usada como una de las métricas de calidad del proceso de agrupamiento, para determinar qué tan homogéneo es cada grupo, y por lo tanto, que tan bueno son [78]. La idea es que una vez finalizado el algoritmo de agrupación, se compare el clúster en el que fue asignado cada elemento con la etiqueta de clase que traía de antemano (información externa). En [78] se define ese proceso de clustering basado en entropía de la siguiente manera:

Sea C el conjunto de clases (o etiquetas) en el data set D:

$$C = (c_1, c_2, \dots, c_n)$$

El algoritmo de agrupamiento produce k clústeres, que particionan a D en k distintos subconjuntos:

$$D = (D_1, D_2, \dots, D_k)$$

Para cada clúster se define la entropía como:

$$E(D_i) = - \sum_{j=1}^n Pr_i(c_j) \log_2 Pr_i(c_j) \tag{3.2}$$

Donde (c_j) es la proporción de puntos de la clase c_j ubicados en el clúster i o D_i . La entropía total de todo el agrupamiento (que considera todos los k clústers) es:

$$Et(D) = \sum_{i=1}^k \frac{|D_i|}{|D|} \times E(D_i) \tag{3.3}$$

Un elemento tiene un grado de indeterminación inicial igual a k (k serían los posibles estados o *cluster*). Si todos tienen la misma probabilidad será: $p=1/k$.

Por ejemplo, para mostrar el uso de la entropía en clustering, se tienen 3 elementos representados en tres vectores de atributos: $v1: \{\text{rojo, alto}\}$, $v2: \{\text{rojo, bajo}\}$ y $v3: \{\text{azul, mediano}\}$. Si se desean crear 2 clusters con ellos, algunas posibles agrupaciones se muestran en la Tabla 3.1. Para cada agrupamiento se puede calcular la entropía total, pero primero se tiene que calcular la entropía (E) de cada clúster usando la Ec. 3.2, para luego determinar la entropía total (Et) de todos clústeres por agrupamiento. Se puede observar que el agrupamiento 1 tiene menor valor de entropía total, por lo que sería la manera correcta de agrupar los vectores utilizando 2 clústeres. En el proceso de clustering se busca establecer similitud entre elementos en base a sus atributos, y la entropía se ha usado como una medida para evaluar la diversidad en un clúster y entre los clústers, y así determinar el mejor agrupamiento. En base a esto, se ha considerado pertinente hacer uso de la entropía en el EEO, donde también se quieren agrupar y organizar conceptos. En este caso, la entropía nos permitirá medir el nivel de desorden, incertidumbre y diversidad, como un criterio importante en un proceso de emergencia ontológica.

TABLA 3.1. EJEMPLO DE ENTROPÍA

Nro. de clúster	Agrupamiento 1		Agrupamiento 2		Agrupamiento 3	
	Vectores	E	Vectores	E	Vectores	E
Clúster 1	v1: {rojo, alto} v2: {rojo, bajo}	1.0	v1: {rojo, alto} v3: {azul, mediano}	2.0	v1: {rojo, alto}	0
Clúster 2	v3: {azul, mediano}	0	v2: {rojo, bajo}	0	v2: {rojo, bajo} v3: {azul, mediano}	2.0
<i>Et</i>		0.66		1.33		1.33

3.5. Definiciones Generales del EEO

A continuación, se presentan las definiciones que se usaron en el desarrollo de la propuesta:

Definición 3.1. Una **Ontología** $O = \{C, P, R, I\}$ es un conjunto de conceptos **C** sobre un dominio específico, con sus propiedades **P**, entre los cuales existen relaciones **R**, y poseen instancias **I** asociadas. Las instancias, en este caso, son los datos que provienen del sistema.

Definición 3.2. Un **Evento Semántico (ES)** es aquella nueva fuente de datos que ingresa al sistema y que posee conceptos que pueden aportar información semántica. Los eventos semánticos son producto de las interacciones locales de las diferentes entidades, sistemas y servicios que participan en el sistema a nivel micro. Para esta propuesta, se consideran los siguientes tipos de fuentes de datos: Documentos en formato XML, XSD, XLS, y ontologías OWL.

Definición 3.3. Un **Meta-concepto (MC)** es un concepto genérico, que agrupa o representa un grupo de conceptos. En este caso, los meta-conceptos no tienen un nombre determinado. Se identifican a través de “Descriptor”, que se conforman por las propiedades de los conceptos que agrupan, y queda de parte del experto colocarles el nombre que considere adecuado.

Definición 3.4. Un **Descriptor (D)** es una etiqueta que identifica a un MC, y caracteriza a los conceptos pertenecientes al MC. Un MC se identifica por 1 o más descriptors. Un descriptor puede estar en la identificación de más de un MC.

Definición 3.5. Una **Meta-ontología (MO)** es una ontología que posee términos genéricos en forma de meta-conceptos, con sus relaciones y propiedades, que pueden ser usados para especificar o agrupar conceptos.

Definición 3.6. Una **Meta-ontología del Dominio (MOD)** es una meta-ontología que posee meta-conceptos específicamente relacionados al dominio del sistema. Por ejemplo, en la Fig. 3.2 se puede observar una *Meta-Ontología* del dominio de transporte, que posee 4 meta-conceptos con sus descriptors

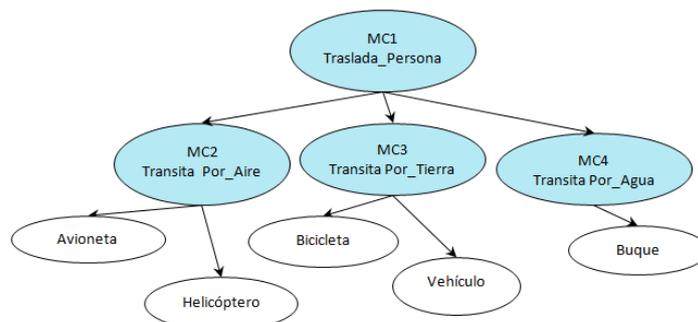


Figura 3.2. Ejemplo de Meta-Ontología de Dominio

Definición 3.7. La **Entropía Conceptual (EC)** determina el nivel de homogeneidad de un MC con respecto a los conceptos asociados a él.

Dado un conjunto D de descriptores:

$$D = (D_1, D_2, D_3, \dots, D_K)$$

La EC de un Meta-Concepto i, tomando como base las Ec. (3.1) y (3.2), será entonces:

$$EC(MC_i) = - \sum_{j=1}^K Pr(D_j) \log Pr(D_j) \quad (3.4)$$

Donde:

- K es el número de descriptores del MC.
- $Pr(D_j)$ es la proporción de conceptos con el descriptor D_j (que tenga entre sus propiedades a D_j), que pertenezcan al Meta-Concepto i, es decir:
 $Pr(D_j) = \text{Total de Conceptos con } D_j / \text{Total de Conceptos en } MC_i$

Definición 3.8. La **Entropía Conceptual General (ECG)** mide la entropía general o total de todos los meta-conceptos de una MO. En base a la Ec. (3.3), se tiene que:

$$ECG(MO) = \sum_{i=1}^N \frac{|MC_i|}{|MO|} \times EC(MC_i) \quad (3.5)$$

Donde:

- N es el número de MC de la MO.
- $|MO|$ es el número de conceptos en la MO.
- $|MC_i|$ es el número de conceptos del MC.

Definición 3.9. Un Meta-Concepto "MC1" es **Sub-Clase** de otro Meta-concepto "MC2" ($MC2 \leftarrow MC1$) si $D2 \subseteq D1$, donde $D1, D2$ son el conjunto de descriptores de $MC1$ y $MC2$, respectivamente.

Definición 3.10. La **Frecuencia (F)** de un concepto, es el número de veces que un concepto aparece en los diferentes ES que ingresan al sistema.

Definición 3.11. Las **Instancias (I)** de un concepto, es el número de objetos o ejemplares de un concepto que aparecen en los diferentes ES que ingresan al sistema.

Definición 3.12. El **Grado de Relevancia (GR)** es una medida que determina el grado de importancia de un concepto, y viene dado por la frecuencia de uso y la cantidad de instancias asociadas (Def. 3.10 y 3.11 respectivamente). La relevancia es uno de los criterios que determina si un concepto es obsoleto, candidato o emergente. Para calcular el GR, se obtienen primero los datos

normalizados (llevarlos a una escala de 0 a 1) de la *Frecuencia* del concepto (F) y de las *Instancias* asociadas al concepto (I), para luego obtener el promedio de estos 2 valores. Para normalizar los datos, se usará la fórmula:

$$X_N = \frac{X}{X_{max}} \tag{3.6}$$

Donde:

X: es el valor a normalizar.

X_{max}: valor máximo del rango de valores.

El GR vienen dado por:

$$GR = \frac{F_N + I_N}{2} \tag{3.7}$$

Donde:

F_N: Frecuencia de uso del concepto.

I_N: Instancias asociadas al concepto.

Este GR puede variar a medida que entran nuevos conceptos al sistema, ya que al llegar un nuevo concepto, los mínimos y máximos globales de I y F pueden cambiar.

Definición 3.13. La **Base de Conocimiento Semántico (BCS)** es donde se almacena la información semántica que surge en el sistema, que incluye tanto el registro de los eventos semánticos que suceden en el sistema, como los MC que se van conformando. Constituye el conocimiento colectivo, lo que representa la memoria social o colectiva del sistema. Está conformado por la Tabla de Anotaciones semánticas (TAS), el Directorio Ontológico (DO), y la Tabla de Meta-Conceptos (TMC) (Fig. 3.3).

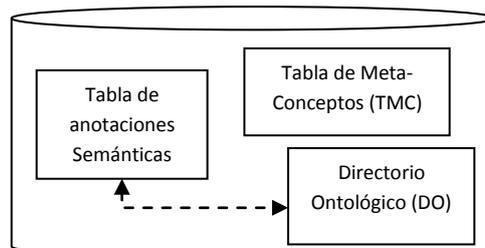


Figura 3.3. Arquitectura de la BCS

Definición 3.14. La **Tabla de Anotaciones Semánticas (TAS)** es una tabla donde se registran los ES que surgen en el sistema (Tabla 3.2). Los modelos conceptuales de esos ES que participan en el sistema, deben registrarse para que puedan ser analizados, y ser usados así, en la emergencia ontológica. Al registrar un ES, cada concepto indica sus propiedades y se debe asociar con algún meta-concepto. Para indicar a que MC pertenece, se debe determinar la similitud entre el concepto y el meta-concepto en base a las propiedades de ambos. Se registran también las Instancias "I" (cantidad de datos asociados), la Frecuencia "F" (las veces que ha aparecido el concepto) y se calcula el GR (Def. 3.12).

TABLA 3.2. TABLA DE ANOTACIONES SEMÁNTICAS

Concepto	MC	Propiedades	TIPO	GR	I	F
Vehículo	MC3	TrasladaPersonas, Transita por Tierra	CE	1	50	1
Avioneta	MC2	TrasladaPersonas, Transita por Aire	CC	0.75	25	1
Buque	MC4	TrasladaPersonas, Transita por Agua	CO	0.20	5	1

Definición 3.15. El **Directorio Ontológico (DO)** es un registro de las ontologías que conforman el Marco Ontológico del sistema (Tabla 3.3). Se debe indicar una descripción de la ontología y su ubicación.

TABLA 3.3. DIRECTORIO ONTOLÓGICO

Ontología	Ubicación
"Transporte"	"<http://www.saoye/transporte.owl/>"
"Onto_Transport"	"<http://www.saoye/Onto_transport.owl/>"

Definición 3.16. La **Tabla de Meta-Conceptos (TMC)** es un registro de los meta-conceptos que se van generando en el proceso de emergencia ontológica, donde se indican los descriptores, a que clase pertenece, y valor de la entropía (ver ejemplo en la Tabla 3.4).

TABLA 3.4. TABLA DE META-CONCEPTOS

MC	Descriptores	Super-clase	Entropía
MC1	Traslada_personas	thing	0.2
MC3	Transita Por_Tierra, Tiene_Ruedas	MC1	0

Definición 3.17. La **Similitud Conceptual (Sim_C)** es la medida que determina la similaridad entre 2 conceptos C_1 y C_2 (o entre 2 meta-conceptos, o entre 1 concepto y 1 meta-concepto). Se obtiene usando la *Similitud Léxica* (Sim_L), que se calcula usando la distancia de Levenshtein¹⁵ entre los nombres de los conceptos, y la *Similitud de Propiedades* (Sim_P). La Sim_P permite obtener una similitud más exacta, y se calcula de la siguiente manera:

$$Sim_P(C_1, C_2) = \frac{|P \cap P'|}{|P \cup P'|} \quad (3.8)$$

Donde:

- P y P' son los conjuntos de propiedades de los conceptos C_1 y C_2 respectivamente.
- $|P \cap P'|$ es la intersección entre conjuntos.
- $|P \cup P'|$ es la unión entre conjuntos.

Para obtener la similitud entre 2 propiedades se usará la similitud léxica.

Si esta similitud es mayor a 0.8 (o a un valor pre-definido) se consideran similares o equivalentes.

¹⁵ La distancia de Levenshtein, distancia de edición o distancia entre palabras es el número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra, se usa ampliamente en teoría de la información y ciencias de la computación [114].

Para calcular Sim_C se promedian las dos similitudes Sim_L y Sim_P , obteniéndose un valor entre 0 y 1. Si son exactamente idénticos, en cuanto a su identificación y propiedades, la similitud será 1.

$$Sim_C(C_1, C_2) = \frac{Sim_L(C_1, C_2) + Sim_P(C_1, C_2)}{2} \quad (3.9)$$

Definición 3.18. El **Umbral Mínimo de Relevancia (UMR)** es el umbral mínimo permitido del GR para que un concepto sea “candidato”. Si GR es menor a este valor, entonces el concepto pasa a ser “obsoleto”.

Definición 3.19. El **Umbral Requerido de Relevancia (URR)** es el valor requerido del GR para que un concepto candidato pase a ser emergente. Si GR es menor a este valor, entonces el concepto pasa a ser candidato.

Definición 3.20. El **Umbral Máximo de Entropía (UME)** es el valor máximo de entropía que puede soportar una MO. Si el valor de entropía supera este valor, se debe reorganizar o reagrupar los MC en la MO.

Definición 3.21. La **Tasa de Estabilización de Relevancia (TER)** es la tasa de disminución del GR al que se someterán todos los conceptos de la TAS, en un mecanismo de estabilización de la BCS. Se usa en el proceso de “Registro Semántico” (Macro-Algoritmo 3.3) en el proceso de retroalimentación negativa para impedir el crecimiento ilimitado de la BCS y reducir progresivamente el GR de los conceptos que van perdiendo relevancia en el sistema. Se puede inicializar en un valor muy pequeño, para que la reducción del GR no sea muy significativa.

Definición 3.22. Un **Concepto Candidato (CC)** es aquel concepto que se registra en la BCS, que surge a partir de un evento semántico del sistema, y es candidato a ser parte de la Ontología Emergente del SAOyE.

$$CC \Rightarrow C \in BCS \mid GR \geq UMR \text{ y } GR < URR$$

Definición 3.23. Un **Concepto Emergente (CE)** es aquel concepto, que por su nivel de GR (mayor al URR) en el sistema, pasa a ser parte de la Ontología Emergente del sistema.

$$CE \Rightarrow C \in BCS \mid GR \geq URR$$

Definición 3.24. Un **Concepto Obsoleto (CO)** es aquel concepto, que por su bajo GR (menor al UMR), se convierte en obsoleto. Al ser un concepto obsoleto, no es considerado en cada una de las iteraciones del proceso de emergencia ontológica.

$$CO \Rightarrow C \in BCS \mid GR < UMR$$

Definición 3.25. Una **Ontología Emergente (OE)** es una ontología que se conforma de CEs, que se organizan en base a MC. Una ontología emergente se va formando dentro de un proceso de emergencia ontológica, donde los CE que

la conforman y la forma en que se agrupan en los MC, dependerán de los umbrales de relevancia y de entropía que se definan durante el proceso.

3.6. Arquitectura General del EEO

El EEO propuesto se inspira en los modelos para sistemas emergentes planteados en [9] [63], donde se indican los componentes y mecanismos que deben estar presentes para que exista un proceso emergente. En la Fig. 3.4, se puede observar el esquema propuesto, cuyos componentes principales son: “Componente de Registro Semántico”, “Componente de Análisis de Conceptos”, “Componente de Integración” y “Componente de Calidad”. La funcionalidad de cada componente se explica con detalle más adelante (Sección 3.7), y en el Macro-Algoritmo 3.1 se muestra cómo interactúan dichos componentes. Estos componentes se organizan en dos modalidades o procesos, que se describen a continuación:

1. *Caracterización Semántica (Proceso Bottom-Up)*: es donde se genera la información semántica desde las fuentes de datos del sistema. Este proceso va de lo individual a lo colectivo ya que cada uno de los modelos conceptuales individuales (tablas, documentos, ontologías, etc.) dan a conocer su información semántica, para que pueda ser procesada y compartida. Entonces, el conocimiento implícito comienza a emerger, a través de procesos donde el sistema realiza el registro semántico de sus eventos y de sus modelos conceptuales y ontologías, para luego, mediante diversos mecanismos de agregación, generar un conocimiento explícito que pueda ser almacenado y comunicado. Utiliza dos mecanismos:
 - *Mecanismo de Socialización* (Componente de Generación de Meta-Ontologías y Componente de Registro Semántico): se registra y comparte la información semántica que ofrecen las diferentes fuentes de datos en el sistema, para generar un conocimiento explícito que se pueda conocer, y que pueda ser procesado y compartido en el sistema. A través de este mecanismo se busca detectar las nuevas propiedades y comportamientos que surgen en el sistema a nivel conceptual.
 - *Mecanismo de Agregación* (Componente de Análisis): se crea un conocimiento explícito clasificado y refinado, a partir de tareas de análisis, procesamiento y depuración del conocimiento que fue compartido, para que pueda ser útil al colectivo.
2. *Estructuración Ontológica (Proceso Top-Down)*: es donde se conforman y se le da forma a las OEs. Este proceso va de lo colectivo a lo individual, y es donde el conocimiento colectivo generado es internalizado, a través de la organización de los CE y MC en la MOD. Está conformado por el Componente de Integración y el Componente de Calidad, que se complementan y representan el “Mecanismo de Apropiación” del proceso de emergencia ontológica:

- *Mecanismo de apropiación* (Componente de Integración y Calidad): el conocimiento explícito es asimilado por el sistema, para ser usado a través de la OE resultante.

El proceso de emergencia ontológica mostrado en la Fig. 3.4 se inicia a partir de ESs que se registran en la BCS, que son conceptos candidatos a incluirse en la ontología emergente. A partir de este registro, se procede al análisis, y en conjunto con la MOD del sistema, se procede a la estructuración de la OE, para reflejar los cambios en el comportamiento en el sistema y en sus fuentes de datos.

En un proceso emergente, la dinámica de aprendizaje colectivo se basa en mecanismos de retroalimentación positiva y negativa. La retroalimentación positiva promueve la creación de estructuras y cambios en el sistema, y la retroalimentación negativa hace el equilibrio al ayudar a estabilizar el comportamiento colectivo [9]. En este caso, la retroalimentación positiva se refleja en el ingreso de nuevos CC al sistema y el incremento de su GR (Def. 3.15), lo que permite reforzar la candidatura de un concepto y estimula la aparición de un CE. La retroalimentación negativa busca crear un mecanismo de estabilización que impida el crecimiento ilimitado de la BCS, reduciendo el GR de aquellos conceptos que no son relevantes en el sistema, para contribuir así a su desaparición.

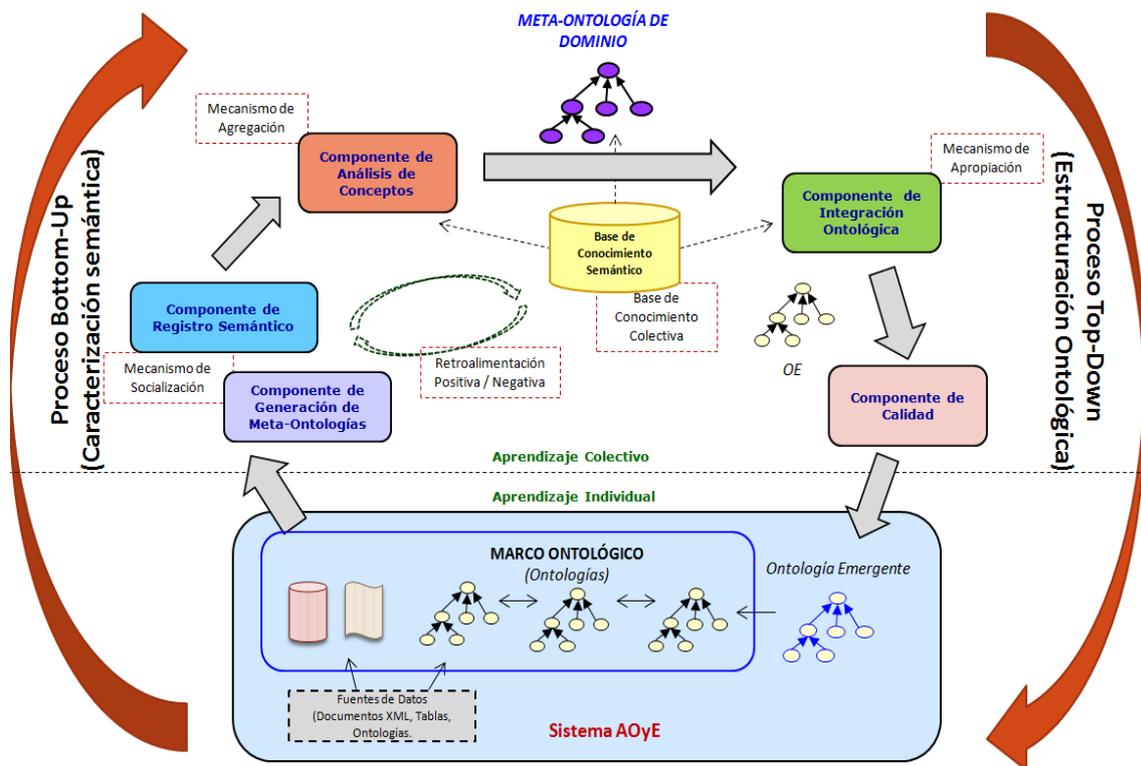


Figura 3.4. Esquema de Emergencia Ontológica

De manera general, las etapas del esquema de emergencia ontológica se pueden observar en el Macro-Algoritmo 3.1. El proceso se inicia con los ESs que

surgen a partir de las fuentes de datos que ingresan al sistema, que por lo general son datos semi-estructurados.

MACRO-ALGORITMO 3.1: ESQUEMA DE EMERGENCIA ONTOLÓGICA

Entradas: Evento Semántico (ES)

Procedimiento:

1. Definir Umbrales (UMR, URR, UME, TER)
2. Generar Meta-Ontologías (Macro-Algoritmo 3.2)
3. Mientras exista un nuevo ES
 - 3.1. Procesar Registro Semántico (Macro-Algoritmo 3.3)
 - 3.2. Procesar Análisis de Conceptos (Macro-Algoritmo 3.5)
 - 3.3. Procesar Integración Ontológica (Macro-Algoritmo 3.6)
 - 3.4. Procesar Verificación de calidad de la OE (Macro-Algoritmo 3.10)

Salidas: OE

En primer lugar, se deben definir los umbrales de relevancia y entropía (Paso 1), ya que estos serán claves en la calidad de la OE resultante (Definiciones 3.16, 3.17 y 3.18). Para definir los valores de umbrales, se puede realizar un proceso de calibración, donde se establecen unos valores iniciales, y en base a los resultados, se pueden ajustar si se requiere obtener ontologías de diferente calidad. Los meta-conceptos de la MOD son elementos indispensables en la estructuración de la OE. Estos meta-conceptos se van formando en el proceso, pero también, es posible invocar la generación de una MOD inicial (Paso 2) a partir de ontologías del dominio registradas (por lo menos 2), previamente alineadas. También pueden existir meta-conceptos previamente definidos por expertos.

Por cada ES (Paso 3), se inicia una iteración donde se ejecutan todos los pasos del algoritmo hasta generar una OE. Esta OE tendrá unas métricas de calidad que permitirán al experto evaluarla, y de ser necesario, ajustar los parámetros iniciales (Umbrales de Relevancia y Entropía).

El primer proceso lo realiza el componente de “Registro Semántico” (Paso 3.1), para extraer la información semántica de los ESs que ingresan al sistema. Luego, el componente de “Análisis de Conceptos” (Paso 3.2) procesa esta información para determinar los CE. El componente de “Integración Ontológica” (Paso 3.3), a través de la MOD, provee la estructura base para la integración de estos nuevos CE. Finalmente, el componente de “Calidad” (Paso 3.4) verifica y evalúa la OE generada, con criterios de calidad específicos. Cada uno de esos componentes actúa de forma independiente, siguiendo una serie de reglas, pero de manera global, todos contribuyen al proceso de emergencia ontológico.

A continuación, se describen los diferentes componentes del Esquema de Emergencia Ontológica.

3.7. Componentes del EEO

Los componentes que participan en el EEO, se organizan en los 2 procesos generales de “Caracterización Semántica” y “Estructuración Ontológica”, ya descritos en la sección anterior.

3.7.1. Proceso de Caracterización Semántica

Dentro del proceso de Caracterización Semántica, que es donde se obtiene información a partir de las fuentes de datos, se definen los siguientes componentes: Componente de Generación de Meta-Ontologías, Componentes de Registro Semántico y Componente de Análisis de Conceptos.

A. Componente de Generación de Meta-Ontologías

En este componente se realiza la generación automática de la Meta-Ontología de Dominio (MOD) en base a la “Teoría de Categorías”. Las MODs, como se indica en la Definición 3.6, contienen información genérica sobre el dominio, y pueden ser instanciadas para generar nuevas ontologías. Una MOD puede ser definida por expertos del conocimiento del dominio, o a través de procesos automáticos de combinación de múltiples ontologías de un mismo dominio, como el que se muestra en el Macro-Algoritmo 3.2. Antes de explicar el proceso, se presenta una breve reseña de la Teoría de Categorías.

Teoría de Categorías

De acuerdo a [79], una categoría consiste en un conjunto de objetos que están relacionados de alguna manera. Una categoría es una colección de datos que satisfacen alguna(s) propiedad(es) en particular. La “Teoría de Categorías” es la teoría matemática de las estructuras, y su mayor importancia radica en su capacidad de expresar relaciones entre estructuras [80]. Basados en esta teoría, se presentan un conjunto de definiciones que se emplearán luego en la generación de meta-conceptos y una MOD [79] [81].

