

Enriquecimiento Semántico del Léxico Extendido de Lenguaje (LEL) mediante Ontologías

Marc Marinho de Alcântara¹, Leandro Antonelli^{1 2}, Diego Torres^{1 3}

¹ LIFIA, Fac. de Informática, Universidad Nacional de La Plata

² CAETI - Facultad de Tecnología Informática - Universidad Abierta Interamericana

³ Departamento de Ciencias y Tecnología, Universidad Nacional de Quilmes, Bernal, Argentina.

marc.alcantara@gmail.com, {leandro.antonelli,
diego.torres}@lifia.info.unlp.edu.ar

Abstract. La fase de elicitación en ingeniería de requisitos tiene como objetivo comprender el Universo de Discurso (UofD) y construir un vocabulario compartido entre los stakeholders y el equipo de desarrollo. El Léxico Extendido del Lenguaje (LEL) es una técnica ampliamente utilizada para este fin, ya que permite representar el conocimiento del dominio mediante lenguaje natural; sin embargo, la ambigüedad inherente a este tipo de lenguaje dificulta la verificación sistemática de la consistencia del léxico a medida que este evoluciona. Este trabajo propone un metamodelo del LEL formalizado en el Lenguaje de Ontologías Web (OWL), que explicita los elementos estructurales del léxico y sus relaciones como un modelo verificable. A diferencia de enfoques tradicionales, en los que el LEL se mantiene como un artefacto descriptivo, la propuesta permite incorporar restricciones formales que habilitan la detección automática de inconsistencias durante la elicitación. Asimismo, este trabajo define un proceso incremental de construcción del léxico guiado por dicho metamodelo, integrando la validación estructural como parte del flujo de trabajo del ingeniero de requisitos y promoviendo la coherencia y trazabilidad del conocimiento del dominio desde etapas tempranas. La propuesta se ilustra mediante su aplicación en un caso de estudio del dominio de una línea aérea, demostrando la viabilidad del enfoque mediante el uso de OWL y SPARQL como base tecnológica.

Keywords: Léxico Extendido del Lenguaje (LEL), Vocabulario de Dominio, Elicitación, GAP Semántico, Ontologías, OWL, Web Semántica.

1 Introducción

La ingeniería de software moderna reconoce que la mayoría de los fallos críticos en sistemas complejos no se originan en errores de codificación, sino en deficiencias de comunicación durante las fases tempranas del desarrollo. La comprensión precisa del Universo de Discurso (UofD) representa el primer desafío cognitivo y metodológico al que se enfrentan los ingenieros[2][3].

Durante la fase de elicitación, el objetivo primordial no es definir qué hará el sistema (es decir, no se busca aún especificar los requerimientos), sino capturar el conocimiento tácito de los expertos del negocio y establecer un vocabulario riguroso, común y compartido entre todos los stakeholders. El principio rector de esta etapa es: “entender el lenguaje del problema antes de preocuparse por la solución”[14][5].

Para este fin, el Léxico Extendido del Lenguaje (LEL), propuesto originalmente por J.C.S.P. Leite, se ha consolidado como una técnica metodológica sobresaliente. El LEL estructura los términos del dominio en símbolos específicos, dotándolos de nociones (denotaciones) e impactos (connotaciones) [8]. No obstante, el crecimiento inherente a cualquier proyecto de escala industrial expone la vulnerabilidad fundamental del LEL: su dependencia exclusiva del lenguaje natural. A medida que el glosario se expande, se vuelve humanamente inmanejable detectar contradicciones sutiles, ambigüedades, sinonimias no declaradas o clasificaciones taxonómicas incompatibles. Detectar estas inconsistencias de forma manual mediante checklists es propenso a errores y consume excesivos recursos, generando un “ruido comunicacional” que inevitablemente se propagará a la fase de requisitos [15].

El problema central que aborda este trabajo es la brecha existente entre la riqueza expresiva del lenguaje natural utilizado en el LEL y la necesidad de un rigor matemático que garantice la integridad del modelo [15]. La hipótesis subyacente es que la inyección de un formalismo lógico en la estructura del LEL permite automatizar la detección de conflictos sin perder la legibilidad del texto para el usuario de negocio [2].

Frente a esta brecha, donde los enfoques tradicionales posponen la validación para etapas tardías, la contribución principal de este artículo es un método preventivo e incremental para validar semánticamente el vocabulario del dominio durante su elicitación. El método guía al ingeniero de requisitos en la estructuración del lenguaje natural, su captura léxica y su transformación hacia un esquema conceptual verificable. De este modo, permite auditar la consistencia del léxico de forma progresiva, asegurando bases de conocimiento sólidas antes de iniciar la especificación funcional.