Una categoría **C**, según la teoría de categorías, es una estructura en la que participan los siguientes elementos:

- Un conjunto de objetos **Obj(C)**, que se denotan como A, B, C, ...
- Un conjunto de morfismos **Mor(C)**, que se denotan como f,g,h, ...
- Una relación que asocia a cada morfismo un par de objetos, que se denota de la siguiente manera:

$$f: A \rightarrow B$$

Donde A y B representan el dominio y el co-dominio del morfismo f, respectivamente. A su vez, un morfismo es una función, propiedad o característica, que asocia un objeto con otro. Por ejemplo, 2 objetos pueden ser: Auto(A) y Persona(P), y el morfismo Trasladar(T) asocia estos 2 objetos:

$$T: A \rightarrow P$$

Una categoría **D** es **sub-categoría** de una categoría **C** si:

- $\text{Obj}(D) \subseteq \text{Obj}(C)$
- $\text{Mor}(D) \subseteq \text{Mor}(C)$

Considerando estas fundamentaciones propias de la teoría de categorías, se define los conceptos usados en el proceso de generación de meta-ontologías.

Definición 3.26. Una **Categoría** es una colección de conceptos que presentan una o más propiedades en común. En base a la teoría de categorías, se tiene que en un contexto X, una categoría *Cat1* está definida por una estructura (C, P), donde:

- C es un conjunto de Conceptos (Objetos en Teoría de Categorías).
- P es un conjunto de Propiedades (Morfismos en Teoría de Categorías), tal que si:

$$f \in P, \text{ entonces } f : A \rightarrow B, \text{ para } A, B \in C$$

donde A es el dominio y B es el rango de la propiedad.

Definición 3.27. El **Alcance** de una categoría está representado por el conjunto de conceptos que conforman su dominio, y son usados por sus propiedades (morfismo).

Definición 3.28. La **Precisión** de una categoría está representada por el conjunto de propiedades que posee esa categoría.

Definición 3.29. Una categoría *Cat1* es **Sub-Categoría** (o “sub-clase”) de *Cat2* ($Cat1 \subseteq Cat2$) si: $C1 \subseteq C2$ y $P1 \subseteq P2$, siendo C1, C2 el conjunto de conceptos, y P1, P2 el conjunto de propiedades de *Cat1* y *Cat2*, respectivamente.

Definición 3.30. Una **Categoría Ontológica (Cat-O)** es aquella categoría conformada por todos los conceptos y propiedades que participan en el contexto.

Definición 3.31. Una **Sub-categoría Genérica (Sub-Cat-G)** es aquella con alcance mayor a un umbral (el umbral será un mínimo definido de acuerdo al universo de conceptos. Si no se define, entonces es igual a 1).

Definición 3.32. Una **Sub-categoría Específica (Sub-Cat-E)** es aquella con alcance igual a un umbral (el umbral será un mínimo definido de acuerdo al universo de conceptos. Si no se define, entonces es igual a 1).

Proceso de Generación Automática de Meta-Ontologías

El proceso automático de generación de meta-ontologías basado en la “Teoría de Categorías” desarrollado en [82], se realiza a partir de la combinación de múltiples ontologías del mismo dominio. Se puede activar, cuando en el Directorio Ontológico (Def. 3.15) se han registrado más de 2 ontologías del mismo dominio.

Para la generación de una meta-ontología de dominio, lo que se requiere es determinar cuáles son los meta-conceptos. Estos meta-conceptos representan las clases genéricas, que poseen propiedades heredables.

Dada un conjunto de alineaciones entre un ontología O y N ontologías pertenecientes al mismo dominio, el objetivo es la generación de una meta-ontología del dominio, aplicando el algoritmo ACO. Los algoritmos ACO se inspiran directamente en el comportamiento de las colonias reales de hormigas, para solucionar problemas de optimización combinatoria. Se basan en una colonia de hormigas artificiales, esto es, agentes computacionales simples que trabajan de manera cooperativa, y se comunican mediante rastros de feromona artificiales. Los algoritmos ACO son constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema, recorriendo un grafo que contiene soluciones parciales. Cada arista del grafo, que representan los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga: la información heurística, que mide la preferencia heurística de moverse desde un nodo hasta otro, y la información de los rastros de feromona artificiales, que mide la “deseabilidad aprendida” de ese movimiento, que imita a la feromona real que depositan las hormigas naturales. Esta información es la que permite el aprendizaje colectivo, y se va alimentado durante la ejecución del algoritmo con las soluciones encontradas por las hormigas [71] [83].

Para la generación de la meta-ontología se toma como insumo la información obtenida en el recorrido realizado por las hormigas en busca el mayor grado de enriquecimiento de la ontología O, en el proceso de combinación múltiple realizado en [84]. La medida de similitud es léxica y estructural (parecido entre padres, hermanos e hijos). En este caso, la medida de similitud definida a nivel de los conceptos se formula también

en base sus “propiedades”, lo que representa un nuevo criterio que contribuye en la determinación de la alineación más adecuada para cada uno de los conceptos de la ontología.

En específico, en el proceso de combinación múltiple, las hormigas seleccionan una de las alineaciones como parte de la solución, a partir de la información heurística que viene dada por una medida de similitud entre los conceptos, y a partir de los rastros de la feromonas, lo que permite explorar todas las posibles soluciones. Las hormigas realizan el recorrido en un grafo de soluciones, que está formado por todas las posibles combinaciones de las alineaciones ya existentes para cada uno de los conceptos alineados, para así ir seleccionando los vértices que formarán parte de una posible solución de combinación (cada nodo representa una alineación de un concepto de la ontología objetivo con cada una de las N ontologías).

Es importante resaltar que las alineaciones ya deben estar definidas en base a los elementos individuales (conceptos) de la ontología objetivo, para poder así construir el grafo. Como ya se dijo, la comparación para medir la similitud entre un par de nodos alineados de dos ontologías se realiza originalmente en base a su similitud léxica dada por la alineación, y su similitud estructural dada por el análisis de sus posiciones, es decir, si dos nodos de dos ontologías son similares sus vecinos deben ser similares de alguna forma.

La solución de un problema particular basado en el algoritmo ACO comienza con el diseño del espacio de búsqueda del problema, en el cual las hormigas realizan la búsqueda de las posibles soluciones. En este caso, el problema viene dado por la necesidad de obtener una meta-ontología a partir de la combinación de una ontología que llamaremos “Ontología Origen” O , con otras N ontologías con las cuales ya está alineada. El proceso de comparación de la ontología O con las ontologías $O'1, O'2, \dots, O'N$ entre las cuales ya existen definidas una serie de alineaciones $A1, A2, \dots, AN$, consiste en seleccionar por cada concepto C perteneciente a O , la mejor alineación posible de las N alineaciones, en base a la mayor similitud y el mayor enriquecimiento que puede obtener la ontología O . Entonces, el espacio de búsqueda es definido por la combinación de las diferentes alineaciones para cada concepto de la ontología O con las N ontologías, como el que se muestra en la Figura 3.5.

Si se tienen entonces M conceptos en la ontología origen y N alineaciones, se tiene un máximo de N^M posibles soluciones para este problema.

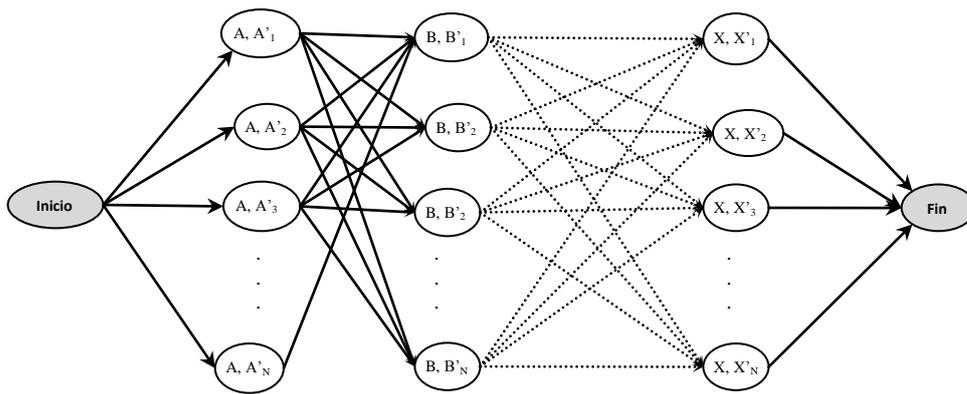


Figura 3.5. Espacio de soluciones para el recorrido de las hormigas

Para el caso específico de la aplicación de ACO, se debe definir un punto de partida donde inician el recorrido todas las hormigas, el cual, en este caso será el concepto raíz de la ontología origen que se desea combinar con las otras N ontologías. La hormiga inicia su recorrido en el vértice etiquetado con "Inicio", y realiza sus movimientos en el espacio de búsqueda de izquierda a derecha (basadas en la feromona y la información heurística disponible). Sólo un nodo puede ser seleccionado por cada columna del espacio de soluciones. En la primera columna están las alineaciones del primer concepto de la ontología origen O (en el ejemplo de la Figura 3.6, concepto A) con los conceptos alineados en las N ontologías (aquí se considera que cada concepto está alineado sólo una vez con algún concepto en una ontología). Después que se ha seleccionado una alineación para un concepto, ya no se puede seleccionar otra para ese concepto, y se continúa con otro concepto de la ontología origen.

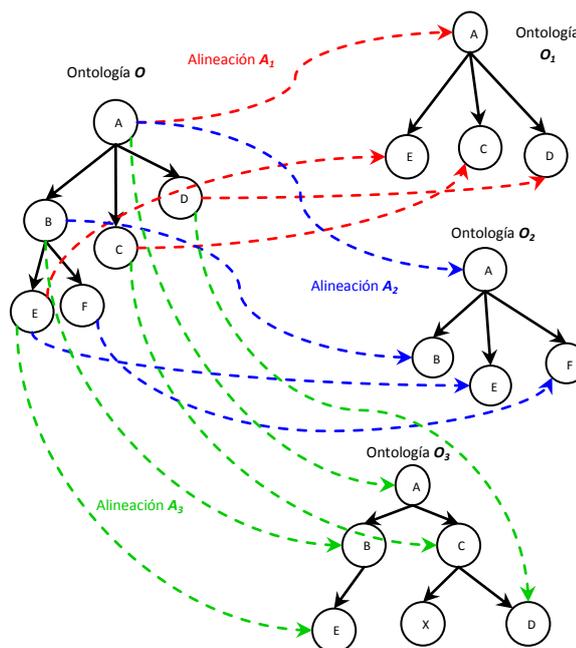


Figura 3.6 Alineaciones entre la ontología O y las ontologías O₁, O₂, O₃

Para visualizar el proceso, tenemos el ejemplo mostrado en la figura 4, donde se representa el espacio de búsqueda de las alineaciones de una ontología origen y otras 3 ontologías mostradas en la figura 3. En este ejemplo, las alineaciones A1, A2 y A3 ya definidas se pueden describir como:

$$A1 = \{(A, A'1), (C, C'1), (D, D'1), (E, E'1)\}$$

$$A2 = \{(A, A'2), (B, B'2), (E, E'2), (F, F'2)\}$$

$$A3 = \{(A, A'3), (B, B'3), (C, C'3), (D, D'3), (E, E'3)\}$$

Entre cada par de conceptos está definida, por la alineación ya existente, una medida de similitud que se representa como:

$$Sim(C, C'i) \quad \forall i= 1, 3$$

Esta medida de similitud debe estar entre 0 y 1, lo que refleja el grado de similitud entre los conceptos alineados, que pueda estar dada, por ejemplo, por una similitud léxica¹⁶. Todos estos valores de similitud se considerarán como una entrada, los cuales se almacenarán en una tabla que se llamara "Tabla de Similitud entre Conceptos". Como ya se ha dicho, es importante aclarar que para usar este método, ya las alineaciones deben estar definidas entre las ontologías. En particular, deben existir las alineaciones y la medida de similitud entre pares de conceptos. Las hormigas deben realizar el recorrido visitando cada concepto de la ontología origen (columna del espacio de soluciones), seleccionando la que consideran la mejor alineación (por ejemplo, un nodo (A, A'i)).

La hormiga inicia seleccionando el concepto que considera como mejor alineación del concepto raíz, luego, con esa selección como parte de la solución, va a seleccionar la mejor alineación para el siguiente concepto, realizando en este caso un análisis en amplitud y un recorrido en profundidad del grafo de búsqueda (lo que implica evaluar a todos los nodos de un nivel antes de decidir continuar con el siguiente), hasta considerar todos los conceptos de la ontología origen.

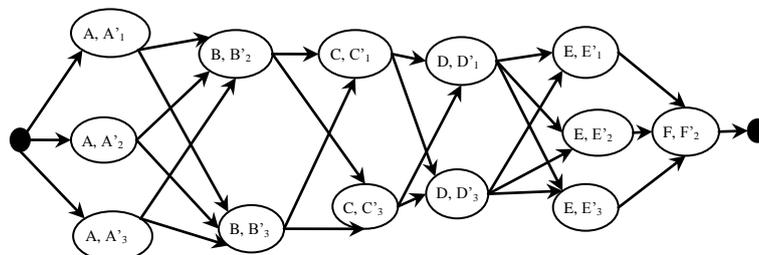


Figura 3.7. Diferentes combinaciones de alineaciones para cada concepto

¹⁶ Similitud de dos expresiones en base a su escritura.

Durante el recorrido que realizan las hormigas por el espacio de búsqueda se realiza un proceso de aprendizaje colectivo, donde por cada alineación visitada que involucra un par de conceptos relacionados, se obtiene información importante para el establecimiento de estas clases genéricas en base a propiedades comunes entre conceptos. Igualmente, al determinar la calidad de la solución para darle valor a la feromona, se considerarán las propiedades comunes de los nuevos conceptos que contribuyen en el GE. Todo el aprendizaje que va obteniendo la hormiga en sus diferentes recorridos, sirve de insumo para conformar una ontología generalizada del dominio, es decir, una meta-ontología.

Ese aprendizaje colectivo que se quiere obtener durante el recorrido de las hormigas por el espacio de soluciones, es el conjunto de propiedades comunes entre los conceptos alineados que se visitan, que pueden formar parte de las propiedades de una clase superior o general, lo que da insumos para una posible definición de los meta-conceptos.

Para registrar y utilizar ese aprendizaje colectivo, en la misma medida en que la hormiga recorre el espacio de soluciones y actualiza con feromona, se va a ir generando una “Matriz de Aprendizaje Colectivo de Propiedades” (MACP), donde las hormigas colocan información relacionada a las propiedades comunes entre los conceptos involucrados en las alineaciones visitadas, y de los nuevos conceptos obtenidos en el enriquecimiento de la ontología origen. En la Tabla 3.5 se muestra la estructura que tendrá la MACP, la cual es una matriz donde se registran las Propiedades Comunes (PC).

TABLA 3.5. MATRIZ DE APRENDIZAJE COLECTIVO DE PROPIEDADES (MACP).

		Conceptos de la Ontología Origen			
		C ₁	C ₂	...	C _N
Conceptos Alineados	C' ₁	PC ₁ ...PC _N
	C' ₂	...			
	...				
	C' _N
Nuevos Conceptos Obtenidos	CN ₁
	CN ₂
	...				
	CN _N

En la Matriz MACP, en el identificador de cada columna están los conceptos de la ontología origen en la combinación múltiple, y en los identificadores de las filas están, en primer lugar, los conceptos alineados, de las ontologías involucradas en la combinación múltiple, y luego, los conceptos nuevos obtenidos por la ontología origen durante el enriquecimiento. Estos conceptos nuevos se consideran importantes para la generación de la meta-ontología, ya que son conceptos vecinos (hijos, hermanos o padres) con los que existe una similitud.

En el Macro-Algoritmo 3.2 se muestra un resumen de todo el proceso. La entrada del proceso está conformada por el conjunto de conceptos pertenecientes a las ontologías que se combinan y sus propiedades comunes. Primeramente (Paso 1), se genera una Tabla de Conceptos con Propiedades Comunes (Tabla 3.5).

TABLA 3.6. TABLA DE CONCEPTOS CON PROPIEDADES COMUNES (TCPC)

Propiedad Común	Conceptos
PC ₁	C ₁ , C ₂ , C _{3...} , C _N
PC ₂	C ₁ , C ₂ , C _{3...} , C _N
...	...
PC _N	C ₁ , C ₂ , C _{3...} , C _N

Luego, se establecen las categorías ontológicas (Cat-O) del dominio, que contienen todos los conceptos y morfismos involucrados en el contexto (Paso 2). A continuación, se determinan las sub-categorías generales (Sub-Cat-G) y específicas (Sub-Cat-E), agrupando categorías en base a su alcance (conceptos que representan) (Paso 3). Las categorías generales serán entonces los meta-conceptos (Paso 4), que se organizarán de forma jerárquica, definiendo relaciones de “sub-clase” entre categorías y sub-categorías, para así estructurar la meta-ontología de dominio (MOD, Paso 5).

MACRO-ALGORITMO 3.2 : GENERACIÓN DE META-ONTOLOGÍA DEL DOMINIO

<u>Entradas:</u> Conceptos y Propiedades Comunes.	
<u>Procedimiento:</u>	
1.	Registrar propiedades comunes entre conceptos en la TCPC (Tabla 3.5).
2.	Establecer las Categorías Ontológicas (Definición 30).
3.	Definir las Sub-Categorías Generales y Específicas (Definiciones 3.31 y 3.32).
4.	Establecer las Sub-Categorías Generales como candidatas a Meta-Conceptos.
5.	Definir las relaciones jerárquicas entre Meta-Conceptos para conformar las Meta-Ontologías.
<u>Salidas:</u> Meta-Ontología del Dominio	

B. Componente de Registro Semántico

Objetivo: generar anotaciones semánticas a través del registro en la BCS de los nuevos conceptos que aparecen en el sistema a partir de los ESs, y que pueden ser candidatos a incluirse en la OE. A partir de esos registros se busca detectar el surgimiento de CEs, para así reflejar los cambios en el comportamiento en el sistema y en sus fuentes de datos.

Descripción General: representa el “mecanismo de socialización”, a través de la cual se comparte la información semántica que ofrecen las diferentes fuentes de datos que intervienen en el sistema. Las anotaciones se realizan a nivel de meta-datos (propiedades de los datos), ya que lo que se necesita es información sobre los modelos conceptuales que se manejan, para así tener realmente información semántica de las fuentes de datos. El orden en que se registren los ESs no tendrá impacto en la ontología resultante. En el Macro-Algoritmo 3.3 se detalla el proceso de registro semántico. En primer lugar, se deben buscar los CC (Paso 1), que según el

tipo de fuente se obtienen aplicando una regla diferente, como se explicará más adelante en el Macro-Algoritmo 3.4. Por cada CC se realiza el registro en la TAS (Def. 3.14) (Paso 2). Para ello, primero se realiza una búsqueda del CC para verificar si ya existe (Paso 2.1). Al realizar la búsqueda se calcula primero la similitud léxica y luego la similitud conceptual (Sim_C)(Def. 3.17). Se considera que se refiere al mismo concepto si supera el 80% de similitud. Si ya existe el concepto, se actualiza el número de instancias del concepto (sumando la cantidad de datos asociados), se le agregan las propiedades, y se incrementa la frecuencia. Si no existe, entonces se registra el nuevo CC indicando las propiedades e instancias asociadas, colocando la frecuencia en 1. Luego se calcula y actualiza el valor del GR (Ec. 3.7) del concepto (Paso 2.2). Para ello, se deben primero normalizar los valores de I y de F, llevándolos a una escala entre 0 y 1 (Ec. 3.6). Por último, se disminuye el GR de todos los conceptos (Paso 3), a razón de la TER (Def. 3.21) que se haya establecido. Esto corresponde al proceso de retroalimentación negativa que se explicó anteriormente, que busca crear un mecanismo de estabilización que impida el crecimiento ilimitado de la BCS y para reducir el GR de aquellos conceptos que no son relevantes en el sistema, así poco a poco pueden ir desapareciendo de la lista de conceptos, para formar parte de un registro de históricos. De esta manera, aquellos conceptos que surgen de forma temporal en el sistema, pero que no son relevantes para ser incluidos en la OE, no serán tomados en cuenta; igualmente, aquellos conceptos que se han dejado de usar, por uno u otra razón, deben ir poco a poco desapareciendo.

MACRO-ALGORITMO 3.3: REGISTRO SEMÁNTICO

Entradas: Evento Semántico (ES)

Procedimiento:

1. Realizar el proceso de "Búsqueda de CC" (macro-algoritmo 3.4)
2. Por cada CC:
 - 2.1. Buscar el CC en la TAS
 - 2.1.1. Si existe
 - 2.1.1.1. Actualizar las instancias, las propiedades e incrementar la frecuencia del concepto.
 - 2.1.2. Si no existe
 - 2.1.2.1. Registro en la TAS el CC, indicando sus propiedades, el número de instancias asociadas, y colocar la frecuencia en 1.
 - 2.2. Actualizar el GR del CC
 - 2.2.1. Normalizar el valor de Instancias
 - 2.2.2. Normalizar el valor de Frecuencias
 - 2.2.3. Calcular el GR del CC
3. Decrementar el GR de todos los conceptos de la TAS, de acuerdo a la TER.

Salidas: Base de Conocimientos Semántico (BCS) actualizada

El proceso de Búsqueda de CCs se detalla en el Macro-Algoritmo 3.4. Este algoritmo recibe como entrada un ES (Fuente de Datos) que debe procesar para retornar los CCs que contenga. La fuente de datos puede pasar por un pre-procesamiento, donde se seleccionan las propiedades más significativas y se descartan aquellas que no aporten casi información.

Igualmente, en el pre-procesamiento se pueden adecuar los nombres de las propiedades, con identificadores estándares. Como entrada, también se puede indicar si existe una propiedad que contenga categorías o tipos de datos, esto sirve para detectar más CCs en una misma fuente de datos.

Por cada ES, se debe determinar el tipo de “ES” (Paso 1), que viene dado por el formato del documento, para seleccionar el tipo de acciones a realizar para procesar la fuente y detectar posibles CCs, así como sus propiedades. Si es un documento XML (Paso 2), se recorre el documento, las etiquetas principales serán los posibles CCs, y las etiquetas anidadas propiedades del concepto. En este caso, en los documentos XML, si se repiten etiquetas principales, no se considerarán como nuevos conceptos, sino como instancias. De la propiedad identificada como categoría se extraerá su contenido para identificar posibles CC, a los que se le asignarán las mismas propiedades. Esta categoría permitirá extraer de un ES varios conceptos. Si es un documento XSL (Paso 3), o tabla de Excel, se seleccionará el nombre del documento como CC, y las columnas serán las propiedades. Se aplicará el mismo proceso para la propiedad identificada como categoría, y se cuentan los datos contenidos en la tabla, que servirán para actualizar las instancias del concepto. Si es un documento XSD (Paso 4), que es una definición de esquema para definir la estructura de documentos XML, se tomarán los “xsd:element” principales como CCs, y los anidados y los “xsd:attribute” como propiedades. En este tipo de documento no se tendrán datos asociados. Igualmente pasa si es una ontología (Paso 5), en este caso se debe registrar en el Directorio Ontológico, y se seleccionan las “clases” como conceptos y los “object property” y su rango como propiedades. El proceso retorna una lista con los CCs que se obtuvieron de la fuente de datos.

MACRO-ALGORITMO 3.4: BÚSQUEDA DE CC

Entradas: Evento Semántico (ES), campo categoría

Procedimiento:

1. Determinar el Tipo de ES
 2. Si es un XML
 - 2.1. Recorrer el documento
 - 2.2. Seleccionar las etiquetas principales como identificador del CC
 - 2.3. Seleccionar las etiquetas anidadas como propiedades del CC
 - 2.4. Si se indicó un campo categoría
 - 2.4.1. Seleccionar el contenido de la propiedad que coincide con el campo categoría
 - 2.5. Se contabiliza los datos asociados al concepto
 3. Si es un XLS
 - 3.1. Seleccionar el nombre del documento como identificador del CC
 - 3.2. Seleccionar las columnas de la primera fila como propiedades del CC
 - 3.3. Si se indicó un campo categoría
 - 3.3.1. Seleccionar el contenido de la propiedad que coincide con el campo categoría
 - 3.4. Se contabiliza los datos asociados al concepto
 4. Si es un XSD
 - 4.1. Recorrer el documento
 - 4.2. Seleccionar el “xsd:element” principal como CC
 - 4.3. Seleccionar los “xsd:attribute” y “xsd:element” anidados como propiedades del CC
 5. Si es un OWL
 - 5.1. Buscar la Ontología en el DO
 - 5.1.1. Si no existe, registrar en el DO la ontología indicando una descripción y su ubicación.
-
-

5.1.2. Seleccionar cada clase y sus object property como CC.
6. Retornar los CCs
Salidas: Un conjunto de CC

C. Componente de Análisis de Conceptos

Objetivo: analizar los conceptos registrados en la BCS, en base a su relevancia y a sus propiedades para que puedan ser considerados como CE.

Descripción General: representa el “mecanismo de agregación”, donde se analizan y clasifican los conceptos registrados en la TAS, para poder generar los CE. Se busca crear también una integración entre conceptos y meta-conceptos y se actualiza las TMC (Def. 3.13). Todo el proceso de puede observar en el Macro-Algoritmo 3.5. En primer lugar, se analizan los GR de cada concepto. Si el GR de un CO aumentó (Paso 1) y es mayor al UMR (Def. 3.16), entonces el concepto cambia a CC. Si es un CC (Paso 3) y su GR es igual o mayor al URR) (Def. 3.17), entonces el concepto pasa a ser CE, pero si es menor al UMR pasa a ser CC. Si es un CE (Paso 3) y su GR es menor al URR, entonces el concepto pasa a ser CC.

Luego de analizar la relevancia de los conceptos y clasificarlos, se deben procesar los CE (Paso 4) para asociarlos a los meta-conceptos y agruparlos, y así organizarlos en la MOD. La MOD representa un elemento clave dentro del mecanismo de agregación, ya que permite recolectar el aprendizaje individual, organizarlo, clasificarlo, y llevarlo a un aprendizaje colectivo explícito, siendo un paso clave en el proceso de emergencia ontológica. Primero, se buscan los MCs (Paso 4.1); para asociar un CE con un MC, se debe seleccionar al que tiene mayor similitud (Paso 4.1.1.2). La similitud entre un CC y un MC se realiza en base a la Similitud Conceptual (Sim_C, Def. 3.17). Se actualiza en la TMC el MC seleccionado, con los nuevos descriptores que aporta el CE. Si no se encuentra ningún MC con el que tenga algún tipo de similitud, entonces se coloca que el CE pertenece a “thing”, que es la clase raíz de todos los conceptos en una ontología. Si no existe ningún MC, entonces se define el primer MC (Paso 4.1.2.1), colocándole como descriptores las propiedades del CE, y se agrega en la TMC. Finalmente, se actualiza el CE en la TAS con la nueva información (Paso 4.2).

MACRO-ALGORITMO 3.5: ANÁLISIS DE CONCEPTOS

 Entradas: "Base de Conocimientos Semántico" (BCS)

Procedimiento:

1. Por cada CO registrado en la TAS
 - 1.1. Si GR aumento y es mayor al UMR, el concepto pasa a ser CC.
2. Por cada CC registrado
 - 2.1. Si el GR es igual o mayor al URR se coloca como CE.
 - 2.2. Si el GR es menor al UMR se coloca como CO.
3. Por cada CE
 - 3.1. Si el GR es menor al URR se coloca como CC.
4. Por cada CE
 - 4.1. Buscar un MC para asociar el CE.
 - 4.1.1. Si existen MC
 - 4.1.1.1. Determinar medida de similitud entre el CE y los MC de la MOD
 - 4.1.1.1.1. Calcular similitud en base a propiedades :Sim_P
 - 4.1.1.2. Seleccionar el MC que posee mayor medida de similitud
 - 4.1.1.3. Actualizar los descriptores del MC con las propiedades del CC entrante
 - 4.1.2. Si no existen MC
 - 4.1.2.1. Se define el primer MC.
 - 4.1.2.2. Se agrega el MC a la TMC.
 - 4.2. Asociar el CE al MC seleccionado en la TAS.

 Salidas: "Base de Conocimientos Semántico" (BCS) actualizada

3.7.2. Proceso Estructuración Ontológica

A. Componente de Integración Ontológica

Objetivo: integrar los MCs y los CEs en la MOD, calculando y evaluando su entropía, para obtener la mejor organización posible.

Descripción General: en este proceso de integración ontológica se organizan los MCs, usando la EC (Def. 3.7) como la métrica que permite determinar qué tan buena o mala ha sido la estructuración de la ontología, considerando para ello los umbrales que se definieron al inicio del proceso. Todo el proceso de integración ontológica se puede observar en el Macro-Algoritmo 3.6. Como entrada se recibe la BCS, ya analizada con los CEs y los MCs definidos. Por cada CE definido (Paso 1), se actualiza el valor de la EC del MC asociado (de acuerdo a Ec. 3.4) (Paso 1.1). Luego de calculada la entropía, se procede a analizarla (Paso 1.2) para decidir si es necesario reorganizar toda la MOD. Si la EC aumentó, entonces quiere decir que el CE ha tenido un impacto negativo al ingresarlo a ese MC, por lo tanto se debe recalcular la ECG de la MOD (Def. 3.8), para determinar si es necesario reorganizarla. Este proceso de re-organizar toda la MOD es un proceso costoso en cuanto a su orden de magnitud, por lo tanto sólo se debe realizar cuando la ECG supere el UME predefinido. Como vemos, los valores definidos inicialmente de estos umbrales serán determinantes en la eficiencia y calidad de todo el proceso. Si la ECG es mayor al UME, entonces se debe realizar una re-agrupación de MC y se invoca el proceso de Organización de la MOD (Macro-Algoritmo 3.7). Luego de re-organizar la MOD, se registran los MCs resultantes con sus valores de entropía en la TMC, y en la TAS se actualizan los MCs asociados a los CE. Los nuevos CE se agregan a la OE (Paso 1.3), registrándose su clase base y sus propiedades.

Aquellos CEs que por su bajo GR pasaron a ser CCs (Paso 2), deben ser eliminados de la OE, por lo tanto se debe actualizar la EC del MC asociado, y actualizarlo en la TMC.

MACRO-ALGORITMO 3.6: INTEGRACIÓN ONTOLÓGICA

Entradas: "Base de Conocimientos Semántico" (BCS)

Procedimiento:

1. Procesar cada CE registrado en la TAS
 - 1.1. Actualizar el valor de la EC del MC asociado al CE.
 - 1.2. Analizar el valor de la EC
 - 1.2.1. Si la EC aumentó
 - 1.2.1.1. Calcular el valor de la ECG
 - 1.2.1.2. Si el valor de la ECG \geq UME
 - 1.2.1.2.1. Re-agrupación de conceptos y Organización de la MOD (Macro-Algoritmo 3.7)
 - 1.2.1.2.2. Registrar los MC resultantes con sus valores de EC en la TMC y actualizar la TAS.
 - 1.3. Agregar el CE a la OE
 - 1.3.1. Registrar clase base
 - 1.3.2. Registrar propiedades
2. Si un CE paso a ser CC
 - 2.1. Actualizar el valor de la EC del MC asociado al CE.
 - 2.2. Actualizar la TMC
 - 2.3. Eliminar el CE de la OE
-

Salidas: OE

El proceso de Organización de la MOD se muestra en el Macro-Algoritmo 3.7. En este proceso se realiza una adaptación del proceso de *Agrupamiento Jerárquico Divisivo*¹⁷ (Paso 2), donde se comienza con un MC inicial general que contiene a todos los conceptos y en cada ciclo se divide en MCs más específicos. Se inicia el ciclo (Paso 3) y en cada iteración se calcula EC (Ec. 3.4) de cada MC (Paso 3.1) y la ECG (Ec. 3.5) (Paso 3.2), hasta obtener una ECG menor al UME (Paso 3.3). Eso indica la culminación del proceso de agrupamiento. Mientras la ECG sea alta, se dividen los MC de mayor EC, para disminuir más rápidamente el valor de ECG. Luego de definidos los nuevos MCs, se organizan las relaciones de "sub-clase" (Paso 5) entre conceptos para estructurar la nueva MOD.

MACRO-ALGORITMO 3.7: ORGANIZACIÓN DE LA MOD

Entradas: MOD

Procedimiento:

1. Salir=False
 2. Agrupar los CE en MC (Agrupamiento Jerárquico Divisivo)
 3. Mientras Salir<>True
 - 3.1. Calcular la EC de cada MC
 - 3.2. Calcular la ECG de la MOD
 - 3.3. Si ECG < UME
 - Salir=true
 - 3.4. Sino
 - Dividir el MC de mayor EC
 4. Organizar los MC que sean "sub-clase" de otros "MC"
-
-

¹⁷ En minería de datos, el agrupamiento jerárquico es un método de análisis de grupos puntuales, que busca construir una jerarquía de grupos. En el Agrupamiento Jerárquico Divisivo se realiza un acercamiento descendente, se comienzan en un grupo, y se realizan divisiones mientras se baja la jerarquía [115].

Salidas: MOD re-organizada

En aquellos casos en que existen una MOD ya definida y se manejan una gran cantidad de conceptos, se puede poblar la MOD aplicando una técnica de computación emergente (Ver Anexo A).

B. Componente de Calidad

Objetivo: establecer criterios para la evaluación de la calidad de la OE resultante.

Descripción General: en este componente se califica la OE generada de acuerdo a criterios de calidad, que permitirán al usuario experto evaluar la ontología resultante. Para determinar la calidad de una OE se considerarán 3 métricas: Robustez, Completitud y Precisión. Estas medidas determinarán que tan buena, representativa y precisa es una OE con respecto al conjunto de conceptos que están almacenados en la BCS.

De manera general, no se puede decir que existe un modo correcto de modelar conceptualmente un dominio [3], por lo tanto, la OE podrá considerarse una solución inicial, con unos valores de calidad, que si es necesario, pueden ajustarse.

La OE es influenciada directamente por los umbrales de relevancia y entropía que se establecen inicialmente, por lo tanto, las medidas de calidad serán un indicador que permitirá ajustar los umbrales para generar una ontología lo más representativa de la BCS. El proceso de Evaluación de la Calidad se resume en el Macro-Algoritmo 3.8. En ese proceso se tienen como entradas la OE y la BCS. Se realizan los cálculos de las métricas de calidad, y se obtienen los valores de calidad que califican a la OE.

MACRO-ALGORITMO 3.8 : EVALUACIÓN DE CALIDAD

Entradas: OE , BCS

Procedimiento:

1. Calcular las Métrica de Calidad
 - 1.1. Calcular Completitud
 - 1.2. Calcular Robustez
 - 1.3. Calcular Precisión

Salidas: Valores de Calidad de la OE

A continuación se definen las 3 métricas de calidad usadas:

Definición 3.33. Completitud: Una OE es Completa (C) con respecto a una BCS, si los todos los conceptos en la BCS generan un CE. Se puede definir el grado de completitud de la siguiente manera:

$$C(OE, BCS) = \frac{|CE_{OE}|}{|C_{ES}|} \tag{3.10}$$

Donde:

- CE_{OE} : Conjunto de CE en la OE
- C_{ES} : Conjunto de conceptos en la BCS.

Definición 3.34. Robustez: Una OE es Robusta (R) con respecto a una BCS, si sus CEs son relevantes en la BCS, es decir mientras más alto sea el GR de los CE, mayor será la robustez de la OE. Se puede definir el grado de robustez de la siguiente manera:

$$R(OE, BCS) = \frac{\sum_{i=1}^N GR(CE_i)}{N} \quad (3.11)$$

Donde:

- CE_i : i -ésimo Concepto Emergente
- N : Total de CEs
- GR : Grado de Relevancia del CE.

Definición 3.35. Precisión: Una OE es Precisa (P) con respecto a una BCS, si las propiedades de los CEs se corresponden exactamente a los descriptores de los MCs a los que están asociados, por lo tanto es lo contrario de la entropía. El grado de precisión va a ser inversamente proporcional a la entropía de la OE. Si la ECG es 0 la precisión es 1, es decir, la OE será de total precisión si no hay entropía o desorden en ninguno de sus meta-conceptos. Si hay entropía la precisión será menor, a mayor entropía menor precisión. La ECG nunca debe superar el umbral UME, ya que si la iguala o la supera según el Macro-Algoritmo 3.6, la entropía será máxima y por lo tanto se debe reorganizar la MO hasta bajar la ECG. Por lo tanto, se calculará la proporción de la ECG con respecto al umbral, para de allí obtener la precisión, entonces se calculará de la siguiente manera:

$$P(OE, BCS) = \begin{cases} 1 - \frac{ECG}{UME} & \text{Si } ECG < UME \\ 0 & \text{Si } ECG \geq UME \\ 1 & \text{Si } ECG = 0 \end{cases} \quad (3.12)$$

Capítulo 4:

CASOS DE ESTUDIO

En este capítulo se presentan algunos casos de estudio, para mostrar la utilidad del *Esquema de Emergencia Ontológico* (EEO) propuesto en este trabajo. En primer lugar, se aplica el EEO en un contexto de “Datos Abiertos” (*Open Data*¹⁸, en inglés), donde en base a un conjunto de documentos de entrada obtenidos de estos portales, se muestra cómo se puede aplicar el proceso de emergencia ontológica. Luego, se explica cómo puede ser usado en Ambientes Inteligentes, y por último, se muestra su utilidad en Ciudades Inteligentes (*Smart Cities*, en Inglés). Para mostrar la aplicación del EEO se implementó una herramienta, para lo cual se usó el entorno de desarrollo NetBeans 7.4 con la plataforma JADE (“*Java Agent DEvelopment Framework*”, por sus siglas en inglés). Además, también se usó JENA, un framework para Java que provee librerías para el manejo de RDF y OWL. En la Fig. 4.1, se puede observar la interfaz de la herramienta. Para más detalles sobre la herramienta ver Anexo B.



Figura 4.1. Aplicación EEO

Inicialmente, se deben configurar los valores permitidos para los umbrales: Umbral Mínimo de Relevancia (UMR) (Def. 3.18), Umbral Requerido de Relevancia (URR) (Def. 3.19), Umbral Máximo de Entropía (UME) (Def. 3.20), así como la Tasa de Estabilización de Relevancia (TER) (Def. 3.21). La TER se inicializa por defecto en un valor muy pequeño (0.01), aunque si se desea se puede

¹⁸ Open Data o Datos Abiertos es una filosofía y práctica que persigue que determinados tipos de datos estén disponibles de forma libre para todo el mundo, sin restricciones de derechos de autor, de patentes, o de otros mecanismos de control de los datos [116].

colocar un valor mayor. Luego de inicializar los umbrales, se deben seleccionar los Eventos Semánticos (ES) (Def. 3.2) que serán procesados, y si existen Meta-Ontologías de Dominio (MOD) se debe indicar, o si se va a generar las MOD. Se puede seleccionar si se visualizan las salidas por cada ES que se procesa, o al final de procesar todos los ES de entrada. Al ejecutar el EEO, se genera la Ontología Emergente y se muestran los valores de calidad de la ontología, y si se desea, se pueden ver la Tabla de Anotaciones Semánticas (TAS) (Def. 3.14), y la Tabla de Meta-Conceptos (TMC) (Def. 3.16).

4.1. Aplicación del EEO en el contexto de Datos Abiertos y Ontologías.

En este caso de estudio, el objetivo es mostrar la *Ontología Emergente* (OE) (Def. 3.25) luego de la aplicación del EEO, considerando un conjunto de ES de entrada, que se clasifican en 3 escenarios diferentes: datos, ontologías y datos-ontología. Esta diferenciación se realiza con el objetivo de visualizar las variaciones que tendrán las OEs, considerando por un lado sólo datos que representan las instancias de los conceptos, por otro lado sólo ontologías donde ya se tiene una representación del conocimiento y no tenemos instancias asociadas, y finalmente, la combinación de ambas. En cada escenario se consideran variaciones en los umbrales para evaluar su efecto en las métricas de calidad, que califican la OE resultante con respecto a los conceptos que se quieren representar almacenados en la *Base de Conocimientos Semánticos* (BCS)(Def. 3.13).

Las fuentes de datos usadas son referentes al dominio de “Turismo”, es decir, datos y ontologías que manejan información sobre el turismo de una región. En este caso los datos se toman de un catálogo de *Open Data* de la administración pública de una ciudad española [85] [86]. Esta red de datos es usada para el desarrollo de productos y servicios que dan soporte a las actividades de las unidades del gobierno y de la ciudad en general, entre ellas el turismo. Se considera éste un caso de SAOyE, ya que allí se presenta el surgimiento espontáneo de información proveniente de diversas fuentes, para formar una red de datos libre. Esta red de datos es dinámica, ya que con frecuencia se están incorporando nuevos datos, por lo que su proceso de actualización es abierto. Las ontologías se tomarán de diferentes catálogos de ontologías disponibles en la web.

La OE representará conceptos referentes al área del turismo, y podría utilizarse luego para el desarrollo de aplicaciones, por ejemplo, servicios para la atención del turista, o como una forma de estandarizar y estructurar los datos,

para que luego puedan ser enlazados y conectados, por ejemplo usando Datos Enlazados (*Linked Data*¹⁹, en inglés).

Descripción general del proceso.

Para aplicar el EEO en el caso de estudio, seguiremos el proceso que se muestra en el Macro-Algoritmo 3.1. En primer lugar, se deben establecer las entrada del proceso, que serán los ES. Los diferentes actores del sistema ponen a disposición sus datos, que representan los ES que pueden registrar en el EEO. Para efectos de mostrar las iteraciones del EEO, se tomarán algunas fuentes de datos del caso de estudio, para el escenario Nro. 1, que se listan en la Tabla 4.1. Estas fuentes de datos se someten a un pre-procesamiento para depurarlas, eliminar información innecesaria, y tratar de unificar la identificación de algunos campos. Por cada ES se realiza el “Registro Semántico” (Macro-Algoritmo 3.3). Cada ES puede aportar uno o más conceptos, que poseen un conjunto de instancias asociadas y propiedades definidas (como se muestran en la Tabla 4.2) luego del proceso de “Búsqueda de CC”.

TABLA 4.1. EVENTOS SEMÁNTICOS

Fuente de Datos	Tipo Formato
LugaresTuristicos.xls	Tabla Excel XLS
Bares.xls	Tabla Excel XLS
Restaurante.xls	Tabla Excel XLS
ActividadesyEventos.xml	Documento XML
ObrasdeArte.xml	Documento XML
Museo.xml	Documento XML
Alojamiento.xls	Tabla Excel XLS

Por cada ES que ingresa al sistema se ejecuta una iteración, donde cada componente del framework (Registro Semántico, Análisis, Integración y Calidad) realiza su función, para dar como resultado una OE (o actualización de una OE existente). Describiremos a continuación las primeras 3 iteraciones, para visualizar el proceso de manera general. Para el primer escenario se establecieron los siguientes umbrales, luego de un proceso de calibración, donde se colocaron unos valores iniciales y se ajustaron de acuerdo a los resultados obtenidos, para obtener OEs de diferente calidad:

- UMR: 0.05
- URR: 0.28
- UME: 1

¹⁹ Linked Data ó Datos Enlazado, describen un método de publicación de datos estructurados que les permite estar interconectados entre si para todo el mundo, sin restricciones de derechos de autor, de patentes, o de otros mecanismos de control de los datos [116].

TABLA 4.2. CONCEPTOS CANDIDATOS

Fuente	Conceptos	Total Datos	Propiedades
LugaresTurísticos.xls	Palacio	6	Nombre, Ubicación
LugaresTurísticos.xls	Torre	5	Nombre, Ubicación
LugaresTurísticos.xls	Plaza	26	Nombre, Ubicación
LugaresTurísticos.xls	Iglesia	22	Nombre, Ubicación
LugaresTurísticos.xls	Cueva	4	Nombre, Ubicación
Bares.xls	Bar-Restaurant	38	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Cafetería	7	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Vinatería	1	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Pizzería	4	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Tapería	3	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Pub	4	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Cervecería	3	Nombre, Capacidad, Ubicación, TipoComidaBebida
Bares.xls	Heladería	1	Nombre, Capacidad, Ubicación, TipoComidaBebida
Restaurante.xls	Restaurante	94	Nombre, Capacidad, Ubicación, TipoComida
Restaurante.xls	Bar-Restaurant	20	Nombre, Capacidad, Ubicación, TipoComida
ActividadesyEventos.xml	Teatro	6	Nombre, Fecha, Precio
ActividadesyEventos.xml	Exposicion	2	Nombre, Fecha, Precio
ActividadesyEventos.xml	Danza	4	Nombre, Fecha, Precio
ActividadesyEventos.xml	Desfile	10	Nombre, Fecha, Precio
ActividadesyEventos.xml	Concierto	8	Nombre, Fecha, Precio
ActividadesyEventos.xml	Conferencia	12	Nombre, Fecha, Precio
ObrasdeArte.xml	Pintura	25	Nombre, Autor, Material
ObrasdeArte.xml	Escultura	18	Nombre, Autor, Material
ObrasdeArte.xml	Monumento	3	Nombre, Autor, Material
ObrasdeArte.xml	Estatua	8	Nombre, Autor, Material
Museo.xml	Museo	10	Nombre, Ubicación, Precio, Exposición
Alojamiento.xls	Hostal	12	Nombre, Capacidad, Ubicación
Alojamiento.xls	Hotel	25	Nombre, Capacidad, Ubicación
Alojamiento.xls	Posada	13	Nombre, Capacidad, Ubicación
Alojamiento.xls	Apartamento Turístico	14	Nombre, Capacidad, Ubicación

• Iteración 1:

- ES: “LugaresTurísticos.xls”.

○ **Registro Semántico:** De acuerdo al Macro-Algoritmo 3.3, se debe realizar el registro en la *Tabla de Anotaciones Semánticas (TAS)*(Def. 3.14) de los *Conceptos Candidatos (CC)*(Def. 3.22): Palacio, Torre, Plaza, Iglesia y Cueva. Se colocan sus propiedades, instancias (datos asociados) y frecuencia (inicialmente en 1). Para calcular el *Grado de Relevancia (GR)*(Def. 3.12), se deben primero normalizar los valores de Instancias (I) y Frecuencia (F). Por ejemplo, el GR del concepto “Iglesia” se calcula de la siguiente manera:

- Primero se obtiene el I_N y el F_N (valores de I y F normalizados), en base a la Ec. 3.6:

$$I_N = \frac{22}{26} = 0.846$$

$$F_N = \frac{1}{1} = 1$$

- Luego se calcula el GR, en base a la Ec. 3.7:

$$GR = \frac{(0.846 + 1)}{2} = 0.923$$

El GR puede cambiar a medida que entran nuevos CCs, ya que se modifican los mínimos y máximos de I y de F.

- **Análisis:** Siguiendo el Macro-Algoritmo 3.5, por cada CC, se evalúa el GR para determinar si puede pasar a Concepto Emergente (CE). En este caso, de acuerdo al URR definido de 0.28, todos los CCs pasan a ser CEs. Luego, se debe asociar cada CE con un Meta-Concepto (MC) (Def. 3.3). En este caso, como aún no hay MCs definidos, se define el primer MC colocando como Descriptores las propiedades del CE : “Nombre” y “Ubicación” (De acuerdo a la Def. 3.3).

- **Integración:** Siguiendo el Macro-Algoritmo 3.6, se calcula la *Entropía Conceptual* (EC) (Def. 3.7) de los MCs asociados a los CEs, de acuerdo a la Ec. 3.4.

- Se tienen los descriptores $D = (\text{“Nombre”, “Ubicación”})$.

Ya que todos los conceptos poseen los descriptores, la proporción (Pr) de conceptos con cada descriptor será $Pr = (1,1)$, entonces el valor de la Entropía será:

$$\begin{aligned} EC(MC_1) &= \sum_{j=1}^2 Pr(D_j) \log Pr(D_j) = Pr(D_1) \log Pr(D_1) + Pr(D_2) \log Pr(D_2) \\ &= 1 \log 1 + 1 \log 1 = 0 \end{aligned}$$

Se actualiza la OE.

- **Calidad:** Se calculan las métricas de calidad de la OE.

- **Compleitud:** De acuerdo a la Ec. 3.10 se calcula la completitud.

$$C(OE, BCS) = \frac{|CE_{OE}|}{|CE_S|} = \frac{5}{5} = 1$$

- **Robustez:** se realiza el cálculo en base al promedio del GR de lo CE , según la Ec. 3.11.

$$R(OE, BCS) = \frac{\sum_{i=1}^N GR(CE_i)}{N} = \frac{0.62+0.59+1+0.92+0.58}{5} = 0.742$$

- **Precisión:** De acuerdo a las Ec. 3.12 como la *Entropía Conceptual General* (ECG) es igual a 0, la precisión toma el valor máximo.

$$P(OE, BCS) = 1, \text{ ya que ECG es igual a } 0$$

Los resultados de la Iteración 1 se pueden observar en la Fig. 4.2.

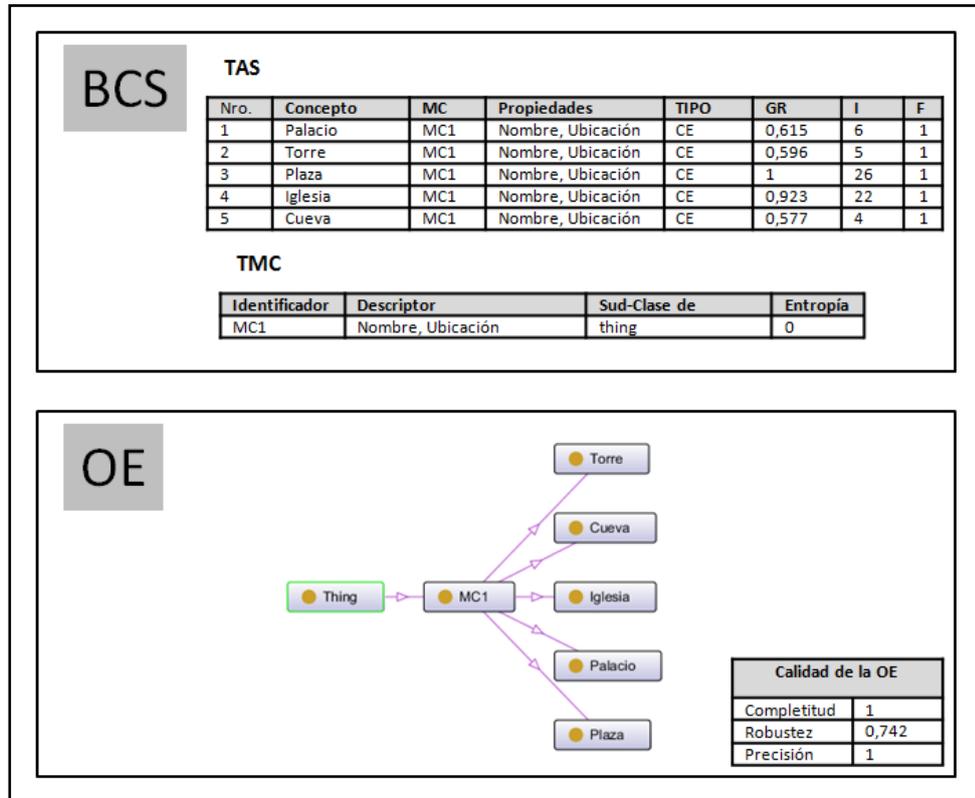


Figura 4.2. Iteración 1

• **Iteración 2:**

- **ES:** “Bares.xls”.

- **Registro Semántico:** Se realiza la búsqueda de los CCs en la TAS, como no están se registran con sus propiedades, instancias y frecuencia. Se vuelve a calcular el GR de los conceptos registrados, ya que cambian los valores de I y de F.

- **Análisis:** Se evalúa el GR de cada concepto y todos son mayores a 0.28, por lo tanto, pasan a ser CEs. Se asocia cada CE con un MC, calculando Sim_P (Ec. 3.8). Por ejemplo, la Sim_P entre el concepto “cafetería” y el concepto “MC1” es:

$$Sim_P("cafetería", "MC1") = \frac{2}{4} = 0.5$$

Como sólo hay un MC, se asocian a éste y se agregan los Descriptores: “Capacidad” y “TipoComidaBebida”.

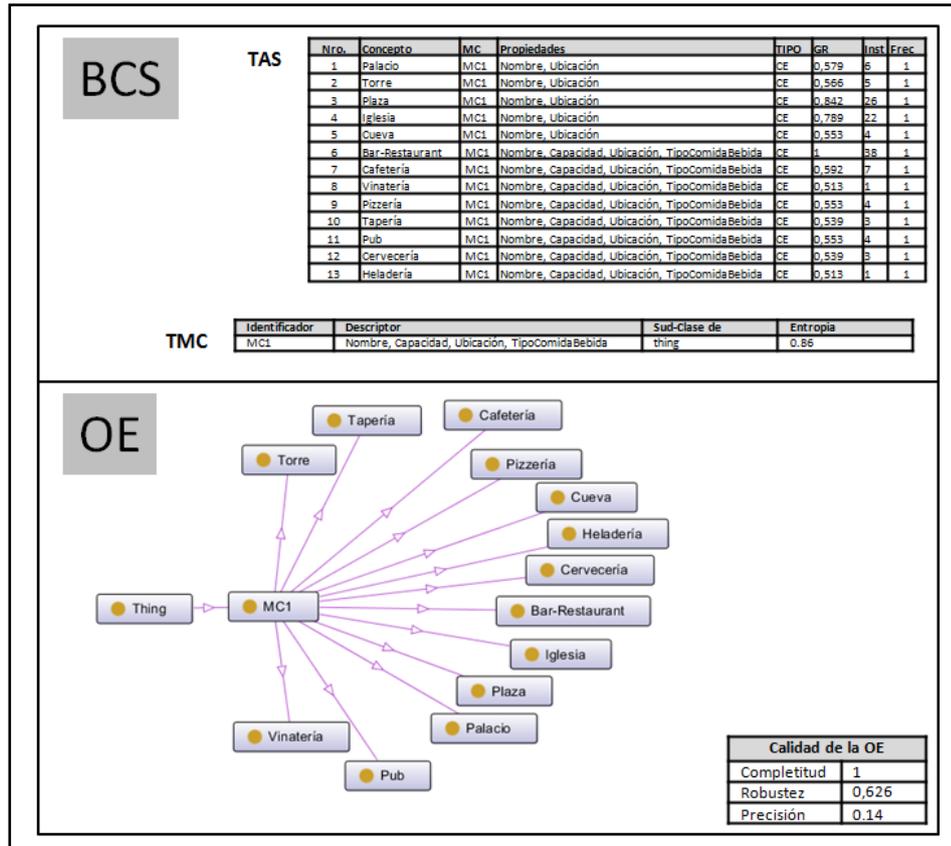


Figura 4.3. Iteración 2

- **Integración:** Se procesan los nuevos CEs. Se vuelve a calcular la EC de MC1 .