La investigación parte de la hipótesis de que la incorporación de una capa ontológica sobre la estructura del LEL puede mejorar su consistencia y calidad semántica, permitiendo adelantar validaciones que normalmente ocurren en etapas posteriores del proceso de ingeniería de software. En este enfoque, el LEL mantiene su rol como mecanismo de comunicación y representación comprensible para los stakeholders, mientras que la ontología introduce capacidades de validación formal y razonamiento automático. Asimismo, se exploran posibles beneficios adicionales derivados del uso de tecnologías semánticas durante las actividades tempranas de ingeniería de requisitos.

El documento sigue una secuencia estructurada para presentar el método y su validación. La Sección 2 es el marco conceptual (Background), presentando los fundamentos teóricos sobre el LEL, la técnica de Kernel Sentences para la simplificación del lenguaje. La Sección 3 constituye el núcleo científico de la propuesta (Proposed Method), donde se profundiza la técnica de meta-modelado del LEL en OWL hallada en la investigación, que permite dar forma a los símbolos LEL en una estructura computable. La Sección 4 describe el método propuesto de aplicación de la meta-

ontología en el proceso de elicitación. La Sección 5 constituye el soporte tecnológico y la prueba de concepto con abordaje del dominio de una aerolínea. La sección 6 contrasta la propuesta con los trabajos relacionados. Finalmente, la Sección 7 sintetiza las conclusiones.

2 Background

2.1 Kernel Sentences

Kernel Sentences son simples construcciones con un solo verbo. Puede ser positivo, negativo, activo o pasivo. Su forma simple y fundamental explica por qué se llama kernel. No contiene ninguna otra construcción del lenguaje manteniendo solamente elementos que permiten un análisis acertado y lógico y al mismo tiempo trayendo elementos que pueden servir como entidades importantes de negocio[16]. Por ejemplo, Ejemplo 1 muestra dos kernel sentences, con el detalle que el verbo ‘estar’ no tiene sentido semántico todavía. Por eso el segundo ejemplo tiene dos verbos: embarca y está. En el Ejemplo 2 muestra dos frases que no son kernel sentences porque tienen dos verbos. Uno tiene ‘validar’ y otro tiene ‘asegurar’. La primera frase puede ser reescrita en dos kernel sentences(Ejemplo 3).

Ejemplo 1. Kernel Sentences [16]

El Pasajero adquiere un billete aéreo
El Pasajero embarca en un vuelo en el terminal que está asignado

Ejemplo 2. Oraciones que no cumplen las reglas de Kernel Sentences [16]

El miembro de operación valida el billete aéreo para asegurar que el pasajero del billete sea el mismo.
El miembro de mantenimiento realiza mantenimiento preventivo para cumplir las reglas de seguridad.

Ejemplo 3. Kernel Sentences reescrita [16]

El miembro de operación valida el billete aéreo
La validación del billete aéreo asegura que el pasajero del billete sea el mismo.

2.2 Language Extended Lexicon

El LEL es un artefacto de ingeniería diseñado para registrar los signos (símbolos) peculiares del contexto de una aplicación. El modelo del LEL obliga al ingeniero a abstraerse de la tecnología y enfocarse en comprender el entorno. Cada entrada (Símbolo) en el léxico se describe en estructura [5]:

Noción (Denotación): Conjunto de oraciones que definen “qué es” el símbolo, detallando sus características estructurales o su pertenencia a un grupo mayor.

Impacto (Connotación): Conjunto de oraciones que describen la respuesta comportamental del símbolo; es decir, “cómo actúa” en el sistema o “qué acciones recaen sobre él”.

Además, por categoría de objetos [5]:

Nombre del Símbolo: Identificador principal y sus posibles sinónimos.

Clasificación: Todo símbolo debe pertenecer estrictamente a una de cuatro categorías mutuamente excluyentes:

Sujeto: Entes activos que realizan acciones (ej. Pasajero, Piloto).

Objeto: Entes pasivos sobre los cuales recaen las acciones (ej. Aeronave, Billeto Aéreo).

Verbo: Acciones, procesos o transacciones ejecutadas por sujetos sobre objetos (ej. Abordar, Cancelar).

Estado: Situaciones transitorias condicionantes (ej. En_Vuelo, Atrasado).

El rigor del LEL humano depende de dos reglas empíricas:

1. *Principio de Circularidad (Clausura):* Exige maximizar el uso de los propios símbolos del LEL para describir las Nociones e Impactos de otros símbolos, formando un hipertexto fuertemente interconectado.

2. *Principio de Vocabulario Mínimo:* Minimizar el uso de palabras ajenas al léxico, limitándose a conectores del lenguaje natural.