Se tienen los descriptores $D=(\text{"Nombre", "Ubicación", "Capacidad", "TipoComidaBebida"})$. Para calcular Pr , se tiene que el total de conceptos es 13, y la propiedad "capacidad" y "tipoComidaBebida" sólo la tienen 8 conceptos. Por lo tanto, de acuerdo a la Ec. 3.4, la proporción de conceptos con cada descriptor es $Pr = (1, 1, 0.62, 0.62)$ y el valor de la EC será:

$$EC(MC_1) = \sum_{j=1}^4 Pr(D_j) \log Pr(D_j) = 1 \log(1) + 1 \log(1) + 0.62 \log(0.62) + 0.62 \log(0.62) = 0.86$$

Como la EC aumentó, se calcula la ECG usando la Ec. 3.5:

$$ECG(MO) = \sum_{i=1}^N \frac{|MC_i|}{|MO|} \times EC(MC_i) = \frac{13}{13} \times 0.86 = 0.86$$

Finalmente, se agregan los CEs en la OE.

- **Calidad:** Se calculan las métricas de calidad de la OE:

- **Complejidad:**

$$C(OE, BCS) = \frac{|CE_{OE}|}{|C_{ES}|} = \frac{13}{13} = 1$$

- **Robustez:**

$$R(OE, BCS) = \frac{\sum_{i=1}^N GR(CE_i)}{N} = \frac{0.58+0.57+0.84+0.78+0.55+1+0.59+0.51+0.55+0.53+0.55+0.53+0.51}{13} = 0.626$$

- **Precisión:** Como la ECG es menor al UME, de acuerdo a la Ec. 3.12:

$$P(OE, BCS) = 1 - \frac{ECG}{UME} = 1 - 0.86 = 0.14$$

Los resultados de la Iteración 2 se pueden observar en la Fig. 4.3.

- **Iteración 3:**

- **ES:** "Restaurante.xls".

- **Registro Semántico:** Se buscan los CCs y se registran en la TAS. El concepto "Bar-Restaurant" ya existe, se incrementan sus instancias a 58 y su frecuencia a 2. El concepto "Restaurante" no existe, por lo que se agrega. Sus instancias son 94, como son altas, modifican considerablemente el GR de todos los conceptos. Por otro lado, la propiedad "TipoComida", de estos nuevos conceptos, posee una similitud con "TipoComidaBebida" de 0.8, por lo tanto se consideran equivalentes (Def. 3.17).

- **Análisis:** Se evalúa el GR de cada concepto, y sólo de 6 conceptos superan el URR 0.28, por lo tanto, pasan a ser CEs. Los demás quedan como CC. Se asocia cada CE con el MC1.

- **Integración:** Se procesan los nuevos CEs. Se vuelve a calcular la EC de MC1.

Se tienen los descriptores:

D=("Nombre", "Ubicación", "Capacidad", "TipoComidaBebida")

Para calcular Pr, se tiene que el total de conceptos es 6, y la propiedad "capacidad" y "tipoComidaBebida" la tienen 3. Por lo tanto, de acuerdo a la Ec. 3.7, la proporción de conceptos con cada descriptor es Pr = (1, 1, 0.5, 0.50) y el valor de la EC será:

$$\begin{aligned}
 EC(MC_1) &= \\
 &= \sum_{j=1}^4 \Pr(D_j) \log \Pr(D_j) = 1 \log(1) + 1 \log(1) + 0.5 \log(0.5) \\
 &\quad + 0.5 \log(0.5) = 1
 \end{aligned}$$

Como la EC aumentó, se calcula la ECG :

$$ECG(MO) = \sum_{i=1}^N \frac{|MC_i|}{|MO|} \times EC(MC_i) = \frac{6}{6} \times 1 = 1$$

Como la ECG alcanzó el máximo permitido (UME=1), se debe re-organizar la Meta-Ontología de Dominio (MOD), específicamente se ejecuta el proceso de Organización de la MOD (Macro-Algoritmo 3.7). En este proceso, se genera una MOD con un valor de entropía 0, con 2 MCs con los siguientes descriptores:

MC1=(“Nombre”, “Ubicación”)

MC2=(“Nombre”, “Ubicación”, “Capacidad”, “TipoComidaBebida”)

De acuerdo a la Def. 3.9, se organiza a MC2 como “sub-clase” de MC1, ya que los descriptores de MC1, están incluidos entre los descriptores de MC2. Se agregan los CEs en la OE.

○ **Calidad:** Se calculan las métricas de calidad de la OE:

- **Compleitud:**

$$C(OE, BCS) = \frac{|CE_{OE}|}{|C_{ES}|} = \frac{6}{14} = 0.428$$

- **Robustez:**

$$R(OE, BCS) = \frac{\sum_{i=1}^N GR(CE_i)}{N} = \frac{0.28+0.39+0.37+0.81+0.29+0.75}{6} = 0.48$$

- **Precisión:**

$$P(OE, BCS) = 1, \quad \text{ya que ECG es igual a 0, según la Ec. 3.12.}$$

Los resultados de la Iteración 3 se pueden observar en la Fig. 4.4.

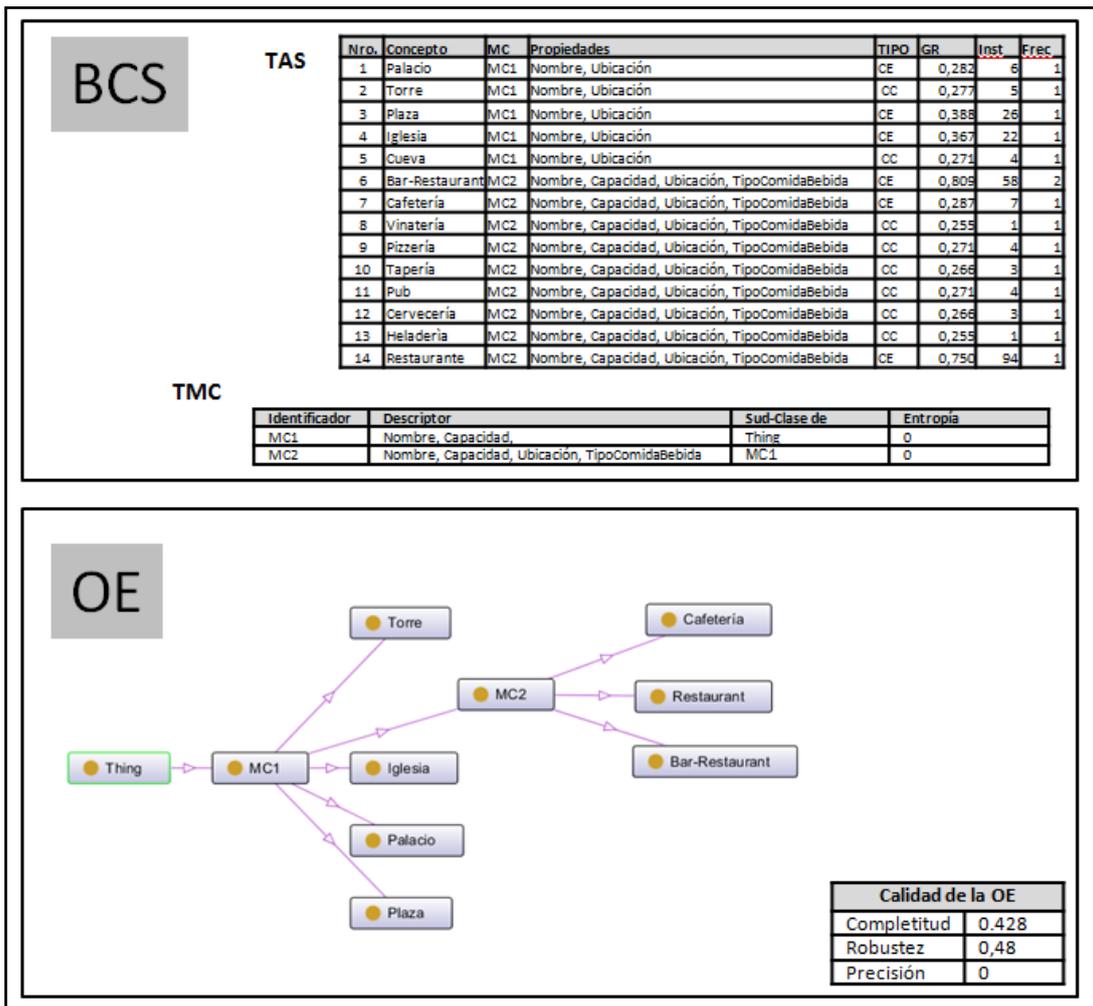


Figura 4.4. Iteración 3

Descripción de los Escenarios de Prueba

- **Escenario 1:**

Como primer escenario se trabaja sólo con datos. El objetivo es generar una OE que represente el conjunto de datos suministrados. Algunas fuentes de datos de ésta área que se tomaron para este caso de estudio se muestran en la Tabla 4.1. Parte de los conceptos que aportan las fuentes de datos luego del proceso de “Búsqueda de CC” se muestran en la Tabla 4.2.

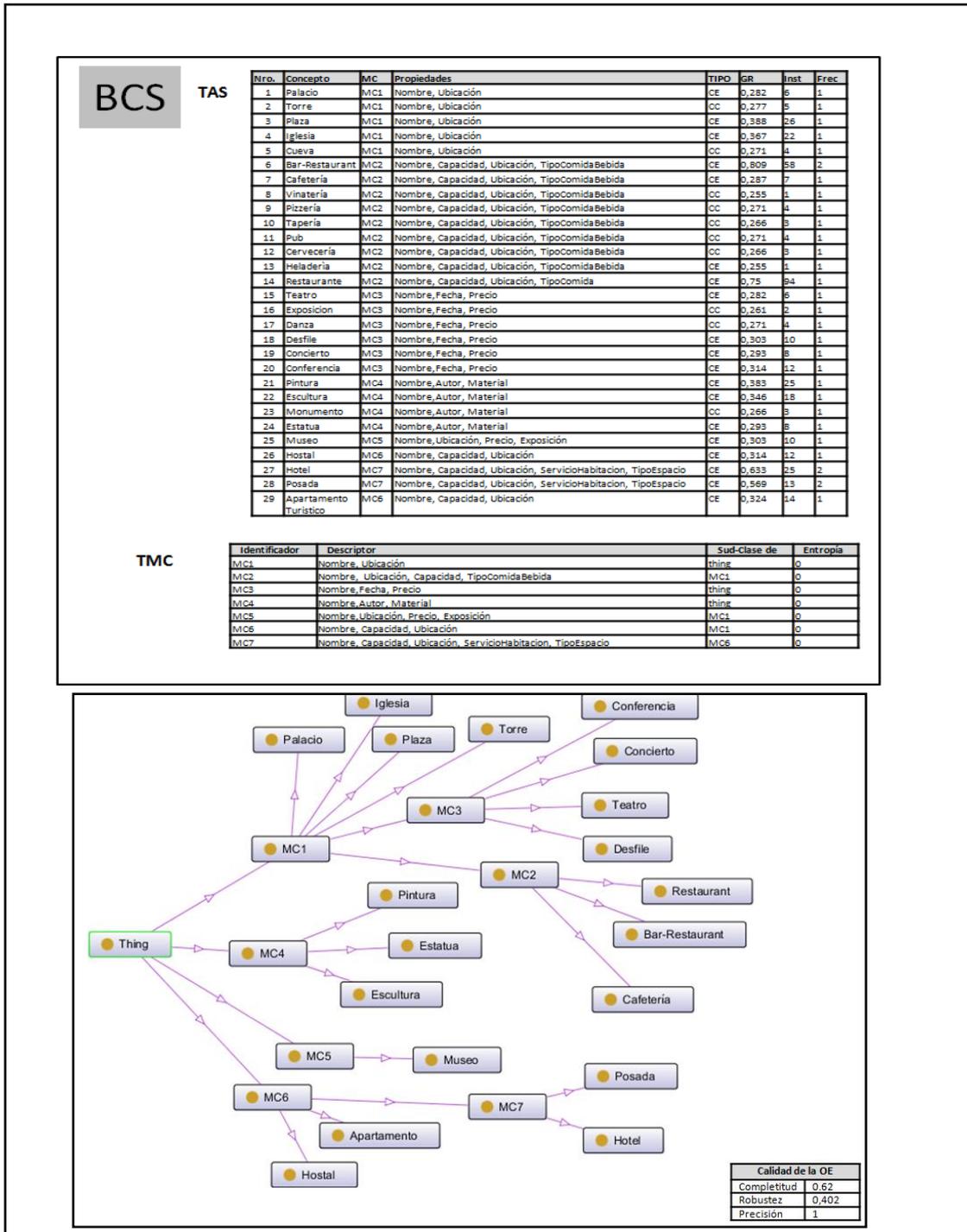


Figura 4.5. OE del Escenario 1

Luego de aplicar el EEO, en la Fig. 4.5 se puede observar la OE con sus métricas de calidad: Completitud=0.62, Robustez=0.402 y precisión=1. En este caso, la ontología resultante cubre en un 62% aproximadamente los conceptos de la BCS. La robustez se calcula en base al promedio del GR de los CEs, en este caso, los CEs son en promedio 40% relevantes. Ese valor podría parecer una relevancia un poco baja, pero se debe considerar que el URR es de 28%, lo que significa que en la OE se está permitiendo la entrada de CEs de baja relevancia. Si se desea aumentar la robustez, se puede elevar el valor del umbral URR, pero puede implicar bajar la cantidad de CE, y por lo tanto, bajar la completitud. Por ejemplo, en la Fig. 4.6 se puede observar la OE resultante si se aumenta el umbral URR a 30%, tal que su completitud bajará a 44%. Es por ello que los valores dados a los umbrales al inicio, son determinantes en la calidad de la OE. La precisión es 100 % en ambos casos, es decir, que en la OE todos los CEs están agrupados adecuadamente.

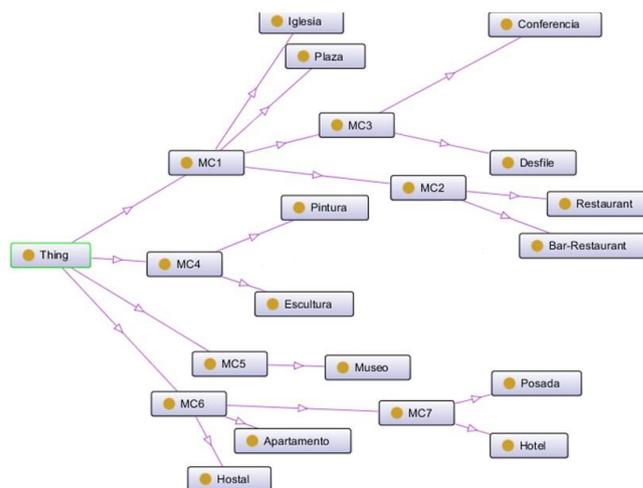


Figura 4.6. OE del Escenario 1 al variar el URR a 30%

▪ **Escenario 2:**

Como segundo escenario se agrega un nuevo ES que será una ontología en español del área de Turismo [85], que posee conceptos y propiedades en el área de turismo. Parte de la ontología se puede visualizar en la Fig. 4.7, donde se muestra el concepto “Hotel”. En esta ontología están definidas, entre otras propiedades, los “Object Properties”: “TieneTipoEspacio” (asociada a la clase

“TipoEspacio”) y “TieneServiciohabitacion” (asociada a la clase “ServicioHabitación”). Se realiza la búsqueda del concepto “Hotel” en la TAS (Fig. 4.8), como ya existe el concepto, se aportan las nuevas propiedades “TipoEspacio” y “Serviciohabitacion”, y se incrementa la frecuencia del concepto, colocándola en 3, pero no se incrementan las instancias, porque en este caso la ontología no tiene individuos registrados. Este incremento en la frecuencia, afecta el GR de todos los conceptos.

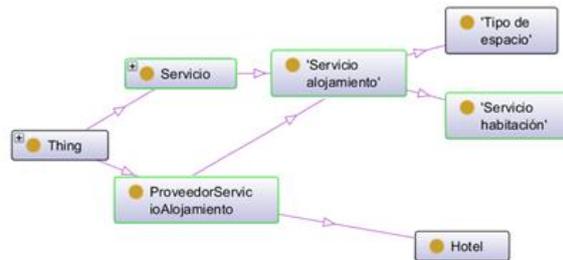


Figura 4.7. Parte de la Ontología Turismo.owl

Al finalizar el proceso, la BSC y la OE se pueden observar en la Fig. 4.8. Al cambiar el GR cambian los CE seleccionados. Si dejamos el mismo URR del escenario anterior sólo se seleccionan 7 CE, y la Completitud de la OE disminuye a un 28%, lo que quiere decir que sólo se estarían representando un 28% de los conceptos de la BCS. Si se varía en este escenario el URR y se coloca en 0.22, la completitud queda en 45%, lo que implica un aumento en el número de CE, y por lo tanto, se están representando más conceptos de la BCS. Sin embargo, la robustez baja de 40% a un 36%, ya que al bajar el URR se incluyen más CE de menor GR, lo que afecta la robustez de la OE. Por otro lado, las nuevas propiedades que aporta el concepto “Hotel” aumentan la entropía del MC6 a 1, pero como está dentro del UME que es 1, es aceptable. La Entropía Conceptual General aumenta un poco, por lo tanto, la precisión se ubica en 85%. Esta disminución de la precisión con respecto al escenario anterior, indica que existe un mayor desorden en los MC, específicamente en el MC6, al incluir un nuevo concepto que posee algunas propiedades diferentes.

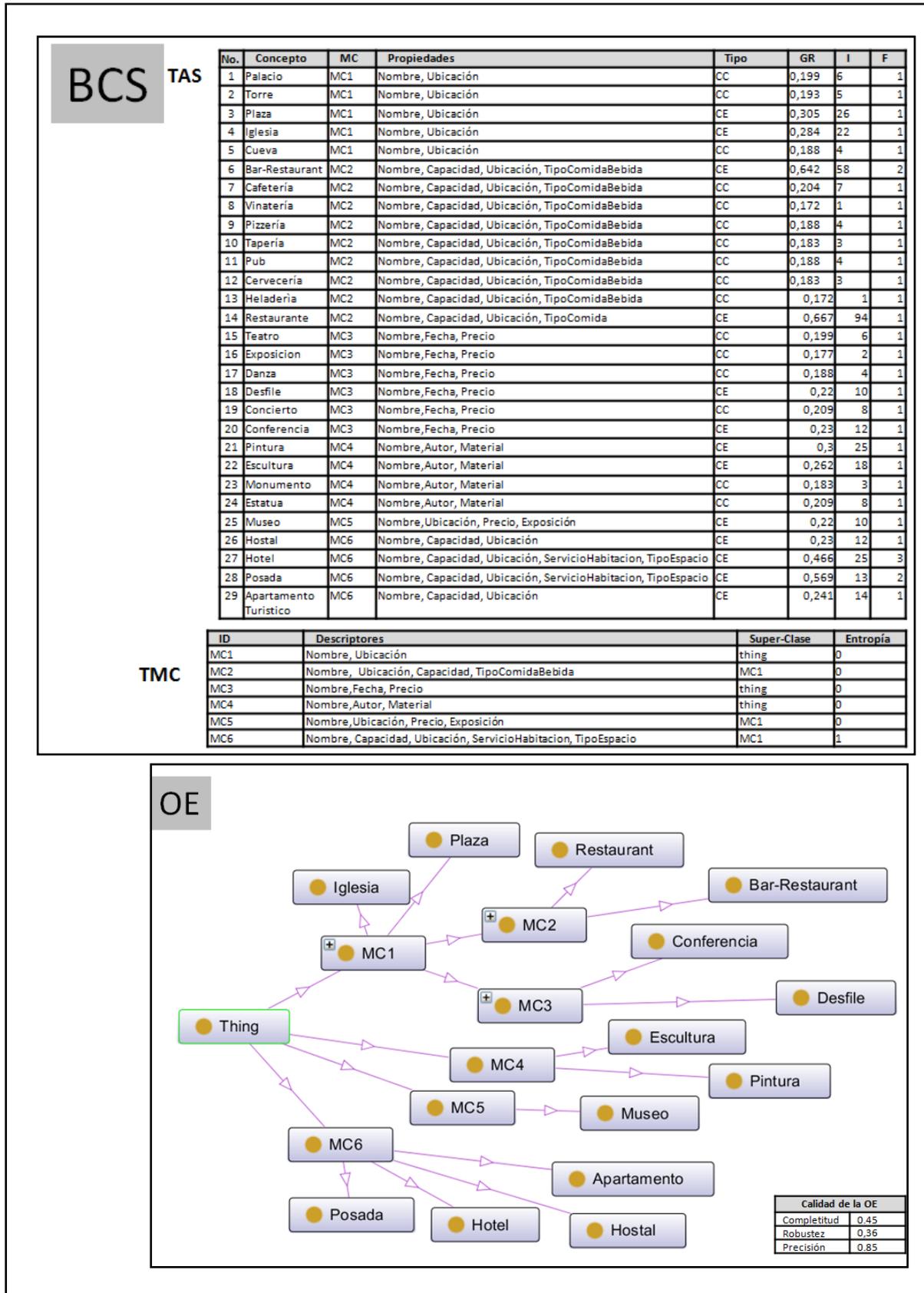


Figura 4.8. OE del Escenario 2

▪ **Escenario 3:**

Como tercer escenario, se usarán diferentes ontologías del área de turismo, que se han obtenido de los catálogos de ontologías Swoogle²⁰ y LOV4IoT²¹. De estos catálogos, se han seleccionado algunas ontologías que se listan en la Tabla 4.3, allí se indican la cantidad de conceptos de cada ontología. En estas ontologías los conceptos están en el idioma inglés. Para este escenario, como se usan sólo ontologías como fuente de datos, que no poseen instancias asociadas, el GR se va a calcular sólo en base a frecuencia. Los valores para los umbrales se colocaron en:

- UMR: 0.2
- URR: 0.5
- UME: 1

Se establecieron estos valores de UMR y el URR, incrementando los umbrales de relevancia, para poder captar los CE de mayor GR. El umbral de entropía se mantiene igual en 1, lo cual se considera un valor aceptable. En los ES, las ontologías presentan una baja homogeneidad, al tener diferencias notables en los identificadores de los conceptos y en sus propiedades, lo que disminuye la frecuencia de repetición del concepto, y por lo tanto, su GR. Esto hace que existan muchos conceptos con bajo GR y alta entropía. Al aumentar los umbrales se busca obtener una OE más pequeña, pero más robusta y precisa.

TABLA 4.3. ONTOLOGÍAS DEL ÁREA DE TURISMO

Ontología	Nro. de Conceptos
qallme-tourism4.owl	120
T438_tourism-uk_no_imports.owl	98
tourism.owl	8
tourismOntology.owl	12
travel.owl	20

Luego de finalizar el proceso del EEO, en la Fig. 4.9 se puede observar la TMC y la OE con sus métricas de calidad: Completitud=15%, Robustez=60% y precisión=45%. Se obtiene una OE no muy completa, pero si con una alta robustez y una precisión medianamente aceptable. Esto debido a que las ontologías que conformaron las fuentes de datos son muy heterogéneas, y con notables diferencias entre sus conceptos y propiedades. Por lo tanto, si se coloca un URR muy bajo se obtendría una OE muy extensa, pero poco robusta y de baja precisión, por la alta entropía de sus MC. Al elevar un poco el URR a 60%, se baja

²⁰ SWOOGLE es un directorio y un motor de búsqueda para recursos RDF disponibles en la Web, incluyendo ontologías [113].

²¹ LOV4IoT (Linked Open Vocabularies for Internet of Things) es un catálogo de ontologías que hace referencia a más de 400 proyectos de basados en ontologías para IoT (Internet de las Cosas), con sus dominios aplicativos [112].

aún más la completitud, llegando a ser 10%, obteniéndose una OE más pequeña, (Fig. 4.10).

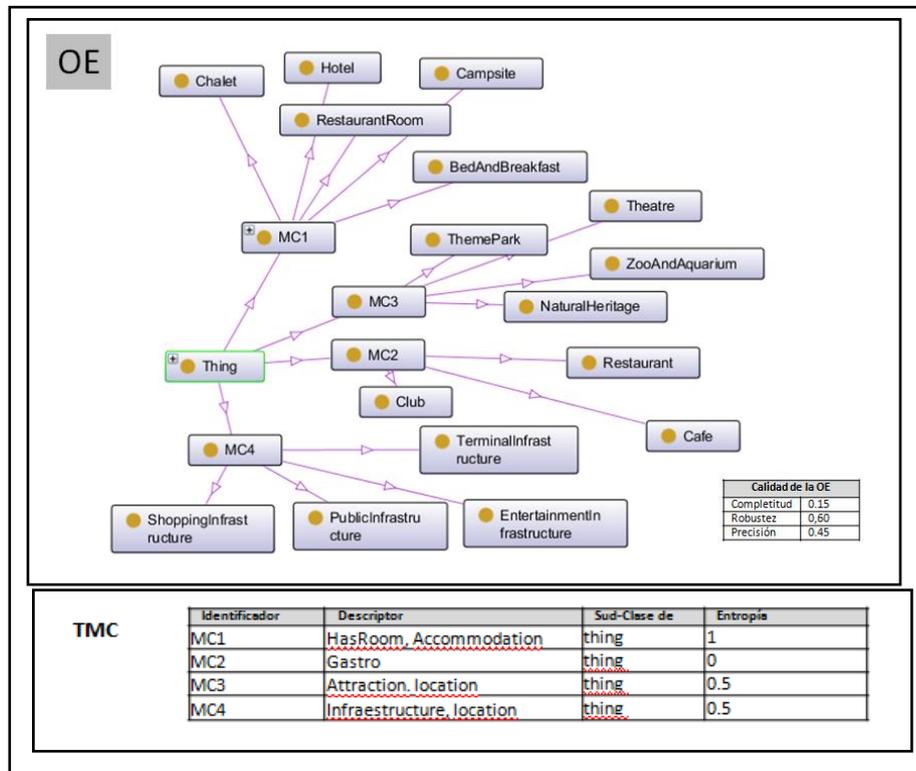


Figura 4.9. OE del Escenario 3

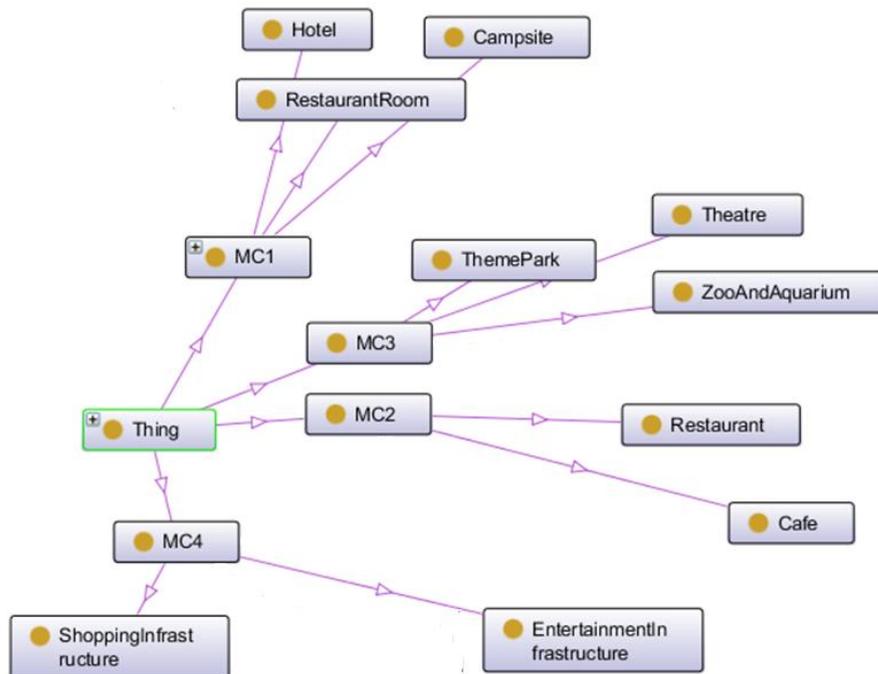


Figura 4.10. OE del Escenario 3 al variar el URR a 60%

4.2. Aplicación del EEO en Ambientes Inteligentes

En este caso, se propone el uso de un middleware reflexivo, llamado MiR-EO, que busca integrar diferentes modelos conceptuales provenientes de los agentes que participan en un Ambiente Inteligente (Aml), para la generación de Ontologías Emergentes (OE). En [87] [88] [89] se presentaron las primeras ideas de esta propuesta. La idea es hacer emerger ontologías, integrando y enriqueciendo meta-ontologías y datos provenientes de las aplicaciones, sensores, dispositivos y servicios, que interactúan en el Aml.

Contextualización

Un Aml es un entorno dinámico que está en constante evolución, y se debe adaptar permanentemente. Los Aml son ambientes físicos, en los que las tecnologías de información y comunicación, y los sistemas de sensores, se integran en los objetos físicos, en la infraestructura, y en el entorno, haciendo que la interacción usuario-sistema sea natural y desinhibida [74]. El objetivo de las infraestructuras de computación de un Aml es proporcionar servicios inteligentes a los usuarios del contexto, de una manera adaptativa y adecuada a sus necesidades [9]. Para ello, es necesario que los dispositivos sean capaces de entender su contexto, y por su naturaleza heterogénea, necesitan también manejar un lenguaje común para el entendimiento y el logro de la interoperabilidad con los demás componentes del Aml [90] [91].

Para la representación del conocimiento de un dominio, así como para describir el contexto, el uso de ontologías es una forma común en el campo de Amls, ya que proveen capacidades de razonamiento, de generación de nuevo conocimiento, lo que permite construir las aplicaciones “inteligentes” requeridas en dichos ambientes [92]. Las ontologías en un Aml son, entonces, una herramienta potente para la representación del conocimiento, ya que permiten establecer un modelo legible y reutilizable para lograr la interoperabilidad semántica en él [93]. Las ontologías son usadas en Aml para describir las propiedades, capacidades e intenciones de los dispositivos, personas y otros objetos físicos en el entorno. Además, permiten conceptualizar información de dominios específicos, entre otras cosas. Esto hace más fácil desarrollar aplicaciones emergentes conscientes del contexto en dichos ambientes, capaces de proveer servicios importantes para los usuarios [91].

Se considera importante, entonces, definir un proceso de aprendizaje ontológico para Amls, que permita enriquecer la ontología de su dominio de

acuerdo a las experiencias del ambiente, a los nuevos elementos en el ambiente, entre otras cosas. Esa es la connotación emergente de la ontología en este caso, entendida como algo que surge de la dinámica del Aml.

En este caso de estudio se muestra como el EEO se ofrece como un servicio dentro de un Aml, a través de la implementación de un middleware reflexivo. La arquitectura propuesta (MiR-EO), está inspirada en el paradigma de computación autónoma, por lo que se diseña como un middleware reflexivo y autónomo. La Computación Autónoma es un modelo computacional de auto-gestión inspirado en el sistema nervioso autónomo de los seres humanos [94]. La arquitectura del modelo de computación autónoma se compone de 6 niveles:

- *Recursos Gestionados*: Son los recursos (hardware o software) que son gestionados.
- *Puntos de Enlace*: Conjunto de sensores y actuadores que se incorporan al sistema para gestionar los recursos.
- *Gestor Autónomo*: Implementa el lazo de control inteligente, que automatiza las tareas de autorregulación de la aplicación. Se compone de cuatro fases identificadas como **MAPE**: **M**onitoreo (obtiene datos y eventos de los sensores), **A**nálisis (donde se da el diagnóstico), **P**lanificación (se definen las acciones a hacer sobre el proceso), y **E**jecución (se envían las órdenes a los componentes a través de los actuadores).
- *Orquestador de Gestores Autónomos*: Si existen varios gestores autónomos que necesitan trabajar en conjunto, proporciona el canal de comunicación para la coordinación entre ellos.
- *Manejador Manual*: Permite a los usuarios configurar los gerentes autónomos, para realizar sus tareas de autogestión, a través de interfaces hombre-máquina.
- *Fuentes de Conocimiento*: Proporciona el acceso a los conocimientos requeridos para la gestión del sistema.

Un Middleware Reflexivo es una capa intermedia entre diversas aplicaciones y/o servicios, donde, a través de la auto-conciencia y auto-referencia, una aplicación puede cambiar su comportamiento según los requerimientos y necesidades del entorno de ejecución. Consta de dos procesos [95]: La *introspección*, que es la habilidad de observar y razonar sobre su propia ejecución; y la *Intersección*, que permite la modificación de su propio estado o estructura, como mecanismo de adaptación. Para su implementación se deben considerar dos niveles:

Nivel Base: En el nivel base se encuentran las aplicaciones, donde se ejecutan las funcionalidades propias de las aplicaciones, y los servicios que las

conforman. En este nivel es donde se realiza el proceso de intersección, para realizar los ajustes necesarios para modificar el estado de ejecución (estructura), o alterar la interpretación o significado de los datos (comportamiento).

Nivel Meta: En el nivel meta es donde el middleware tiene su capacidad de reflexión, para observar y razonar acerca de los estados de ejecución de las aplicaciones, y determinar cómo adaptar sus estructuras y comportamientos a las necesidades del entorno. La reflexión que se da en este nivel, le confiere al middleware la capacidad de desarrollar sistemas computacionales que pueden ser sensibles a su ambiente.

El escenario que se muestra en este caso de estudio se desarrolla en un Aml educativo, específicamente, un Salón de Clases Inteligente (SaCI) [96]. El SaCI ha sido modelado usando el paradigma multiagente, y ha sido caracterizado usando la arquitectura AmICL [97] (Middleware que caracteriza un Aml de aprendizaje basado en C-Learning²²) para desplegarse. AmICL propone dos tipos de agentes, uno para caracterizar los componentes de software, y el otro para definir los componentes de hardware, en el ambiente. Estos agentes se distribuyen en las siguientes capas:

- *Capa de Aml Físico (CAF):* En esta capa se caracterizan los diversos dispositivos presentes en el SaCI (inteligentes o no) como agentes. Ejemplos de posibles agentes en CAF son Pizarra Inteligentes, Robots, etc.
- *Capa de Aml Lógico (CAL):* En esta capa se caracterizan los diversos componentes de software en SaCI como agentes. Ejemplos de posibles agentes es el sistema recomendador de contenidos digitales educativos, o el sistema académico.

También están las capas de Gestión de AmICL:

- *La Capa de Gestión del SMA (CSMA):* está compuesta por una comunidad de agentes que soporta la ejecución de aplicaciones multi-agentes.
- *La Capa de Gestión de Servicios (CGS):* es donde se registran, descubren, invocan, etc., los servicios requeridos por el ambiente en un momento determinado.

Específicamente se propone la emergencia ontológica basada en el análisis de contexto como un servicio en Aml. Para ello, se integra el AmICL con servicios de Consciencia de Contexto de CARMICLOC [98] (Middleware orientado a servicios que se encargan de observar el contexto y modelarlo), enriqueciendo el modelo semántico de AmICL a través de una base de conocimiento ontológica que permite a los agentes de AmICL adaptar sus servicios a los requerimientos

²² Ambientes de aprendizaje electrónico basados en la Nube [97]

del momento. Particularmente, esa extensión de AmICL permite hacer emerger modelos ontológicos adecuados al contexto, a través de un conjunto de reglas definidas y el uso de meta-ontologías.

El uso del MiR-EO, en este caso se propone para la generación de Ontología Emergente que permitan responder a los requerimientos emergentes del SaCI. Para ello, se deben implementar los componentes ofrecidos por nuestra arquitectura para crear servicios que permiten procesar las diferentes fuentes de información en el AmI. Esto permite obtener modelos de conocimientos (ontologías) adaptables a las necesidades y a la dinámica del AmI, que serán usados en la implementación de los servicios y dispositivos que en ella subyacen.

Arquitectura del MiR-EO

La arquitectura propuesta (Fig. 4.11) tiene por objetivo la generación de nuevas ontologías, a partir de la recolección y procesamiento de modelos de conocimiento heterogéneos presentes en el AmI, para ponerlas a disposición como un servicio. Este conocimiento, que debe registrarse en el middleware, se integra y procesa en la arquitectura, para ofrecer nuevos modelos conceptuales a través de Ontologías Emergentes, que estarán disponibles para su uso en el AmI.

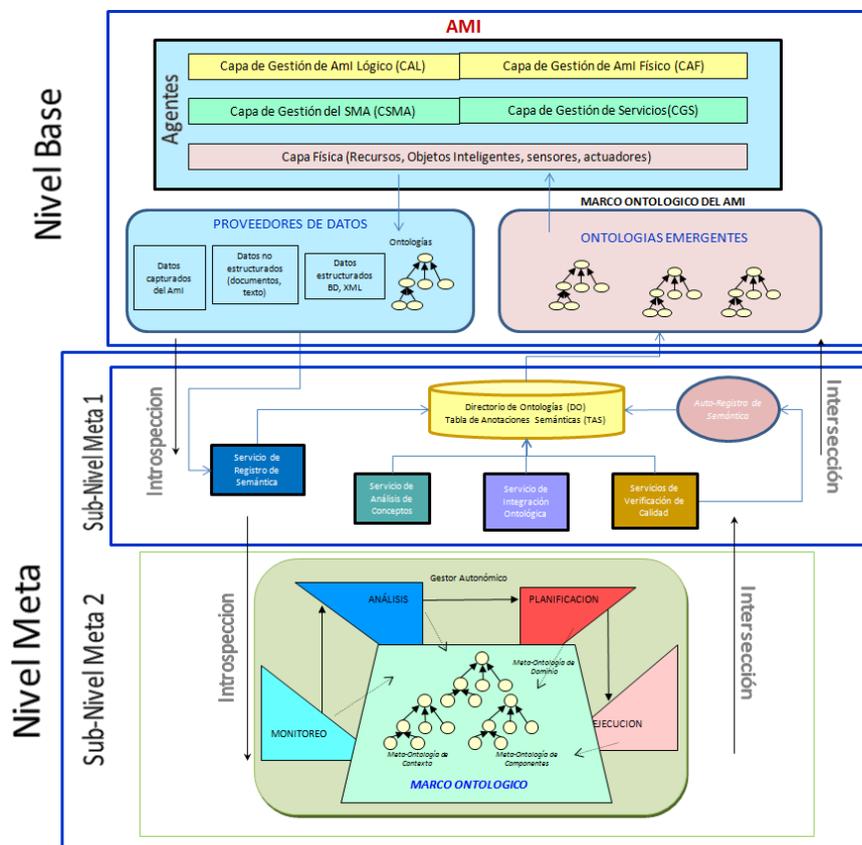


Figura 4.11. Middleware Reflexivo para Servicios de Ontologías Emergentes (MiR-EO).

La propiedad de reflexión del MiR-EO, le da la posibilidad de monitorear, tanto los eventos semántico (nuevos modelos conceptuales, nuevos conceptos y propiedades), como los servicios que dan soporte al Aml, para adaptarlos y adecuarlos a nuevos requerimientos. Para ello, usa su propio marco ontológico, basado en meta-ontologías, que le permite realizar ajustes y auto-gestionar el proceso evolutivo del marco ontológico del Aml. Se está hablando entonces de dos marcos ontológicos: uno que contiene todo el modelo conceptual del Aml, y que se gestiona con MiR-EO, y otro interno del middleware, formado por meta-ontologías, que permiten definir de forma genérica, los conceptos y las reglas que rigen las ontologías del Aml, para apoyar el proceso evolutivo de esas ontologías. En la Fig. 4.12 se puede observar el uso de los dos marcos ontológicos en el gestor autónomo.

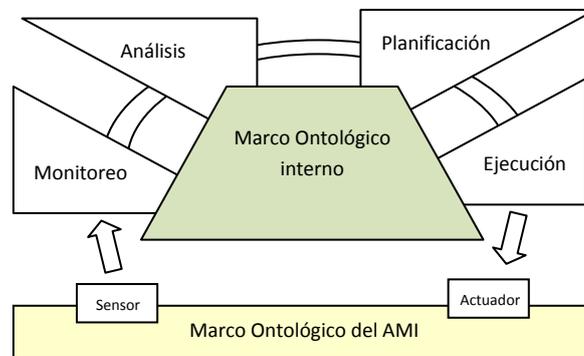


Figura 4.12. Marcos Ontológicos en el Gestor Autónomo del MiR-EO en el Aml.

La arquitectura propuesta consta de los dos niveles propios de los middleware reflexivos:

- **Nivel Base:** En este nivel están todos los componentes del Aml, tanto los recursos (materiales, objetos, usuarios) como los dispositivos (sensores y actuadores). En este nivel se encuentran las aplicaciones y los agentes que ejecutan las funcionalidades del Aml. Se tiene un modelo conceptual de este nivel, que se define en el marco ontológico de MiR-EO, para la gestión del Aml. A través de los puntos de enlace (sensores y actuadores desplegados en el Aml), es posible gestionar los recursos del Aml (por ejemplo, las ontologías), para realizar los procesos adaptativos que requieran (tales como la inclusión de nuevos conceptos).
- **Nivel Meta:** En este nivel es donde el middleware tiene su capacidad de reflexión sobre el funcionamiento de sus servicios, y sobre el

comportamiento ontológico del Aml, para, entre otras cosas, razonar y adaptar su estructura a las necesidades conceptuales del Aml, y de ser necesario, realizar las adaptaciones que requieran las ontologías del Aml. Todo ello es realizado a través de sus servicios. Además, es el responsable del proceso de introspección del nivel base. Este nivel se divide en 2 sub-niveles meta:

- En el **Sub-Nivel Meta 1** se encuentran los diferentes servicios que implementan los Componentes del EEO (ver Capítulo 3, Sección 3.5), que de forma colaborativa contribuyen en la generación, actualización y enriquecimiento de las diferentes ontologías que conforman el marco ontológico del Aml. Además, es donde se da el proceso de emergencia ontológica del Aml. En este nivel es donde se realiza el proceso de monitoreo, a través del registro semántico, para detectar los eventos semánticos en el sistema, y a través del proceso de intersección, realizar las adaptaciones o ajustes necesarios del marco ontológico del Aml.

- En el **Sub-Nivel Meta 2** es donde se estructura el modelo **MAPE** del Gestor Autónomo del Middleware:

- **Monitoreo:** El monitoreo se encarga de detectar los eventos semánticos en tiempo real en el Aml, es decir, qué nuevos datos y modelos conceptuales están ingresando en el Aml, y determina la necesidad de registrarlos. Se implementa principalmente, a través del servicio de “Registro Semántico”, para generar la “Tabla de Anotaciones Semánticas” y así conocer cuál es el estado actual en el ambiente a nivel conceptual. Este proceso se apoya en las meta-ontologías de contexto y de componentes, que ofrecen información sobre las propiedades y características de los diferentes elementos del ambiente y posibles conceptos.
- **Análisis:** Se realiza un análisis de lo recogido por el monitor. Se invoca el servicio de “Análisis de Conceptos”, para analizar y clasificar los conceptos registrados en la “Base de Conocimientos Semánticos”, y de acuerdo a su relevancia y a sus propiedades, determinar posibles “Conceptos Emergentes”. Realiza también el diagnóstico para determinar posibles conceptos emergentes y nuevos meta-conceptos, así como la necesidad de generar una meta-ontología de dominio, en base al conjunto de ontologías de un mismo dominio registradas en el sistema.
- **Planificación:** Aquí se evalúan las acciones a seguir para integrar y organizar los conceptos emergentes obtenidos en el análisis. Para ello, se integran los nuevos meta-conceptos y los conceptos

emergentes en la meta-ontología, basada en el cálculo de la entropía, para obtener la mejor organización posible.

- Ejecución: Finalmente, se realiza el proceso de integración y organización de la meta-ontología, para generar la “Ontología Emergente” con sus métricas de calidad. Esta ontología emergente podrá ser utilizada luego en el Aml.

Como se puede apreciar, en esta arquitectura existen un conjunto de componentes distribuidos entre el nivel base y el nivel meta, que trabajan en conjunto para lograr el modelo autónomo del middleware. El manejador autónomo y la base de conocimientos (marco ontológico compuesto por meta-ontologías) trabajan de manera conjunta, para realizar la gestión inteligente de sus servicios y de las ontologías del sistema. El marco ontológico, a través de sus meta-ontologías, provee la base de conocimientos necesaria para los procesos de detección de necesidades a nivel semántico, así como para el análisis y planificación de las acciones más adecuadas a realizar.

Marco Ontológico del Middleware

Aquí se encuentran las meta-ontologías, que constituyen la base de conocimientos a ser usada por los gestores autónomos del middleware, que permiten la gestión de las ontologías del Aml. Entre las meta-ontologías que se proponen manejar en este caso se tienen 2 básicas: la meta-ontología de componentes y la meta-ontología de contexto, que son predefinidas. La meta-ontología de dominio, en cambio, no puede predefinirse, ya que varía de acuerdo al dominio donde se desenvuelve el Aml, y puede ser definida inicialmente por expertos o generarse a partir de las ontologías que participan en el Aml. A continuación se describen las meta-ontologías propuestas.

Meta-Ontología de Componentes

Esta meta-ontología provee la estructura genérica de categorías y relaciones que describen los componentes, los usuarios, las entidades (objetos y dispositivos) que están en el ambiente, que interactúan entre sí, para ofrecer sus servicios o realizar sus actividades y funciones en el Aml. La Fig. 4.13 muestra las entidades de la meta-ontología de componentes del Aml, y sus relaciones. Entre los meta-conceptos que allí se observan se tienen:

- Recursos: objetos tangibles (sillas, mesas y computador, entre otros) que se encuentran en el Aml. Los recursos pueden ser objetos o contenedores (un contenedor puede tener dentro de sí varios objetos, por ejemplo, un estante o biblioteca).

- Espacios: los recursos están ubicados en diferentes espacios o lugares, que conforman el ambiente (salón, laboratorio y cocina, entre otros)
- Dispositivos: en los espacios del Aml hay diferentes dispositivos, que eventualmente tienen sus propios sensores y actuadores (cámaras, teléfonos y pizarras inteligentes, entre otros).
- Servicios: Son las diferentes funcionalidades que ofrecen los dispositivos, que pueden ser usados por los usuarios, recursos u otros dispositivos, en el ambiente.
- Usuarios: son los que hacen uso del Aml, de sus recursos, espacios y dispositivos, para la realización de sus actividades, por ejemplo: alumnos, profesores y médicos, entre otros.

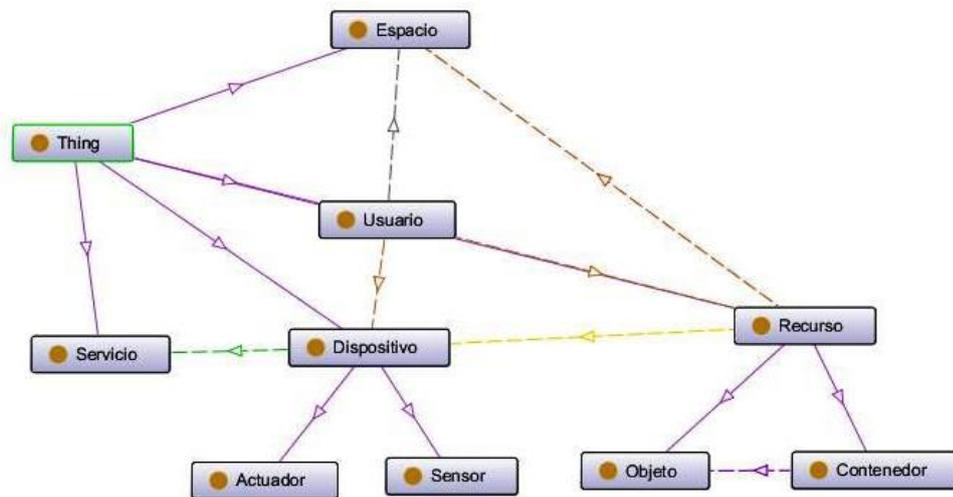


Figura 4.13. Meta-Ontología de Componentes

Meta-Ontología de Contexto

Esta meta-ontología es una extensión de la meta-ontología de componentes, y contiene el modelo conceptual para definir la situación o estado en que se encuentran los componentes del Aml: localización, tiempo, condiciones ambientales, actividades y perfiles, entre otros. La información del contexto es fundamental, ya que caracteriza a todos los elementos del Aml (definidos en la ontología de componentes), y definen los servicios que deben activarse en el Aml, de acuerdo a la condición de sus componentes en un momento dado. La información de contexto es determinante también, para la ubicación de nuevos conceptos en una ontología, ya que, por ejemplo, a través de medidas de similitud entre los objetos, en base a sus propiedades y comportamiento, se podrá ubicar en la taxonomía de una ontología de dominio a

un nuevo concepto. La Fig. 4.14 muestra los elementos de la meta-ontología de contexto del Aml, y sus relaciones, usadas en este trabajo. Ahora bien, en la literatura existe un buen número de ontología de contexto que podrían ser usadas por nuestro middleware, como la propuesta en [98].

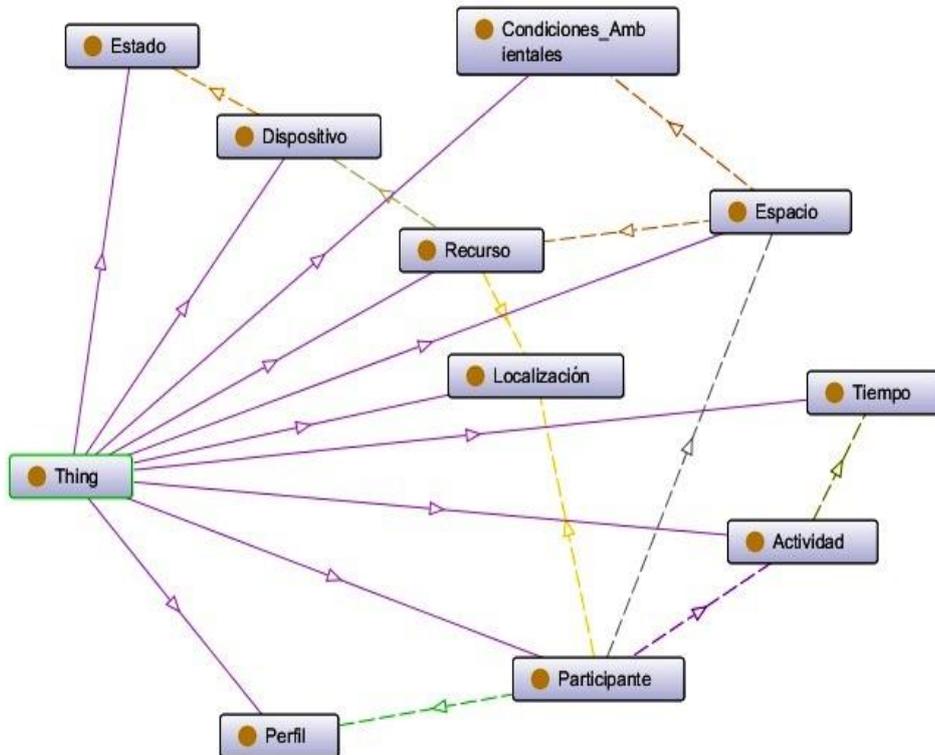


Figura 4.14. Meta-Ontología de Contexto

Entre los meta-conceptos que se definen en la meta-ontología de contexto, además de los componentes del Aml que también forman parte del contexto (definidos en la meta-ontología de componentes), se tienen:

- Estado: indica el estado en que se encuentra un dispositivo: encendido, apagado, activo, inactivo, en pausa; así como sus propiedades: color, tamaño, temperatura.
- Perfil: indica las características de los usuarios o participantes: estado, hábitos, necesidades, preferencias, rol.
- Condiciones Ambientales: define características del entorno: iluminación, temperatura, humedad, ruido, presión.
- Actividad: caracteriza las actividades que realizan los participantes: su duración, recursos usados, resultados.
- Tiempo: determina el momento en que tiene lugar un determinado contexto, indicando día, hora y lapso.

- Localización: determina la localización de los usuarios y de los recursos en el ambiente. Puede ser, por ejemplo, una localización relativa o absoluta (latitud, longitud).

Meta-Ontologías de Dominio

En cuanto a las meta-ontologías de dominio de un Aml, estas no pueden ser predefinidas para todos los Aml, ya que dependen de las funcionalidades y objetivos de ellos (por ejemplo, un Aml educativo o médico). Estas meta-ontologías pueden ser proporcionadas por fuentes externas (expertos del dominio), o generarse en las iteraciones del proceso de emergencia ontológica.

Las Meta-Ontologías son las que de alguna manera orquestrarán todo el conocimiento generado, para mantener la integridad y la coherencia entre los conceptos.

La relación entre las ontologías de componentes, de contexto y de dominio es determinante para los servicios del Aml, ya que la ontología de contexto modela el estado en que se encuentran los componentes del ambiente, y a su vez, los componentes en el ambiente están asociados a algún concepto del dominio que caracteriza el Aml. Por ejemplo, el contexto de un usuario en un momento dado viene dado por su perfil, su localización y sus preferencias, lo que determina los servicios que se pueden activar para ese usuario en ese momento.