3 Ontología del LEL basada en un metamodelo

El Léxico Extendido del Lenguaje (LEL) ha sido tradicionalmente utilizado como un artefacto descriptivo en el proceso de ingeniería de requisitos, permitiendo capturar y organizar el vocabulario del dominio a partir de nociones e impactos asociados a cada término. Sin embargo, en su forma clásica, el LEL carece de una formalización explícita que permita verificar automáticamente la consistencia de su estructura o validar las relaciones entre sus elementos.

En este contexto, este trabajo propone una formalización ontológica del LEL basada en un metamodelo, en la cual los elementos estructurales del léxico —símbolos, nociones, impactos y tipos semánticos— son representados como clases y relaciones dentro de una ontología OWL. Esta formalización no solo permite representar el vocabulario del dominio, sino también explicitar las reglas que gobiernan su construcción.

A diferencia de enfoques tradicionales, en los cuales dichas reglas permanecen implícitas o dependen de la interpretación del analista, la propuesta desarrollada en este trabajo las expresa como restricciones formales, habilitando su verificación mediante herramientas de razonamiento automático. De este modo, el LEL pasa de ser un artefacto predominantemente descriptivo a un modelo estructurado y verificable.

El enfoque adoptado se apoya conceptualmente en la idea de metamodelado del dominio, en la que los elementos del lenguaje son tratados como instancias de una estructura abstracta común. En particular, presenta afinidad con el patrón Adaptive Object Model (AOM) propuesto por Fowler [20], en el que la estructura del dominio se representa mediante metadatos interpretables. No obstante, en esta propuesta dichos metadatos son formalizados mediante una ontología OWL, lo que permite incorporar restricciones explícitas y mecanismos de validación automática.

En las siguientes subsecciones se describe la ontología del metamodelo LEL, incluyendo sus clases, propiedades y axiomas principales.

3.1 Formalización ontológica del metamodelo LEL

Con el fin de formalizar la estructura del Léxico Extendido del Lenguaje (LEL) y habilitar su validación automática, se desarrolló una ontología OWL que representa explícitamente los elementos constitutivos del léxico y las relaciones entre ellos. Esta ontología constituye una de las principales contribuciones de este trabajo, ya que no se limita a documentar conceptos del dominio, sino que explicita reglas estructurales que pueden ser procesadas por herramientas de razonamiento.

La estructura del dominio es representada mediante metadatos interpretables. Sin embargo, en este trabajo dichos metadatos se formalizan mediante una ontología OWL, lo que permite incorporar restricciones explícitas y habilitar mecanismos de validación automática que no están presentes en implementaciones tradicionales de AOM.

La ontología fue desarrollada en Protégé, lo que permitió su modelado, inspección y verificación con motores de inferencia compatibles con OWL. En esta sección, se describen sus clases, propiedades y axiomas principales, ya que estos forman parte central del aporte del artículo.

3.2 Estructura conceptual del metamodelo

La ontología se organiza alrededor de la clase *LELSymbol*, que representa cada símbolo del léxico y, por lo tanto, cada concepto relevante del Universo del Discurso (UoD). En términos conceptuales, un símbolo es la unidad central del modelo: es el elemento que nombra una noción del dominio y que se encuentra caracterizado por un tipo semántico, una descripción conceptual y un conjunto de efectos o comportamientos asociados.

Para formalizar esta estructura, el metamodelo incorpora tres clases que completan la definición de cada símbolo:

Notion: representa la noción del símbolo, es decir, su descripción conceptual y sus características dentro del dominio.

Impact: representa los efectos, acciones o comportamientos asociados al símbolo.

SymbolType: representa la clasificación semántica del símbolo.

La clase *SymbolType* se especializa en las cuatro categorías tradicionales del LEL propuestas por Leite: (i)*SubjectType*, (ii)*ObjectType*, (iii)*VerbType*, (iv)*StateType*

Estas subclases permiten distinguir la naturaleza semántica del símbolo dentro del léxico. Así, el metamodelo no solo registra un término, sino que también establece si este representa un sujeto, un objeto, una acción o un estado del dominio.

Desde esta perspectiva, el modelo define una estructura en la cual cada *LELSymbol* debe estar asociado a una única noción, a un único impacto y a una única clasificación semántica, preservando la forma canónica de definición del LEL.

3.3 Propiedades de la ontología

Las relaciones entre los elementos del metamodelo se expresan mediante un conjunto de propiedades objetuales:

hasNotion: relaciona un *LELSymbol* con su *Notion*.

hasImpact: relaciona un *LELSymbol* con su *Impact*.

hasType: relaciona un *LELSymbol* con su *SymbolType*.

referencesSymbol: permite que una *Notion* o un *Impact* referencien otros símbolos del léxico.

Las tres primeras propiedades modelan la composición estructural de un símbolo. La propiedad *referencesSymbol* introduce la capacidad de construir definiciones interrelacionadas, reflejando la característica del LEL de describir conceptos mediante otros conceptos del mismo universo.