Ejemplo de uso del MiR-EO

Para realizar una ejemplificación del uso del MiR-EO, se realizan simulaciones con datos artificiales, ya que no se tienen disponibles datos reales de un SaCl. En este caso, se tomaron como eventos semánticos el registro que podrían generar los agentes que controlan algunos dispositivos del ambiente, así como el ingreso de los usuarios al SaCl. Se tomó sólo una pequeña muestra de datos, usando la herramienta implementada para mostrar la ejecución del EEO (ver al inicio del capítulo). Sin embargo, es importante resaltar que en este caso, los componentes del esquema propuesto se deben implementar como servicios en el Aml, de manera de captar el comportamiento del sistema, y ofrecer las OEs resultantes para ser usadas por los diferentes servicios del Aml. En el caso de los ingresos de los usuarios, los registros son suministrados por el Agente Dispositivo (AD) que controla la entrada al salón (por ejemplo, un dispositivo de lectura de código de barra que autoriza o no el acceso a los usuarios). Luego de obtener a través del dispositivo de identificación el código del usuario, se recupera información a través de un servicio de búsqueda de personas pertenecientes a la

institución para poder dar acceso al lugar y obtener información del usuario como sus datos personales y tipo de usuario (estos servicios se implementan apoyados en las meta-ontologías de contexto y de componentes). Igualmente existen AD para los diferentes recursos y dispositivo presentes en el ambiente, que suministran información sobre sus propiedades. La meta-ontología de contexto ofrece un modelo conceptual sobre propiedades de usuarios, recursos y dispositivos, como: perfil, localización y estado, que se registran para aportar información adicional a los conceptos, y así poder agruparlos para conformar los MC. Igualmente, este agente requiere conocer la información de contexto sobre: cuál actividad corresponde a ese ambiente para ese momento, cuáles son los usuarios autorizados a entrar y cuáles dispositivos deben estar presentes. Parte de los conceptos que aportan estos eventos semánticos, luego del proceso de “Búsqueda de CC”, se muestran en la Tabla 4.4.

TABLA 4.4. CONCEPTOS CANDIDATOS DEL AMI

Fuente	Conceptos	Total Datos	Propiedades
RegistroEntrada.txt	Alumno	40	Codigo, Nombre, Apellido, Perfil, Indice Academico, Materias cursa
RegistroEntrada.txt	Profesor Ordinario	3	Codigo, Nombre, Apellido, Perfil, Categoría, Departamento, Asignaturas Administra
RegistroEntrada.txt	Auxiliar Docente	2	Codigo, Nombre, Apellido, Perfil, Categoría, Departamento, Asignaturas Administra
RegistroEntrada.txt	Profesor invitado	1	Codigo, Nombre, Apellido, Perfil, Categoría, Departamento, Asignaturas Administra
RegistroEntrada.txt	Personal Administrativo	4	Codigo, Nombre, Apellido, Perfil, Unidad Dependencia
DispositivosSalida.txt	Pantallas Informativa	10	Codigo, Nombre, localización, TipoSalida, Estado
DispositivosSalida.txt	Pizarra Interactiva	2	Codigo, Nombre, localización, TipoSalida, Estado
DispositivosSalida.txt	Tablet	20	Codigo, Nombre, localización, TipoSalida, Estado
DispositivosSalida.txt	Laptop	25	Codigo, Nombre, localización, TipoSalida, Estado
DispositivosSalida.txt	SmartPhone	35	Codigo, Nombre, localización, TipoSalida, Estado

Para aplicar el EEO, los valores para los umbrales que se establecieron en:

- UMR: 0.2
- URR: 0.5
- UME: 1

Estos valores se establecieron así, luego de verificar que, al tener todos los conceptos frecuencia de 1, los valores de relevancia estaban entre 0.5 y 1.

Luego de todo el proceso, en la Fig. 4.15 se puede observar la OE, con sus métricas de calidad: Completitud=1, Robustez=0.67 y precisión=1. En este caso, la ontología resultante cubre en un 100% los conceptos de la “Base de Conocimiento Semántica”. En cuanto a la robustez se observa que los “Conceptos Emergentes” son en promedio 67% relevantes. La precisión se ubica

en 100 %, es decir, en la ontología emergente todos los conceptos están agrupados adecuadamente.

En un Aml, la “Ontología Emergente” puede ser empleada para configurar los diferentes servicios que se ofrecen, y registrar información general sobre todos los eventos del sistema, y en conjunto con la meta-ontología de componentes y de contexto, caracterizar todos los componentes del Aml. Se debe resaltar que con este proceso de emergencia ontológica, se puede obtener ontologías que reflejen la realidad conceptual que se maneja en el Aml en un momento dado, que pueden ser actualizadas en el proceso iterativo del esquema, con nuevos eventos semánticos que surjan. Esto ocurre así, ya que la dinámica de este tipo de sistemas, que están en constante evolución, no permite tener ontologías estáticas predefinidas, sino que es necesario adaptarlas permanentemente a los nuevos requerimientos.

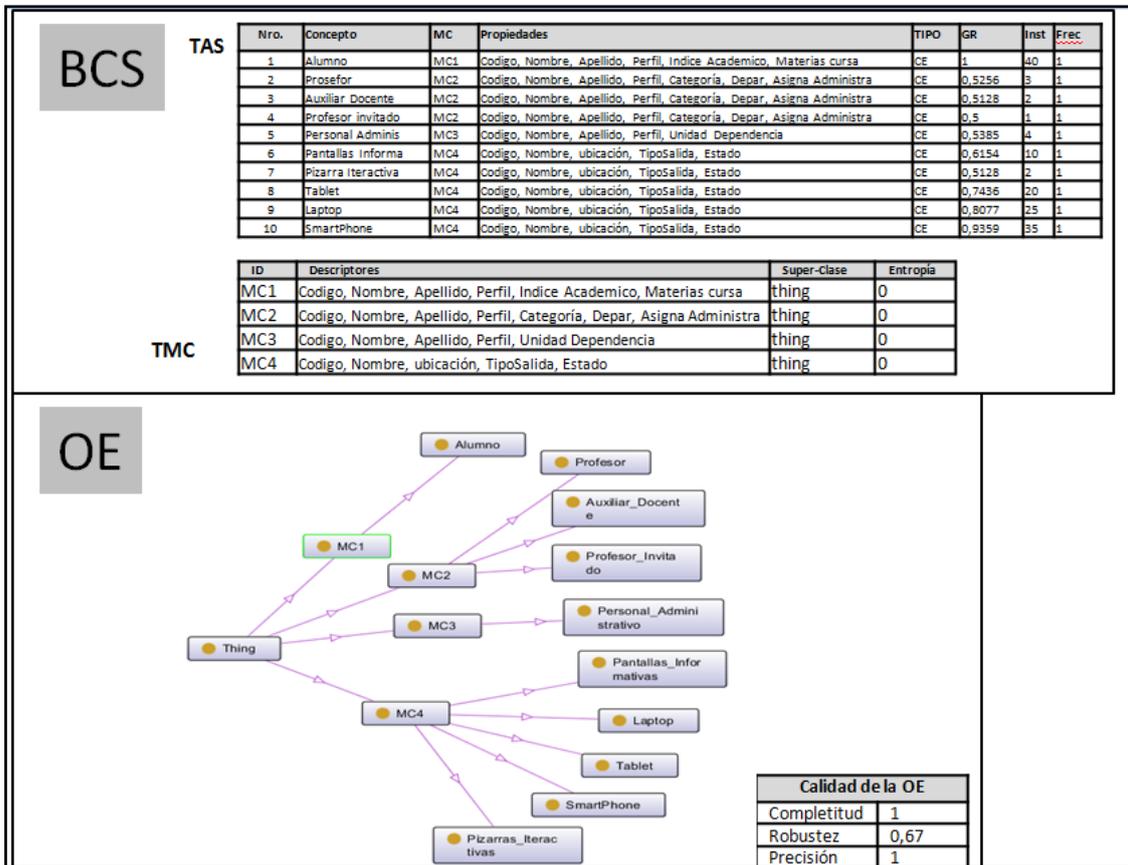


Figura 4.15. OE del Aml.

4.3. Aplicación del EEO en Ciudades Inteligentes

En este caso de estudio, se propone una adaptación conceptual del EEO para su uso en ciudades inteligentes, con el objetivo de generar ontologías emergentes que reflejen la realidad y la dinámica de los diferentes conceptos que se manejan en la implementación de los servicios que se ofrecen en la ciudad.

Contextualización

Las ciudades inteligentes (*Smart Cities*) son entornos inteligentes a gran escala, con grandes cantidades de datos que se utilizan para mejorar la calidad de vida de sus ciudadanos. Las ciudades inteligentes buscan mejorar el tráfico, la salud, el Gobierno, la sociabilidad, entre otros aspectos, de una ciudad, a través de sistemas inteligentes [99]. Uno de los principales problemas en el desarrollo de una ciudad inteligente es describir la enorme cantidad y las múltiples fuentes de información, y el uso de datos de diferentes dominios.

En este caso de estudio, se muestra la creación de un middleware reflexivo para Smart Cities (MiSCi) [100], que desde la visión de entornos inteligentes modelados como agentes inteligentes, pueden adaptarse al dinamismo existente en una ciudad, utilizando ontologías emergentes que no solo se adaptan al contexto del momento, sino que también pueden responder a situaciones imprevistas. Esta arquitectura reflexiva autónoma se definió como un sistema multiagente (MAS), que tiene agentes para caracterizar tanto a los individuos, dispositivos y aplicaciones (que brindan servicios a particulares), que conviven en la ciudad inteligente. Su arquitectura se basa en servicios web, permitiendo que sus servicios sean consumidos por las aplicaciones (en nuestro caso, Agentes), conscientes del contexto o no. Los agentes pueden crear ontologías emergentes temporales o permanentes, que permiten resolver una situación particular, según el contexto.

La arquitectura de MiSCi será la base para tomar decisiones inteligentes conscientes del contexto en la ciudad inteligente (Fig. 4.16). MiSCi es capaz de detectar el entorno, razonar, planificar y ejecutar una solución, en una forma independiente, proactiva, consciente del contexto e inteligente.

Dentro de esta arquitectura se define una Capa de Emergencia Ontológica (CEO), donde se despliegan el conjunto de servicios que se ofrecen del EEO. Estos servicios son los que permitirán crear ontologías contextuales para enfrentar cada situación en la ciudad inteligente.

El propósito de esta capa es proporcionar un conjunto de servicios con tareas muy específicas, para hacer posible la generación de ontologías emergentes. Estas ontologías proporcionarán una visión adecuada del contexto de la ciudad inteligente. Los servicios de actualización automática de las ontologías permiten adaptarlas a la dinámica y evolución de su entorno. Esta capa permite monitorear las fuentes de datos y ontologías para adaptarlas a nuevos requerimientos, detectando conceptos emergentes que se pueden incluir en las ontologías para permitir así la evolución del marco ontológico de la ciudad inteligente.

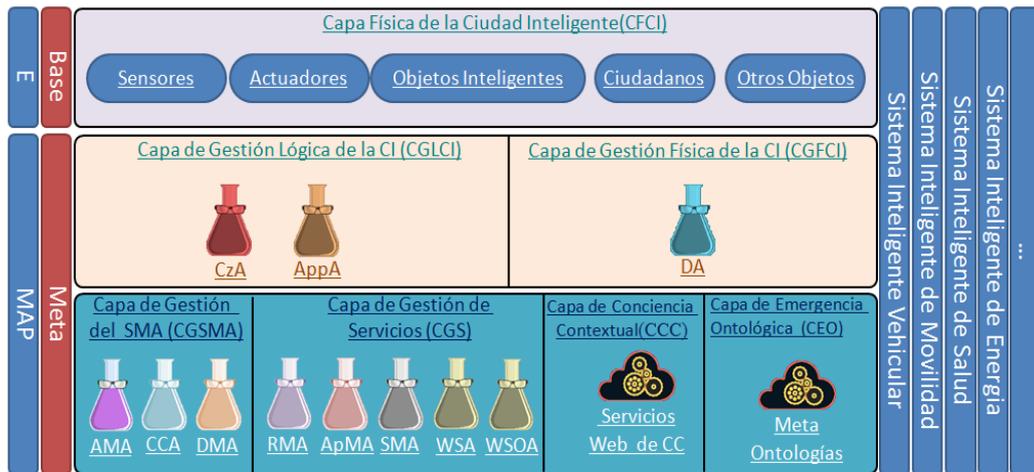


Figura 4.16. Arquitectura de MiSci.

El Marco Ontológico para las Ciudades Inteligentes.

El marco ontológico es compuesto por ontologías distribuidas, que son administradas por diferentes agentes y servicios dentro de la ciudad inteligente, que son heterogéneas (tienen, por ejemplo, diferencias en la identificación de sus conceptos y en sus estructuras), ya que cada fuente de conocimiento tiene sus propios diseñadores, y por lo general, tienen diferentes diseños y estructuras.

El marco ontológico de una ciudad inteligente debe poder representar los nuevos objetos y comportamientos en la ciudad, por lo tanto, aunque algunos elementos pueden ser predefinidos, otros necesariamente deben estar "emergiendo". Las ontologías representan el modelo de conocimiento presente en la ciudad inteligente. El conocimiento puede estar disperso en diferentes fuentes de datos y dispositivos (bases de datos, documentos, datos monitoreados, dispositivos portátiles) y debe evolucionar de acuerdo con la dinámica del contexto.

Específicamente, la arquitectura puede manejar múltiples ontologías distribuidas en agentes de la ciudad inteligente, y en una meta-ontología central. Esta meta-ontología central es administrada por el OEL, que proporciona servicios ontológicos para permitir la emergencia ontológica.

La meta-ontología central se utiliza para definir el modelo general de los componentes y dominios presentes en la ciudad inteligente. Se muestra un ejemplo de esta meta-ontología en la Fig. 4.17.

Esta meta-ontología vincula y estructura las ontologías de dominio que pueden existir en los diferentes espacios, y organiza la información semántica a partir de diferentes fuentes de datos que surgen en la ciudad inteligente, por la incorporación de nuevos dispositivos, agentes o servicios. En este modelo ontológico se consideran datos del perfil del ciudadano, y las actividades o eventos que tienen lugar, teniendo en cuenta el contexto en el que se producen: el dominio espacial (como lugares y objetos) y el dominio funcional (como salud, economía y energía). En cada dominio, también puede existir meta-ontologías para definir genéricamente su modelo conceptual, manteniendo la integridad y consistencia entre los conceptos, mostrando información global sobre cómo deben ser estructurados y cómo se relacionan entre sí.

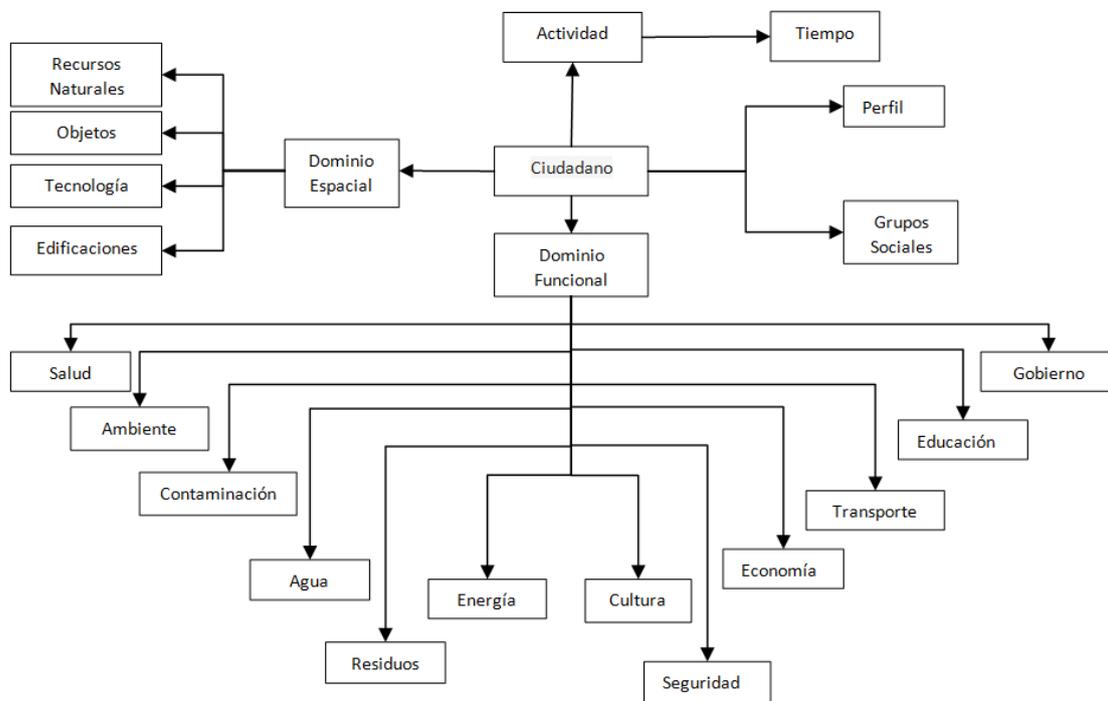


Figura 4.17. Meta-Ontología para una ciudad inteligente.

En la arquitectura propuesta, los agentes involucrados en la ciudad inteligente deben registrar (por medio del componente de Registro Semántico

del EEO) información sobre los recursos, componente y eventos del sistema, y sobre las ontologías que manejan, en la ciudad inteligente. En este proceso de registro semántico, se propone para este caso, que los MCs deben estar relacionados con algunos de los dominios definidos en la meta-ontología central (Fig. 4.17). De esta manera, para este caso, en la TMC (Def. 3.16) se agrega un dato donde se especifica el “Dominio” al que pertenece el MC. En la Tabla 4.5 se muestra, con un ejemplo, como queda la estructura de la TMC para el caso de ciudades inteligentes.

TABLA 4.5. TABLA DE META-CONCEPTOS EN CIUDADES INTELIGENTES

MC	Descriptorios	Super-clase	Entropía	Domino
MC2	Transita Por_Tierra, Tiene_Ruedas	MC1	0	Transporte
MC3	Transita Por_Aire, Tiene_Alas	MC1	0	Transporte

El campo “Dominio” permitirá organizar los MCs, y facilitar los procesos de búsqueda y organización de las meta-ontologías, y por supuesto, de las diferentes ontologías emergentes que vayan surgiendo en la ciudad inteligente. Al momento de realizar el registro semántico, será necesario identificar a que dominio pertenece el evento semántico, para así poder seleccionar el MC del dominio que le corresponda. Esto es importante, ya que los conceptos se podrán asociar a los meta-conceptos que le corresponden realmente de acuerdo a su dominio, de una manera más rápida, reduciendo los tiempos de búsqueda, y mejorando la precisión de la OE resultante, considerando que en una ciudad inteligente se manejan una gran cantidad de conceptos de dominios diferentes, que incluso pueden tener propiedades similares.

Extensión de la Capa de Emergencia Ontológica considerando la Computación en la Niebla (Fog Computing)

La “Computación en la Niebla” (*Fog Computing*, en inglés) es una extensión del paradigma de “Computación en la Nube”²³ (*Cloud Computing*, en inglés). Es un modelo donde el procesamiento de aplicaciones se concentran en dispositivos en el borde de la red, en lugar de estar completamente en la nube, de modo que los datos se procesan localmente en un dispositivo inteligente, en lugar de enviarse a la nube [101]. La “Computación en la Niebla” requiere una nueva generación de aplicaciones y servicios, con una interacción fructífera entre la Nube y la Niebla, en particular, cuando se trata de análisis y gestión de datos [102].

El despliegue de las plataformas de computación en la ciudad inteligente, se apoya en la nube al hacer uso de un grupo de computadoras y servidores

²³ La computación en la nube es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet [103]

conectados entre sí a través de Internet, formando una red para el almacenamiento de grandes cantidades de datos y servicios [103]. En estos sistemas se genera un volumen y variedad de datos sin precedentes, por lo tanto, se requiere un despliegue eficiente del middleware para evitar que cuando los datos lleguen a la nube para su análisis, la oportunidad de actuar en la ciudad inteligente se haya ido. En una ciudad inteligente, donde existen múltiples nodos y aplicaciones que interactúan para ofrecer servicios a los ciudadanos, la alta calidad y la baja latencia²⁴ son de gran importancia. La implementación de los middleware clásicos para ciudades inteligentes basado en un paradigma de computación en la nube, puede convertirse en un problema para las aplicaciones que requieren respuesta en tiempo real, soporte de movilidad y/o distribución geográfica [104]. La computación en la niebla extiende la nube para estar más cerca de las cosas que se producen y actúan en la ciudad inteligente, y así dar respuestas más rápidas [102].

En [105] se propone extender MisCi utilizando el paradigma de la computación en la niebla, para dar soporte a esta situación. La capa de niebla es una característica clave de la arquitectura ampliada de MiSCi, para implementarlo y cumplir con los requisitos de rendimiento, calidad de servicio y baja latencia, para el procesamiento de datos y atención de requerimientos de los usuarios en una ciudad inteligente. En particular, un agente en MisCi se encarga de decidir qué información debe procesarse localmente o en la nube. Para tomar esta decisión, se apoya en una ontología descrita en la Fig. 4.18. En esta meta-ontología, llamada “Ontología de Niebla” (*Fog ontology* en inglés), se asigna una ponderación a cada elemento de la meta-ontología, por lo que el agente puede saber qué elementos tienen alta prioridad para ser procesados localmente.

La Ontología de Niebla será entonces la meta-ontología del EEO, donde cada meta-concepto de la meta-ontología general de la ciudad inteligente (que representa un dominio) tendrá asociado un atributo “prioridad”, que representa el peso de cada meta-concepto, que se calcula en base a la relevancia de los CEs que estén asociados a los MCs del dominio correspondiente. Esto implica que los dominios aumentan su peso de acuerdo al número y relevancia de los CEs que surge en la OE, lo que nos da un indicador de los dominios de mayor prioridad en una ciudad inteligente, y donde los nodos de niebla pueden ser más útiles. En la Fig. 4.6 se observa como quedaría la tabla de meta-conceptos. El cálculo de la

²⁴ En informática la **latencia** es la suma de retardos temporales dentro de una red. Un retardo es producido por la demora en la propagación y transmisión de paquetes dentro de la red [101]

prioridad se realizará en el proceso de Integración Ontológica (Macro-Algoritmo 3.6). Al momento en que algún concepto entra o sale de la ontología emergente, se actualizaría la Prioridad (Pr) del meta-concepto asociado, realizando el cálculo de la siguiente manera:

$$Pr(MC) = \frac{\sum_{i=1}^{TM} GR(CE_i)}{TM}$$

Donde:

- GR: Grado de Relevancia del CE asociado al MC.
- TM: Total de CE asociados al Meta-Concepto.

Por ejemplo si el MC2 tiene 5 CE asociados con GR de 0.5, 0.7, 0.6 y 0.7 la Pr(MC2) es 0.6. Y en el caso de MC3 si tiene por ejemplo 2 CE asociados con GR de 0.4 y 0.5 la Pr(MC3) es 0.5. En este caso en promedio el dominio “Transporte” tiene una Prioridad de 0.6.

TABLA 4.6. TABLA DE META-CONCEPTOS EN LA ONTOLOGÍA DE NIEBLA

MC	Descriptores	Super-clase	Entropía	Domino	Prioridad
MC2	Transita Por_Tierra, Tiene_Ruedas	MC1	0	Transporte	0.6
MC3	Transita Por_Aire, Tiene_Alitas	MC1	0	Transporte	0.5

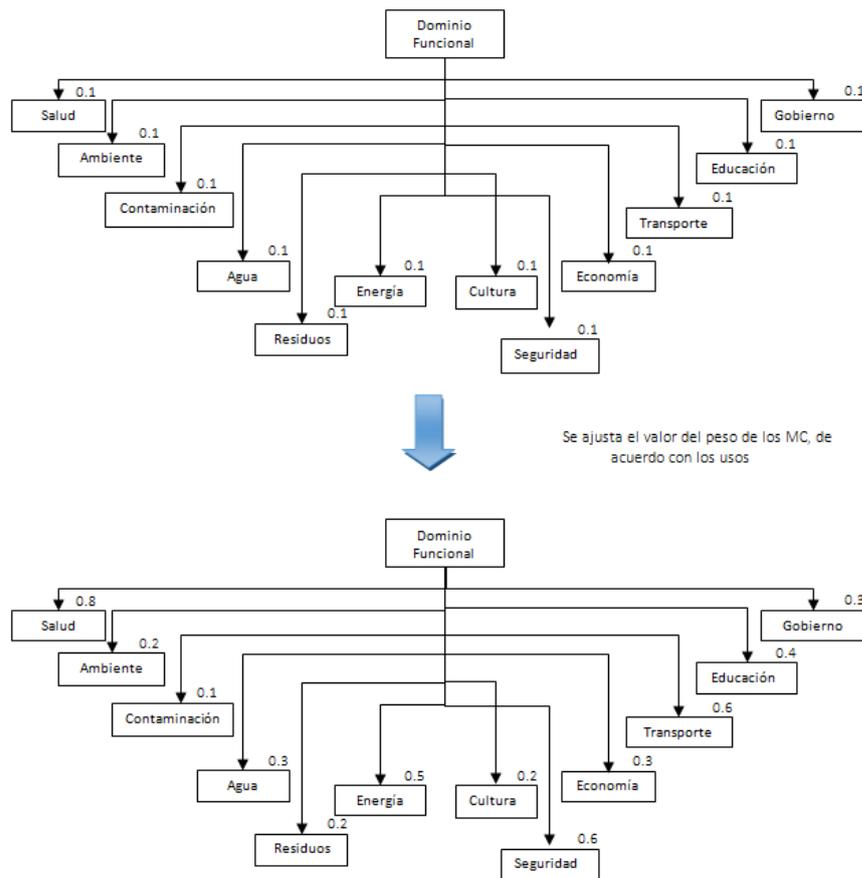


Figura 4.18. Ontología de Niebla

4.4. Análisis de Resultados

En el primer caso de estudio se muestran diferentes escenarios, donde se usaron datos y ontologías, para visualizar las variaciones que puede presentar la OE. Allí se muestra el impacto que tienen los criterios de relevancia y la entropía, así como los umbrales definidos al inicio en la OE. Es importante resaltar que el orden en que se procesen los ES no serán determinantes en el resultado. Los parámetros de calidad de la ontología resultante nos muestran que tan completa, robusta y precisa es la OE, con respecto a esa BCS. Una completitud alta nos indica que la OE tiene una gran cobertura de la BCS. Una OE robusta nos indica que sus CEs tienen en general una alta relevancia. Finalmente, una OE de alta precisión nos indica que sus CEs se han agrupado de forma adecuada, es decir, tienen una baja entropía.

En el segundo caso de estudio al aplicar del EEO en un Aml educativo, específicamente, en un Salón de Clases Inteligente, se presenta un Middleware Reflexivo para la Emergencia Ontológica (MiR-EO). Este Middleware, como se muestra en el trabajo presentado en [88] es otro aporte de este trabajo. Esta arquitectura permite desplegar los diferentes componentes del EEO, para de forma autónoma, mantener un marco ontológico actualizado de acuerdo a las necesidades del Aml, monitoreando constantemente los eventos semánticos que surgen en el ambiente, para ofrecer la OE como un servicio.

En el tercer caso de estudio se realiza una propuesta conceptual de adaptación y aplicación del EEO en la implantación de ciudades inteligentes. Se muestra un Middleware reflexivo para Ciudades Inteligentes (MiSCi), donde se despliegan un conjunto de servicios que implementan los diferentes componentes del EEO. Este Middleware en una propuesta presentada en conjunto con otros trabajos doctorales, como se muestra en el trabajo presentado en [100]. El objetivo de estos servicios es crear OEs que puedan ofrecer una visión adecuada del contexto de la ciudad inteligente y que puedan ser usadas por las diferentes aplicaciones que lo requieran. En este caso de estudio, se define una meta-ontología central que usa el EEO, que representa de manera general la ciudad inteligente. Esa meta-ontología define algunos dominios específicos, de manera que se puedan agrupar los meta-conceptos y organizarlos para que sea más fácil su manejo al momento de generar una OE. Finalmente, como una extensión de esta arquitectura para su uso en *Fog-Computing*. Este trabajo se presentó en [105] y allí se propone incluir una ponderación a cada meta-concepto de la meta-ontología en base al número de conceptos emergente que surgen asociados a ese meta-concepto, para así determinar la prioridad de los conceptos, y saber cuáles dominios deben considerar los agentes para realizar procesamientos de manera local.

Conclusiones y Recomendaciones

En este trabajo se ha desarrollado un Esquema de Aprendizaje Ontológico (EEO) para Sistemas Auto-Organizados y Emergentes (SAOyE) cumpliendo los objetivos planteados al inicio de la investigación. En primer lugar, se han estudiado las metodologías, métodos, técnicas y herramientas para el aprendizaje ontológico disponibles actualmente. Igualmente, se estudiaron los aspectos teóricos relacionados a los sistemas auto-organizados y emergentes. Posteriormente, se definió un esquema de aprendizaje ontológico, considerando las características de los sistemas auto-organizados y emergentes. Finalmente, se verificó el esquema propuesto a través de su aplicación en casos de estudios.

En la estructuración del esquema propuesto, se definieron un conjunto de conceptos que se usaron en el desarrollo de la propuesta, así como un conjunto de tareas, mecanismos y procesos que se consideran necesarios para la emergencia ontológica. Se definió el concepto de “Emergencia Ontológica” como el proceso a través del cual surge una ontología, o se adapta una ontología existente a los cambios y nuevos componentes de un dominio dinámico, emergente y auto-organizado, a través de la aplicación de un conjunto de reglas. El esquema propuesto permite generar una Ontología Emergente (OE) a partir de diversas fuentes de datos (tablas, documentos, ontologías, etc.). Se definieron 4 componentes que conforman el esquema: “Componente de Registro Semántico”, “Componente de Análisis de Conceptos”, “Componente de Integración” y “Componente de Calidad”. Estos componentes trabajan en dos fases o modalidades: La Caracterización Semántica y la Estructuración Ontológica, y conforman los mecanismos de socialización, agregación y apropiación, propios de los sistemas emergentes, en este caso, de la emergencia ontológica. Estos componentes, junto con la Base de Conocimiento Semántica (BCS) y las Meta-Ontologías (MO), realizan un proceso de caracterización semántica, para generar espontáneamente nueva información ontológica que luego se estructura para obtener una ontología emergente. El proceso se realiza en base a una serie de iteraciones, donde por cada evento semántico que ingrese al sistema se ejecuta una iteración para obtener un resultado parcial, o sea, una OE, que puede crecer o reestructurarse en la siguiente iteración. Cada OE tiene asociada unas métricas de calidad para que pueda ser evaluada.

Entre los aspectos más importantes del esquema propuesto, se encuentran las métricas consideradas en el proceso: relevancia y entropía. El grado de relevancia se determina en función de la frecuencia de aparición de un concepto y

sus instancias asociadas, lo que permite tener un indicador que califica un concepto como emergente. La relevancia se considera determinante, ya que en un SAOyE, por su dinamismo, pueden existir en un momento dado una gran cantidad de conceptos participando, por lo que se requiere extraer o detectar aquellos conceptos predominantes, de mayor participación. Por otro lado, la entropía, que es un indicador de incertidumbre en los SAOyE, se utiliza para medir el desorden en la forma en que se organizan los conceptos. Estas dos métricas se utilizan en el proceso de generación de OE, ya que los conceptos se considerarán emergentes, en función de su grado de relevancia, y la agrupación y estructuración de los conceptos en los meta-conceptos en la OE se realiza en función de los niveles de entropía (reducir la entropía permite una mejor organización de los conceptos en la OE).

Los parámetros iniciales, como los umbrales de relevancia y de entropía, deben definirse al inicio del proceso, y serán claves en la calidad de una OE. La calidad de una OE se mide con los criterios de completitud, precisión y robustez. Si los umbrales son muy bajos, pueden generar un OE de mayor completitud, que incluya un mayor número de conceptos, pero de menor precisión. Los umbrales pueden ajustarse para generar resultados diferentes, considerando información que surge del proceso emergente, como la calidad de la ontología resultante.

Hay muchos trabajos que tratan la evolución de las ontologías y el aprendizaje ontológico [21] [22] [23] [24] [25] [26] [27], pero no hay estudios sobre la generación de una ontología emergente de forma dinámica, en base a fuentes de datos que surgen dinámicamente en el propio sistema. No se han encontrado trabajos que consideren múltiples fuentes de datos y modelos conceptuales para construir una ontología, en un sistema que podría ser muy dinámico y no predeterminado. El proceso de emergencia ontológica propuesto es automático y espontáneo, con una participación reducida del usuario, utilizando como base las métricas de relevancia y entropía. Todo esto se verifica en casos de estudios, donde se ha observado que se pueden obtener varias OEs en base a diferentes umbrales definidos para las métricas.

Otro aspecto importante de la propuesta es la generación de meta-ontologías de forma automática. Una meta-ontología contiene información genérica sobre el dominio, y puede ser instanciada para generar una nueva ontología. En el proceso automático para la generación de meta-ontologías, se establecen categorías y sub-categorías a partir de un conjunto de conceptos con propiedades comunes, para conformar meta-conceptos. La "Teoría de Categorías" fue la base teórica usada para el diseño del proceso de generación de meta-conceptos, y en la estructuración de una meta-ontología. De acuerdo a esta teoría, se pueden definir categorías agrupando datos que poseen propiedades comunes, y a su vez, se pueden establecer relaciones entre ellas. Basados en esas capacidades de la teoría de categorías, se presentaron un conjunto de definiciones, a partir de las cuales se definió el proceso de generación de meta-ontologías.

Con el propósito de verificar y mostrar la utilidad del EEO propuesto, se aplicó la propuesta en 3 casos de estudios. Un primer caso de estudio consiste en fuentes

de datos conformadas por un conjunto de datos obtenidos de portales de Open Data u ontologías abiertas. En un segundo caso se simula un Aml, y el último caso de estudio se orientó a una aplicación sobre ciudades inteligentes. Para ejecutar el EEO, se desarrolló una aplicación que ejecuta las iteraciones, configurando previamente los umbrales y los datos de entrada. La aplicación genera la OE de cada iteración, con sus valores de calidad asociados. Los valores de calidad de las OEs generadas por nuestra propuesta, evalúan la completitud, relevancia y precisión de la ontología resultante, con respecto al conjunto de conceptos que se quiere representar que están en la BCS.

En el primer caso de estudio se muestran diferentes escenarios, donde se usaron datos y ontologías, para visualizar las variaciones que puede presentar la OE. Primeramente, se muestra detalladamente cómo se ejecuta paso a paso cada iteración en el proceso. Luego, en cada escenario se muestra la OE final con sus parámetros de calidad, con respecto a la BCS. Estos parámetros de calidad nos muestran que tan completa, robusta y precisa es la OE, con respecto a esa BCS.

En el segundo caso de estudio se propone la aplicación del EEO en un Aml educativo, específicamente, en un Salón de Clases Inteligente, a través de un Middleware Reflexivo para la Emergencia Ontológica (MiR-EO). Este Middleware representa otro aporte importante de este trabajo. Esta arquitectura inspirada en el paradigma de computación autónoma, posee un nivel base donde se ejecutan las aplicaciones del Aml, y un nivel meta donde se desarrolla la emergencia ontológica, ejecutando el modelo MAPE. Toda esta arquitectura propuesta permite desplegar los diferentes componentes del EEO, para de forma autónoma, mantener un marco ontológico actualizado de acuerdo a las necesidades del Aml, monitoreando constantemente los eventos semánticos que surgen en el ambiente, para ofrecer la OE como un servicio.

En el tercer caso de estudio se muestra un Middleware reflexivo para Ciudades Inteligentes (MiSCi), donde se despliegan un conjunto de servicios que implementan los diferentes componentes del EEO. Allí se define una meta-ontología central que usa el EEO, que representa de manera general la ciudad inteligente. Esa meta-ontología define algunos dominios específicos, de manera que se puedan agrupar los meta-conceptos y organizarlos para que sea más fácil su manejo al momento de generar una OE. Finalmente, como una extensión de esta arquitectura para su uso en *Fog-Computing*, se propone incluir una ponderación a cada meta-concepto de la meta-ontología en base al número de conceptos emergente que surgen asociados a ese meta-concepto, para así determinar la prioridad de los conceptos, y saber cuáles dominios deben considerar los agentes para realizar procesamientos de manera local.

El EEO representa una propuesta novedosa en el ámbito del aprendizaje ontológico para sistemas dinámicos, donde se requiere una actualización constante de sus ontologías, en la misma medida que los sistemas van cambiando, para así mantener modelos conceptuales consistentes de los componentes cambiantes del sistema que representan. Su importancia es mayor, si consideramos que muchos

sistemas, para poder satisfacer las demandas del mundo actual en constante avance y evolución, necesariamente requieren trabajar de forma auto-organizada y emergente. En general, los diferentes aspectos considerados en la investigación son muy relevantes, lo que se evidencia con la publicación de diversos artículos (ver Anexo C).

En trabajos futuros, es importante continuar evaluando criterios adicionales que puedan complementar el grado de relevancia, a fin de determinar con mayor precisión los conceptos emergentes, como por ejemplo, considerar el factor de impacto de un concepto, ya sea en base a los datos asociados, a información de contexto, o a criterio de expertos. Otro aspecto importante es establecer mecanismos para determinar automáticamente los umbrales de relevancia y entropía más apropiados, como por ejemplo, a través de algún proceso de retroalimentación, donde en base a los valores de calidad resultantes y a algunos criterios definidos (p. ej. valores máximos y mínimos aceptables en las métricas de calidad, dependiendo del dominio), los umbrales puedan ajustarse. Estas métricas se definen al principio del proceso, pero son claves en la EO resultante y en sus niveles de calidad.

También, como trabajo futuro se considera importante probar el EEO propuesto en escenarios reales de Aml y Ciudades Inteligentes. En este trabajo se realizaron propuestas importantes en estos ámbitos, en base a datos simulados, con la siguiente limitación: se usaron pocos datos, una pequeña muestra, no muy diversa, entre otras cosas. En escenarios reales, con la gran cantidad de datos diversos, de diferentes situaciones que se suscitan, seguramente van a generar una mayor cantidad de meta-conceptos, que pueden enriquecer aún más la OE. La aplicación del EEO en estos escenarios permitirá afinar sus componentes, y adecuarlos a requerimientos propios que pueden surgir en este tipo de sistemas, como el pre-tratamiento de los datos específicos del entorno, la definición de las características propias del contexto usando el EEO para brindar servicios de conciencia del contexto, entre otras cosas. En el caso de Ciudades Inteligentes, se podrían organizar las meta-ontologías de dominio y las OEs en diferentes marcos ontológicos, de acuerdo a los diferentes dominios que se manejen (espacial, funcional, etc.).

Finalmente, se recomienda el uso del EEO como una herramienta de apoyo en las implementaciones y desarrollo de aplicaciones donde las ontologías son determinantes (por ejemplo, las aplicaciones vinculadas al área de la Web Semántica). Particularmente, los expertos en el desarrollo de estas aplicaciones y en el diseño de ontologías, pueden encontrar en nuestra propuesta una poderosa herramienta para organizar gran cantidad de conceptos dinámicos, descubrir nuevos modelos conceptuales a gran escala y de forma más rápida, para obtener así diferentes versiones de ontologías y meta-ontologías que pueden ser usadas casi de forma inmediata, y que a su vez, pueden irse refinando o complementando en el tiempo por el mismo proceso implícito en nuestra propuesta.

Referencias

- [1] T. Gruber, «A Translation Approach to Portable Ontology Specifications,» *Technical Report Knowledge Systems Laboratory, Stanford University*, 1993.
- [2] P. Borst, «Construction of Engineering Ontologies for Knowledge Sharing and Reuse,» *Tweente University*, p. Tesis Doctoral, 1997.
- [3] F. Zablith, G. Antoniou, M. d'Aquin, G. Flouris, H. Kondylakis, E. Motta y M. Sabou, «Ontology evolution: a process-centric survey,» *The Knowledge Engineering Review*, vol. 30, nº 1, pp. 45-75, 2015.
- [4] M. Barrera, H. Núñez y E. Ramos, «Ingeniería Ontológica,» *Lecturas en Ciencias de la Computación*, 2012.
- [5] F. García, «Web Semántica y Ontologías,» *Tendencias en el Desarrollo de Aplicaciones Web, Universidad de Salamanca*, pp. 1-23, 2006.
- [6] L. Zhou, «Ontology learning: state of the art and open issues,» *Information Technology and Management*, vol. 8, nº 3, pp. 241-252, 2007.
- [7] Q. Quboa y M. Saraee, «A State-of-the-Art Survey on Semantic Web Mining,» *Intelligent Information Management*, vol. 5, nº 1, pp. 10-17, 2013.
- [8] N. Perozo, J. Aguilar, O. Terán y H. Molina, «A Verification Method for MASOES,» *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 43, nº 1, pp. 64-76, 2013.
- [9] J. Aguilar, *Introducción a los Sistemas Emergentes, Talleres Gráficos, Universidad de Los Andes*, 2014.
- [10] X. Tao, Y. Li y R. Nayak, «A knowledge retrieval model using ontology mining and user profiling,» *Integrated Computer-Aided Engineering*, vol. 15, nº 4, pp. 313-329, 2008.
- [11] J. Lyons y L. John, *Linguistic semantics: An introduction*, Cambridge University Press, 1995.
- [12] P. Cudré-Mauroux, «Loose ontological coupling and the Social Semantic Web,» *Advances in Intelligent Web Mastering*, pp. 11-15, 2012.

- [13] K. Aberer, P. Cudré-Mauroux, A. Ouksel, T. Catarci, M. Hacid, A. Illarramendi y R. Studer, «Emergent semantics principles and issues,» *Database Systems for Advanced Applications*, pp. 25-38, 2004.
- [14] T. Rodríguez y J. Aguilar, «Ontological learning for a dynamic semantics ontological,» *DYNA*, vol. 81, nº 187, 2014.
- [15] Y. Wilson, «Learning Lightweight Ontologies from Text across Different Domains using the Web as Background Knowledge,» Thesis Doctoral, University of Western Australia., 2009.
- [16] G. Giraldo, G. Urrego y J. Marín, «Extracción de elementos de una ontología del dominio a partir de documentos tipo esquema,» *Avances en Sistemas e Informática*, vol. 6, nº 2, pp. 11-20, 2009.
- [17] H. De Souza, A. Carvalho y M. Cavalcanti, «Ontologias Emergentes: Uma Nova Abordagem para Integração de Ontologias,» de *Proc. 23rd Brazilian symposium on Databases*, 2008.
- [18] J. Tang, H. Leung, Q. Luo, D. Chen y J. Gong, «Towards Ontology Learning from Folksonomies,» de *Proc. Twenty-First International Joint Conference on Artificial Intelligence*, 2009.
- [19] G. Aranda, «Tratamiento de la semántica emergente mediante sistemas de agentes basados en conocimiento,» Tesis Doctoral, Universidad de Sevilla, 2001.
- [20] S. Wang, W. Wang, Y. Zhuang y X. Fei, «An ontology evolution method based on folksonomy,» *Journal of applied research and technology*, vol. 13, pp. 177-187, 2015.
- [21] J. Hernández, «Ontologías autónomas mediante el uso de redes neuronales auto-organizadas,» de *Proc. Latin American and Caribbean Conference for Engineering and Technology*, 2013.
- [22] F. Zablith, «Dynamic ontology evolution,» de *Proc. ISWC 2008 Doctoral consortium*, 2008.
- [23] F. Zablith, M. d'Aquin, M. Sabou y E. Motta, «Using ontological contexts to assess the relevance of statements in ontology evolution,» *Knowledge Engineering and Management by the Masses*, pp. 226-240, 2010.
- [24] S. Benomrane, Z. Sellami y M. Ayed, «An ontologist feedback driven ontology evolution with an adaptive multi-agent system,» *Advanced Engineering Informatics*, vol. 30, pp. 337-353, 2016.

- [25] P. Lambrix, Z. Dragisic, V. Ivanova y C. Anso, «Visualization for ontology evolution,» de *2nd International Workshop on Visualization and Interaction for Ontologies and Linked Data*, 2016.
- [26] F. Osborne y E. Motta, «Pragmatic Ontology Evolution: Reconciling User Requirements and Application Performance,» de *International Semantic Web Conference (ISWC)*, 2018.
- [27] H. Kondylakis y N. Papadakis, «EvoRDF: evolving the exploration of ontology evolution,» *The Knowledge Engineering Review*, 2018.
- [28] M. Moshirpour, R. Alhajj, M. Moussavi y B. Far, «Detecting emergent behavior in distributed systems using an ontology based methodology,» de *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC)* , 2011.
- [29] T. Froehner, M. Nickles y G. Weiss, «Towards modeling the social layer of emergent knowledge using open ontologies,» de *Proc. of The ECAI 2004 Workshop on Agent-Mediated Knowledge Management*, 2004.
- [30] P. Mika, «Ontologies are us: A unified model of social networks and semantics,» de *Proc. Semantic Web–ISWC 2005*, 2005.
- [31] M. Safyan, Z. Qayyum, S. Sarwar, M. Iqbal, R. Castro y A. Al-Dulaimi, «Ontology evolution for personalised and adaptive activity recognition,» *IET Wireless Sensor Systems*, 2019.
- [32] E. Puerto, J. Aguilar y T. Rodriguez, «Automatic Learning of Ontologies for the Semantic Web: A case study on lexical learning,» de *Proc. Multiagent System Based Learning Environments (MASLE)*, 2012.
- [33] M. Philippe, «Webkb.org,» [En línea]. Available: <http://www.webkb.org/>. [Último acceso: 23 Julio 2014].
- [34] A. Maedche y S. Staab, «The Text-to-Onto Ontology Learning Environment,» de *Proc. Eight International Conference on Conceptual Structures*, 2001.
- [35] P. Velardi, R. Navigli, A. Cuchiarrelli y R. Neri, «Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies,» *Learning from Text: Methods, evaluation and applications*, pp. 92-106, 2004.
- [36] P. Velardi, S. Faralli y R. Navigli, «OntoLearn Reloaded: A Graph-based Algorithm for Taxonomy Induction,» *Computational Linguistics*, vol. 39, nº 3, 23 JULIO 2013.
- [37] r. g. AKSW, «DL-Learner is a tool for supervised Machine Learning in OWL and Description Logics,» [En línea]. Available: <http://dl-learner.org/Projects/DLLearner>. [Último acceso: 23 Julio 2014].

- [38] N. Foundation y N. Foundation, «Text2Onto,» [En línea]. Available: <http://neon-toolkit.org/wiki/1.x/Text2Onto>. [Último acceso: 23 Julio 2014].
- [39] B. Ganter, G. Stumme y R. Wille, *Formal concept analysis: foundations and applications*, Springer, 2005.
- [40] G. Antoniou y F. Van Harmelen, *A Semantic Web Primer*, Massachusetts Institute of Technology, USA, 2008.
- [41] A. Zouaq, D. Gasevic and M. Hatala, "Unresolved Issues in Ontology Learning," in *Proc. 3rd Canadian Semantic Web Symposium*, 2011.
- [42] O. Corcho, M. Fernández y A. Gómez, «Ontological Engineering: Principles, Methods, tools and Languages,» *Ontologies for Software Engineering and Software Technology*, pp. 1-48, 2006.
- [43] G. Stumme, A. Hotho y B. Berendt, «Semantic Web Mining State of the art and future directions,» *Web semantics: Science, services and agents on the world wide web*, vol. 4, nº 2, pp. 124-143, 2006.
- [44] F. Christian y M. Reyes, «Análisis de la relación entre la ingeniería del conocimiento y la gestión del conocimiento en base al modelo de Nonaka y Takeuchi,» *Intangible Capital*, vol. 1, nº 9, 2005.
- [45] R. Studer, V. Benjamins y D. Fensel, «Knowledge Engineering: Principles and methods,» *Data & knowledge engineering*, vol. 25, nº 1, pp. 161-197, 1998.
- [46] J. Palma, E. Paniagua, F. Martín y R. Marín, «Ingeniería del Conocimiento. De la Extracción al modelado de Conocimiento,» *Revista Iberoamericana de Inteligencia Artificial*, nº 11, pp. 46-72, 2000.
- [47] A. Martín y C. León, «Técnicas Inteligentes para la Recuperación de Conocimiento en la Enseñanza Universitaria,» *nPixel-Bit: Revista de Medios y Educación Universidad de Sevilla*, nº 38, pp. 75-89, 2011.
- [48] R. Pedraza, L. Codina y C. Rovira, «Web semántica y ontologías en el procesamiento de la información documental,» *El profesional de la información*, vol. 16, nº 6, pp. 569-578, 2007.
- [49] A. Maedche y S. Staab, «Ontology Learning for the Semantic Web,» *IEEE Intelligent Systems*, vol. 16, nº 2, pp. 72-79, 2001.
- [50] «RDF Primer,» World Wide Web Consortium, [En línea]. Available: <http://www.w3.org/TR/rdf-primer/>. [Último acceso: 23 Julio 2014].

- [51] «OWL Web Ontology Language Overview,» World Wide Web Consortium, [En línea]. Available: <http://www.w3.org/TR/owl-features>. [Último acceso: 23 Julio 2014].
- [52] T. Gruber, «Ontolingua: A mechanism to support portable ontologies,» Stanford University, Knowledge Systems Laboratory, 1992.
- [53] H. Knublauch, R. Fergerson, N. Noy y M. Musen, «The Protégé OWL plugin: An open development environment for semantic web applications,» de *Proc. The Semantic Web—ISWC 2004*, 2004.
- [54] «METHONTOLOGY,» Semantic Web, [En línea]. Available: <http://semanticweb.org/wiki/METHONTOLOGY>. [Último acceso: 8 Julio 2014].
- [55] «TOVE Ontology Project,» Enterprise Integration Laboratory University of Toronto, [En línea]. Available: <http://www.eil.utoronto.ca/enterprise-modelling/tove>. [Último acceso: 23 Julio 2014].
- [56] «KACTUS ESPRIT Project 8145,» Human-Computer Studies Laboratory , University of Amsterdam, [En línea]. Available: <http://hcs.science.uva.nl/projects/Kactus/>. [Último acceso: 8 Julio 2014].
- [57] C. Rangel, J. Aguilar, M. Cerrada y J. Altamiranda, «An Approach for the Emerging Ontology Alignment based on the Bees Colonies,» de *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, 2015.
- [58] A. Gangemi, D. Pisanelli y G. Steve, «Ontology integration: Experiences with medical terminologies,» de *Formal ontology in information systems*, IOS Press, 1998, pp. 98-94.
- [59] J. Euzenat y P. Shvaiko, *Ontology Matching*, Springer, 2013.
- [60] A. Cuevas y A. Guzman, «Automatic Fusion of knowledge stored in Ontologies,» *Journal Intelligent Decision Technologies - Engineering and management of IDTs for knowledge management systems*, vol. 4, nº 1, pp. 5-19, 2010.
- [61] M. Shamsfard y A. Barforoush, «The state of the art in ontology learning: a framework for comparison,» *The Knowledge Engineering Review*, vol. 18, nº 4, pp. 293-316, 2003.
- [62] A. Maedche y S. Staab, «Ontology learning,» de *Handbook on ontologies*, Berlin, Springer, 2004, pp. 173-189.
- [63] N. Perozo, «Modelado Multiagente Para Sistemas Emergentes y Auto-Organizados,» Tesis Doctoral, Universidad de Los Andes-Universidad Paul Sabatier, 2011.
- [64] G. Weiss, *Multiagent Systems. Modern Approach to Distributed Artificial Intelligence*,

Cambridge, USA: MIT Press, 1999.

- [65] J. Aguilar, A. Bolivar, F. Hidrobo y M. Cerrada, *Sistemas MultiAgentes y sus Aplicaciones en Automatización Industrial*, Talleres Graficos (2da Ed.), Universidad de Los Andes, 2012.
- [66] E. Rich y K. Knight, *Artificial Intelligence*, McGraw-Hill, 1994.
- [67] J. Ferber, *An introduction to Distributed Artificial Intelligence and Multi-Agent Systems*, Addison-Wesley, 1999.
- [68] F. Schweitzer, *Brownian Agents and Active Particles Collective Dynamics in the Natural and Social Sciences*, Springer Verlag, Berlin - Heidelberg, 2003.
- [69] A. Berrones y E. Schaeffer, «Computación emergente: la ventana de oportunidad,» *Ciencia UANL*, vol. 12, nº 2, pp. 122-123, 2009.
- [70] E. Bonabeau, M. Dorigo y G. Theraulaz, *Swarm Intelligence from Natural to Artificial Systems*, Oxford University Press, 1999.
- [71] S. Alonso, O. Cordon, I. Fernandez y F. Herrera, «La metaheurística de optimización basada en colonias de hormigas: modelos y nuevos enfoques,» *Optimización inteligente: técnicas de inteligencia computacional para optimización*, pp. 261-314, 2001.
- [72] G. Nicolis y I. Prigogine, *Self-Organization in Non-Equilibrium Systems*, JohnWiley, 1977.
- [73] T. Wolf y T. Holvoet, «Emergence and Self-Organisation: a statement of similarities and differences,» *Lecture Notes in Artificial Intelligence*, vol. 3464, pp. 96-110, 2004.
- [74] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx y K. De Bosschere, «Towards an extensible context ontology for ambient intelligence,» *Ambient intelligence*, pp. 148-159, 2004.
- [75] J. Rifkin, J. Rifkin y T. Howard, «Entropy: A new world view,» *North American Review*, vol. 266, p. 65-70, 1981.
- [76] A. Etxeberria y L. Bich, «Auto-organización y autopoiesis,» de *Diccionario Interdisciplinar Austral*, 2017.
- [77] C. Shannon, «A mathematical theory of communications,» *Bell System Technical Journal*, p. 379-423, 1948.
- [78] T. Li, S. Ma y M. Ogihara, «Entropy-based criterion in categorical clustering,» de *Proc. of the 21 International Conference on Machine Learning*, 2004.

- [79] D. Spivak, *Category theory for the sciences*, 2014.
- [80] S. Aliyu, S. B. Junaidu y A. D. Kana, «A Category Theoretic Model of RDF Ontology,» *International Journal of Web & Semantic Technology (IJWesT)*, vol. 6, nº 3, 2015.
- [81] A. Asperti y G. Longo, *Categories, types, and structures: an introduction to category theory for the working computer scientist*, 1991.
- [82] M. Mendonça, J. Aguilar y N. Perozo, «Application of Category Theory in the Generation of Meta-Ontologies,» *Ingénierie des Systèmes d'Information*, vol. 23, pp. 11-38, 2018.
- [83] M. Dorigo y T. Stützle, «Ant colony optimization: overview and recent advances,» de *Handbook of metaheuristics*, 2010, pp. 227-263.
- [84] M. Mendonça, J. Aguilar y N. Perozo, «An approach for Multiple Combination of Ontologies based on the Ants Colony Optimization Algorithm,» de *Proc. Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*, Ecuador, 2015.
- [85] «Catalogo de Datos del Gobierno de España,» [En línea]. Available: <http://datos.gob.es/es/catalogo>. [Último acceso: 25 11 2018].
- [86] «Gobierno de Navarra,» [En línea]. Available: <http://www.gobiernoabierto.navarra.es/es/open-data/datos/catalogo>. [Último acceso: 25 11 2018].
- [87] M. Mendonça, J. Aguilar y N. Perozo, «Middleware Reflexivo Semántico para Ambientes Inteligentes,» de *Proc. Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa)*, Caracas, 2014.
- [88] M. Mendonça, J. Aguilar y N. Perozo, «MiR-EO: Reflective Middleware for Ontological Emergency in Intelligent Environments,» *Latin American Journal of Computing*, vol. 3, 2016.
- [89] J. Aguilar, M. Jerez, M. Sánchez y M. Mendonca, «Emergencia Ontológica Basada en Análisis de Contexto, como servicio para Ambientes Inteligentes,» *DYNA*, vol. 84, nº 200, pp. 28-37, 2017.
- [90] L. González y J. Echeverri, «Modelado Conceptual de Usuarios en Ambientes Ubicuos Mediante Agentes Y Ontologías,» *Revista EIA*, nº 16, pp. 115-126, 2011.
- [91] J. Kiljander, A. Ylisaukko-oja, J. Takalo-Mattila, M. Eteläperä y J. P. Soinen, «Enabling semantic technology empowered smart spaces,» *Journal of Computer Networks and Communications*, 2012.

- [92] M. Peters, S. Brink, S. Sachweh y A. Zündorf, «Performance considerations in ontology based ambient intelligence architectures,» de *Ambient Intelligence-Software and Applications*, 2013, pp. 121-128.
- [93] M. Bruno y J. Ierache, «Ontología para la Interacción de Agentes en un Hábitat inteligente,» de *Proc.XII Congreso Argentino de Ciencias de la Computación*, 2006.
- [94] I. A. C. A. Team, An architectural blueprint for autonomic computing, Technical report, IBM Corporation, Hawthorne, 2006.
- [95] P. Maes, «Concepts and Experiments in Computational Reflection,» de *Proc. ACM Conference on Object-Oriented Programming*, 1987.
- [96] C. J. A. J. y. S. M. Valdiviezo P., «Conceptual Design of a Smart Classroom Base on Multiagent System,» de *Proc.2015 International Conference on Artificial Intelligence (ICAI'15)*, 2015.
- [97] A. J. C. J. y. V. P. Sánchez M., «A Smart Learning Environment based on Cloud Learning,» de *Proc. International Journal of Advanced Information Science and Technology (IJAIST)*, 2015.
- [98] J. M. E. E. y. V. T. Aguilar J., «CARMiCLOC: Context Awareness Middleware in Cloud Computing,» de *Proc. Latin American Computing Conference (CLEI)*, 2015.
- [99] S. Pellicer, G. Santa, A. Bleda y R. Maestre, «A global perspective of smart cities: a survey,» de *Proc. Seventh International Conference, Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 2013.
- [100] J. Aguilar, M. Jerez, M. Mendonça y M. Sánchez, «MiSCi: Autonomic Reflective Middleware for Smart Cities,» *Proc. Congreso Internacional de Tecnología e Innovación (CITI)*, pp. 241-253, 2016.
- [101] S. Yi, c. Li y Q. Li, «A survey of fog computing: concepts, applications and issues.,» de *Proceedings of the 2015 Workshop on Mobile Big Data*, 2015.
- [102] F. Bonomi, R. Milito, J. Zhu y S. Addepalli, «Fog computing and its role in the internet of things,» de *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012.
- [103] S. Babu, A. Lakshmi y B. Rao, «A study on cloud based Internet of Things: CloudIoT,» de *Communication Technologies (GCCT) 2015 Global Conference*, 2015.
- [104] A. Giordano, G. Spezzano y A. Vinci, «Smart Agents and Fog Computing for Smart City Applications,» de *International Conference on Smart Cities*, 2016.

- [105] J. Aguilar, M. Jerez, M. Mendonça y M. Sánchez, «An extension of the MISCI Middleware for Smart Cities based on Fog Computing,» *Journal of Information Technology Research (JITR)*, vol. 10, nº 4, 2017.
- [106] M. Mendonça, J. Aguilar y N. Perozo, «Una Ontología Emergente para Ambientes Inteligentes basada en el Algoritmo de Optimización por Colonias de Hormigas,» de *Proc. Latin American Computing Conference (CLEI)*, Uruguay, 2014.
- [107] K. Morik, «Balanced cooperative modeling,» de *Multistrategy Learning*, Springer US, 1993, pp. 109-127.
- [108] F. Roper, «Método para la evaluación automática de la organización de textos argumentativos,» Tesis Doctoral, Universidad Nacional de Colombia, 2014.
- [109] «DTD Tutorial,» W3Schools, [En línea]. Available: https://www.w3schools.com/xml/xml_dtd_intro.asp. [Último acceso: 23 Julio 2014].
- [110] «XML Extensible Markup Language,» World Wide Web Consortium, [En línea]. Available: <https://www.w3.org/TR/xml/#dt-xml-doc>. [Último acceso: 23 Julio 2014].
- [111] «XML Schema Definition Language (XSD),» World Wide Web Consortium, [En línea]. Available: <https://www.w3.org/TR/xmlschema11-1/>. [Último acceso: 23 Julio 2014].
- [112] «Linked Open Vocabularies for Internet of Things,» [En línea]. Available: <http://lov4iot.appspot.com/>.
- [113] «SWOOGLE,» [En línea]. Available: <http://swoogle.umbc.edu>.
- [114] V. Levenshtein, «Binary codes capable of correcting deletions, insertions, and reversals. In,» *Soviet physics doklady*, vol. 10, nº 8, pp. 707-710, 1966.
- [115] G. Szekely y M. Rizzo, «Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method,» *Journal of classification*, vol. 22, nº 2, pp. 151-183, 2005.
- [116] F. Bauer y M. Kaltenböck, *Linked open data: The essentials*, Edition mono/monochrom, Vienna, 2011.

Apéndices

5.1. Apéndice A: Mecanismo de Organización de una Meta-Ontología basado en ACO

A continuación se presenta el uso de una técnica propia de la computación emergente, como es el algoritmo de Optimización por Colonias de Hormigas ACO (Ant Colony Optimization, por sus siglas en inglés), en la estructuración de la ontología emergente, específicamente en la organización o población de de la MOD. Antes de mostrar el proceso, se explica brevemente en que consisten los Algoritmos ACO.

Algoritmos ACO

Los algoritmos ACO se inspiran directamente en el comportamiento de las colonias reales de hormigas, para solucionar problemas de optimización combinatoria. Se basan en una colonia de hormigas artificiales, esto es, agentes computacionales simples que trabajan de manera cooperativa, y se comunican mediante rastros de feromona artificiales. Los algoritmos ACO son constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema, recorriendo un grafo que contiene soluciones parciales. Cada arista del grafo, que representan los posibles pasos que la hormiga puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga: la información heurística, que mide la preferencia heurística de moverse desde un nodo hasta otro, y la información de los rastros de feromona artificiales, que mide la “deseabilidad aprendida” de ese movimiento, que imita a la feromona real que depositan las hormigas naturales. Esta información es la que permite el aprendizaje colectivo, y se va alimentado durante la ejecución del algoritmo con las soluciones encontradas por las hormigas [71] [83].

Estos algoritmos son muy usados en problemas de optimización combinatoria. En [106] se realizó una primera propuesta del uso de ACO en la generación de una ontología para ambientes inteligentes. En estos problemas, un aspecto importante es definir el espacio de soluciones que será recorrido por la colonia de hormigas en la búsqueda de la solución óptima. En este caso, el espacio de soluciones viene dado por la MOD, que debe ser recorrida, para ubicar los CEs. El problema viene dado por la necesidad de agregar adecuadamente cada CE en la MOD, a través de la creación de una relación “es-un”, como sub-clase de uno de los meta-conceptos. En este proceso de clasificación, se debe caracterizar el nuevo concepto, para ubicarlo en el mismo grupo de conceptos similares ya almacenados en las ontologías, basado en sus propiedades. La idea es que

por cada CE, las hormigas puedan seleccionar un meta-concepto al cual asociarlo, a través de la información heurística que viene dada por una medida de similitud entre los conceptos (Def. 3.17), y también, a través de los rastros de la feromona, lo que le permitirá explorar todas las posibles soluciones.

Determinación del Espacio de Soluciones

Para la aplicación de ACO a un problema dado, es necesario, en primer lugar, definir el espacio de soluciones que será recorrido por la colonia de hormigas en la búsqueda de la solución óptima. En este caso, se considera que el problema viene dado por la necesidad de agregar un concepto emergente CE a la MOD, a través de la creación de una relación “es-un” con algún meta-concepto (Fig. A.1).

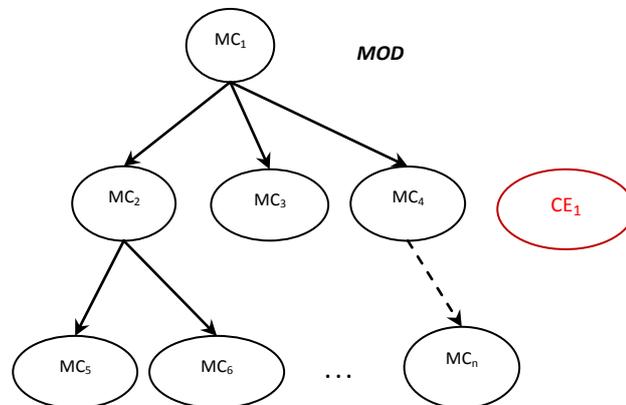


Figura A.1. Un CE a incluir en una MOD

Si en la meta-ontología ya existen n meta-conceptos, entonces se tiene n posibilidades o lugares donde agregar el concepto emergente CE_1 :

$$CE_1 \subseteq MC_1$$

$$CE_1 \subseteq MC_2$$

$$CE_1 \subseteq MC_3$$

...

$$CE_1 \subseteq MC_n$$

Así, se puede ver en la Fig. A.2 todas las posibles ubicaciones del nuevo concepto en la taxonomía ya existente. Si se necesitan agregar dos conceptos CE_1 y CE_2 , entonces tendríamos un espacio de soluciones mucho más grande, y así sucesivamente.

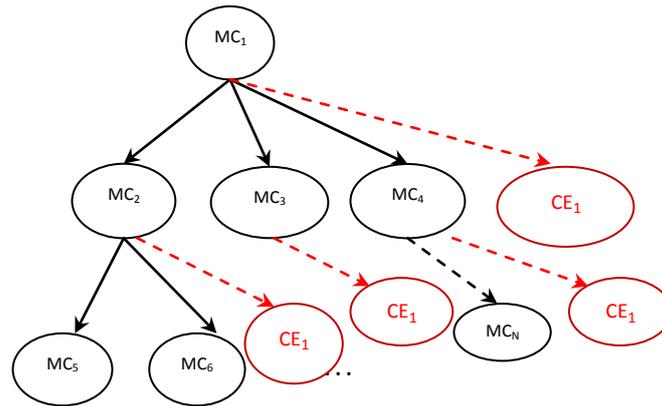


Figura A.2. Posibles posiciones para ubicar un CE en la MOD

De manera general, si se tienen m CEs para agregar en una MOD de n meta-conceptos, el problema tendría un espacio de n^m posibles soluciones.

Determinación de la Información Heurística

El problema central es determinar cuál de todas las soluciones posibles es la más óptima, es decir, cuál es la mejor ubicación de un CE en la MOD, en función de la mayor semejanza con los meta-conceptos existentes, en base a la Similitud Conceptual (Def. 3.17).

Así, el valor de la información heurística η de cómo el concepto emergente CE_1 se pueda ubicar con una relación "es-un" del meta-concepto MC_1 , viene determinado por el valor de la Sim_C con el MC_1 . De esta manera, la semejanza (SEM) de CE_1 con respecto a MC_1 vendrá dada por:

$$SEM_{CE_1MC_1} = Sim_C(CE_1, MC_1)$$

Entonces, el valor de η que representa el factor de decisión de la hormiga de seleccionar una arista, viene dado en función de SE_{CEMC} . Si la hormiga está parada en un meta-concepto R , podrá decidir colocar a CE como sub-clase de MC (que es hijo de R), en función de la semejanza entre el CE y el MC.

Finalmente, si el meta-concepto R tiene N meta-conceptos hijos, entonces la hormiga puede comparar con los N meta-conceptos, y también, considerará el meta-concepto base actual R en su decisión. Si ninguno de los valores de similitud de los N meta-conceptos hijos supera el valor de similitud con el meta-concepto actual R , la hormiga seleccionará quedarse con la solución R . Al no existir un cambio en la solución actual, se considera que la hormiga no va a agregar más niveles a la solución actual.

Algoritmo General basado en ACO

El Macro-Algoritmo A.1 presenta el algoritmo ACO usado en nuestro trabajo. De manera general, se explicarán los elementos más importantes del algoritmo: la inicialización de las hormigas, la función de transición en el grafo de soluciones, y la función de evaporación y actualización de la feromona.

MACRO-ALGORITMO A.1: ALGORITMO GENERAL DE ACO

Entradas: Espacio de Soluciones

Procedimiento:

```

//Inicialización de parámetros
// Rastro inicial de feromona en cada conexión
// (Z= número de conexiones)
// (c= Valor positivo muy pequeño)
1. Repetir de j=1 hasta Z
  1.1.  $\tau_j = c$ 
      // Número de hormigas en la colonia
2.  $m = n$ 
   // Pesos (importancia) de la inf. heurística y memorística (feromona)
   // Peso de la feromona
3.  $\alpha = X$ 
   // Peso de la heurística
4.  $\beta = Y$ 
5. Mientras (Verificar_Terminación())
  5.1. Repetir desde k=1 hasta m (numero_hormigas)
    5.1.1. Crear_Nueva_Hormiga()
          //Inicializar_Hormiga
    5.1.2.  $k=id\_hormiga$ 
          //Actualizar_Memoria_Hormiga
    5.1.3.  $r=estado\_inicial (X1 \subseteq thing)$ 
    5.1.4.  $Lk=r$ 
    5.1.5. Mientras (estado_actual  $\neq$  estado_objetivo)
      //Calcular probabilidades de transición
      5.1.5.1.  $P_{(r,s)}^k = \frac{\gamma_{(r,s)}^\alpha \cdot \eta_{(r,s)}^\beta}{\sum_{u \in J_r^k} \gamma_{(r,u)}^\alpha \cdot \eta_{(r,u)}^\beta}$ 
      //Aplicar probabilidades de transición
      5.1.5.2. siguiente_estado = aplicar_política_decisión(P,r)
      5.1.5.3.  $r = siguiente\_est;$ 
      //Actualización de la feromona
      5.1.5.4.  $\gamma_{(r,s)} = \gamma_{(r,s)} + \Delta \gamma_{(r,s)}$ 
      //actualiza estado interno
      5.1.5.5.  $Lk=Lk \cup r$ 
    5.1.6. liberar_recursos_hormiga(id_Hormiga)
  //Evaporacion de la Feromona

```

Salidas: Mejor Solución

a) *Inicialización de las hormigas:* Es importante destacar, que debe existir una MOD inicial definida, donde la hormiga realizará el recorrido. Se debe definir un punto de partida o estado inicial donde la hormiga comienza su recorrido por el grafo de soluciones. En este caso, se inicia en la raíz de toda ontología, que es el concepto general “thing”. Todos los conceptos posibles de la ontología inicialmente pueden tener una relación “es-un” con el concepto “thing”, por lo tanto, esa es la primera solución posible:

$CE_1 \subseteq thing$

El proceso de recorrido del grafo (ontología) por las hormigas en la búsqueda de la solución se resume en el Macro-Algoritmo A.2. La hormiga inicia con esa solución, y luego evalúa otras posibles soluciones, para determinar si CE_1 puede clasificarse como una subclase de algunos de los meta-conceptos derivados del meta-concepto seleccionado como solución actual. Si el meta-concepto actual no tiene meta-conceptos derivados, entonces en ese momento se tiene una solución para el concepto emergente CE_1 . El proceso se debe realizar para todos los conceptos a agregar en la MOD.

MACRO-ALGORITMO A.2: BÚSQUEDA DE UNA SOLUCIÓN

Entradas: MOD

Procedimiento:

6. Repetir desde $i=1$ hasta $i=\text{Numero de Hormigas}$;
 - 6.1. Solución_Actual="Thing";
 - 6.2. Seguir=Verdadero;
 - 6.3. Repetir mientras (Seguir==Verdadero)
 - 6.3.1. Buscar hijos de Solución_actual();
 - 6.3.2. Si "Tiene hijos"
 - 6.3.2.1. Repetir desde $j=1$ hasta $j=\text{Número de Hijos}$;
 - 6.3.2.2. Calcular Semejanza();
 - 6.3.2.3. Nueva_Solucion=Mayor_Semejanza();
 - 6.3.2.4. Si Nueva_Solucion==Solucion_Actual
 - 6.3.2.5. Seguir=Falso
 - 6.3.2.6. De lo Contrario
 - 6.3.2.7. Solucion_Actual=Nueva_Solucion
 - 6.3.3. Si "No Tiene hijos"
 - 6.3.3.1. Seguir=Falso

Salidas: MOD re-organizada

b) *Función de transición:* Para la aplicación de ACO se hará uso de una función de probabilidad, que determinará cuál sub-clase seleccionar a partir de una ubicación específica en la ontología. De esta forma, para ir construyendo la solución, la hormiga k debe escoger cuál es el siguiente elemento de la solución desde la ubicación " r ". Para ello, usa una función de probabilidad para seleccionar el elemento " s ", que se calcula de la siguiente manera:

$$P_{(r,s)}^k = \frac{\gamma_{(r,s)}^\alpha \cdot \eta_{(r,s)}^\beta}{\sum_{u \in J_r^k} \gamma_{(r,u)}^\alpha \cdot \eta_{(r,u)}^\beta} \quad \text{Si } s \in J_r^k$$

Donde:

- $\gamma_{(r,s)}$: Cantidad de feromona.
- $\eta_{(r,s)}$: Información Heurística en función de la semejanza SEM.
- J_r^k : Nodos aún no visitados por la hormiga k desde r .
- α y β : Importancia de la información memorística (feromona) y heurística.

c) *Evaporación de la feromona:* Luego que todas las hormigas han terminado su recorrido, se realiza el proceso de evaporación de la feromona. Los rastros de la feromona en cada arco se van reduciendo de acuerdo a un factor constante, a través de la siguiente función:

$$Y_{(r,s)} = (1 - \rho) \cdot Y_{(r,s)}$$

Donde:

- $\rho \in (0,1]$ es la tasa de evaporación

d) *Actualización de la feromona*: Mientras cada hormiga está en el proceso de construcción de la solución, por cada arista que selecciona debe actualizar la feromona, depositando una cantidad de feromona en esa conexión. De esta manera, la feromona en cada arista se actualiza de acuerdo a la siguiente función:

$$Y_{(r,s)} = Y_{(r,s)} + \Delta Y_{(r,s)}$$

Donde:

$\Delta Y_{(r,s)}$ es el incremento de la feromona, que viene dado por la sumatoria de las cantidades de feromona dejadas por las hormigas en el arco (r, s) :

$$\Delta Y_{(r,s)} = \sum_{k=1}^M \Delta Y_{(r,s)}^k$$

$\Delta Y_{(r,s)}^k$ es la cantidad de feromona dejada por la hormiga k en el arco (r, s) , la cual está en función de la *Calidad de la Solución* encontrada por la hormiga k

$$\Delta Y_{(r,s)}^k = f(C(S_K))$$

Donde:

- S_K : meta-concepto seleccionado por la hormiga k como mejor solución

La función para obtener la "*Calidad de la Solución*" encontrada viene dada por el nivel de Entropía Conceptual (EC) del meta-concepto, al colocar el CE en esa posición. Ya que la cantidad de feromona depositada por la hormiga debe ser mayor en la medida que será mejor la calidad de la solución, la solución se considera mejor mientras menor sea la Entropía. Para obtener un valor de calidad adecuado se debe normalizar (llevar a una escala de 0 a 1) el valor de la EC. El valor normalizado de EC se obtiene dividiéndolo entre el UME, que es el valor máximo posible de entropía permitido:

$$EC_N(S_K) = \frac{EC(S_K)}{UME}$$

Luego con el valor normalizado de la EC, se obtiene la calidad de la solución:

$$C(S_K) = 1 - EC_N(S_K)$$

5.2. Apéndice B: Herramienta para la Aplicación del EEO.

Para mostrar la aplicación del EEO se implementó una herramienta, para la cual se usó el entorno de desarrollo NetBeans 7.4 con la plataforma JADE ("Java Agent DEvelopment Framework", por sus siglas en inglés). Además, también se usó JENA, un framework para Java que provee librerías para el manejo de RDF y OWL. En la Fig. B.1 se muestran las partes de la herramienta.

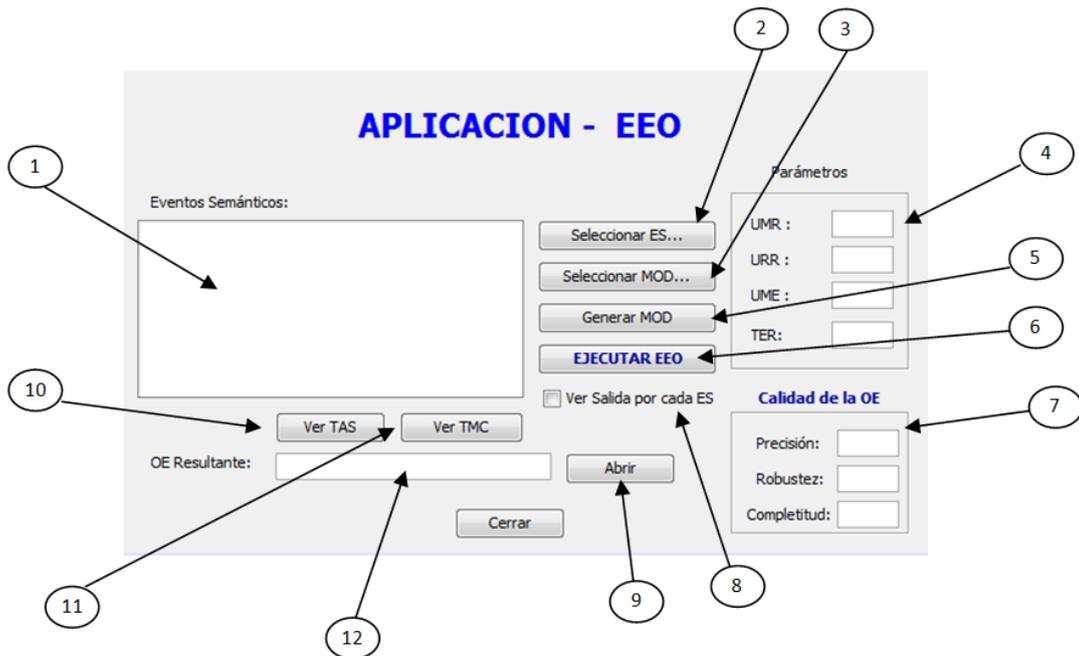


Figura B.1. Partes de la herramienta de aplicación del EEO.

A continuación se explica la interface de la herramienta y sus partes :

1. Se muestran los eventos semánticos que se han seleccionado.
2. Para seleccionar los archivos (XML, XLS, OWL) que corresponden eventos semánticos.
3. Selecciona Meta-Ontologías de Dominio Pre-definidas.
4. Definir parámetros iniciales: UMR, URR, UME, TER.
5. Generar Meta-Ontologías de Dominio a partir de ontologías ya registradas.
6. Ejecutar el EEO.
7. Muestra los valores de calidad de la OE.
8. Para indicar si se desea visualizar una salida (OE) por cada evento semántico, o al final de procesar todos los eventos semánticos seleccionados.
9. Visualizar la OE resultante.
10. Ver la Tabla de Anotaciones Semánticas resultante.
11. Ver la Tabla de Meta-Conceptos resultante.
12. Muestra el nombre de la OE resultante.

5.3. Apéndice C: Artículos publicados en el marco de la tesis

1. MENDONÇA, M.; AGUILAR, J.; PEROZO, N., "An Emergent Ontology for Ambient Intelligence based on an Ant Colony Optimization algorithm", *In Proceedings of Computing Conference (CLEI) Uruguay, 2014, p. 1-11.*
2. MENDONÇA, M.; AGUILAR, J.; PEROZO, N., "Middleware Reflexivo Semántico para Ambientes Inteligentes", *In Proceedings of Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa), 2014, Venezuela, p. 24-32.*
3. MENDONÇA, M.; PEROZO, N.; AGUILAR, J., "An approach for Multiple Combination of Ontologies based on the Ants Colony Optimization Algorithm", *In Proceedings of Asia-Pacific Conference on Computer Aided System Engineering (APCASE), 2015, Ecuador, p. 140-145.*
4. AGUILAR, J.; JEREZ, M.; MENDONCA, M.; SÁNCHEZ, M., "MiSci: Autonomic Reflective Middleware for Smart Cities", *In Proceedings of Congreso Internacional de Tecnología e Innovación (CITI), 2016, Ecuador, p. 241-253.*
5. MENDONÇA, M.; AGUILAR, J.; PEROZO, N., "MiR-EO: Reflective Middleware for Ontological Emergency in Intelligent Environments", *Latin American Journal of Computing Faculty of Systems Engineering National Polytechnic School Quito-Ecuador, 2016, vol. 3, no 2, p.25-39.*
6. AGUILAR, J.; JEREZ, M.; MENDONCA, M.; SÁNCHEZ, M., "Emergencia Ontológica Basada en Análisis de Contexto, como servicio para Ambientes Inteligentes", *DYNA: International Journal of the Facultad de Minas, Universidad Nacional de Colombia, Medellin, 2017, vol. 84, no 200, p. 28-37.*
7. AGUILAR, J.; JEREZ, M.; MENDONCA, M.; SÁNCHEZ, M., "An extension of the MiSci Middleware for Smart Cities based on Fog Computing", *Journal of Information Technology Research (JITR), 2017, vol. 10, no 4 , p. 23-41.*
8. MENDONÇA, M.; PEROZO, N.; AGUILAR, J., "Application of Category Theory in the Generation of Meta-Ontologies ", *Ingénierie des Systèmes d'Information Journal Lavoisier, 2018, vol 23, p. 11-38.*
9. MENDONÇA, M.; PEROZO, N.; AGUILAR, J., "Ontological Emergence Scheme in Self-Organized and Emerging Systems", *Advanced Engineering Informatics. En revision.*