3.4 Organización estructural

La Figura 6 presenta la estructura general de clases del metamodelo. En dicha organización, *LELSymbol* ocupa el rol central y se vincula con *Notion*, *Impact* y *SymbolType*. A su vez, *SymbolType* se descompone en las cuatro categorías semánticas mencionadas.

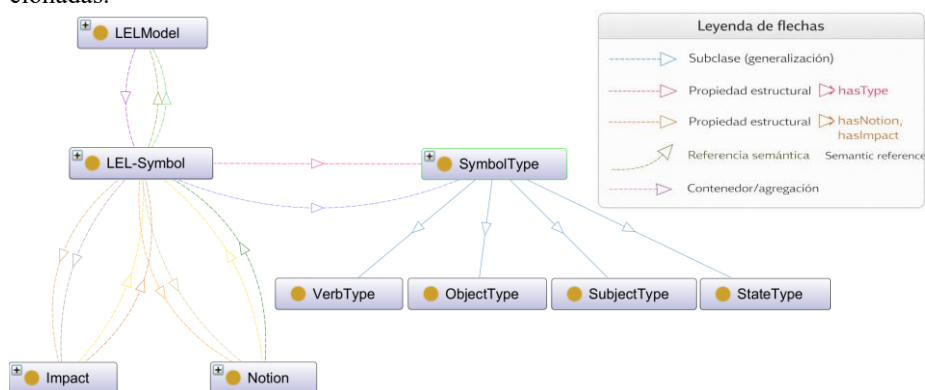


Fig. 6 – LEL Meta Model clases

Esta organización permite representar el LEL como una estructura tipada y no como una lista plana de términos, haciendo explícitas las relaciones y dependencias entre los elementos del léxico.

3.5 Axiomas principales del metamodelo

Las reglas estructurales del LEL fueron formalizadas como axiomas OWL, vinculados explícitamente a las clases del modelo.

Un axioma central establece que cada símbolo debe poseer exactamente una noción, un impacto y un tipo, como se muestra en el Listing 1:

Listing 1. – Restricciones cardinales del metamodelo

```
:LELSymbol rdfs:subClassOf
[ a owl:Restriction ;
  owl:onProperty :hasNotion ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  owl:onClass :Notion
```

```

],
[ a owl:Restriction ;
  owl:onProperty :hasImpact ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  owl:onClass :Impact
],
[ a owl:Restriction ;
  owl:onProperty :hasType ;
  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
  owl:onClass :SymbolType
].

```

Este conjunto de restricciones formaliza la completitud estructural mínima de cada símbolo dentro del léxico.

Otro axioma relevante restringe las referencias semánticas, estableciendo que las nociones y los impactos solo pueden referenciar símbolos definidos en el propio LEL:

Estas restricciones refuerzan la coherencia interna del léxico, evitando referencias externas y promoviendo la construcción de significado dentro del propio universo modelado.

La ontología desarrollada no solo representa los elementos del LEL, sino que explicita sus reglas estructurales de manera verificable. Esto permite tratar el léxico como un artefacto formal, sujeto a validación automática en términos de completitud y consistencia.

Listing 2 – Restricciones de Notion e Impact

```

:Notion rdfs:subClassOf
[ a owl:Restriction ;
  owl:onProperty :referencesSymbol ;
  owl:allValuesFrom :LELSymbol
].

:Impact rdfs:subClassOf
[ a owl:Restriction ;
  owl:onProperty :referencesSymbol ;
  owl:allValuesFrom :LELSymbol
].

```

De este modo, la propuesta transforma al LEL en un modelo estructurado y verificable, ampliando su utilidad más allá de su uso tradicional como herramienta descriptiva.

3.6 Disponibilidad

La especificación completa de la meta-ontología se encuentra disponible en el repositorio del proyecto:

<https://github.com/classeasoft/research-ontolel-airline/blob/main/metamodel/lel-metamodel.ttl>

4 Método propuesto

El método propuesto se estructura como un proceso iterativo que parte del Universo del Discurso (UofD) y culmina en la obtención de conocimiento de dominio validado. Inicialmente, la información del dominio es recolectada a partir de entrevistas, documentos y otras fuentes relevantes, a partir de las cuales se identifican y formulan Kernel Sentences que capturan el conocimiento esencial. Estas sentencias sirven como base para la construcción del Léxico Extendido del Lenguaje (LEL), en el cual cada símbolo es descrito mediante su noción, impacto y tipo semántico. Posteriormente, el LEL es transformado en una ontología OWL conforme al metamodelo propuesto, permitiendo representar formalmente la estructura del dominio. Sobre esta ontología se aplican procesos de inferencia orientados a verificar propiedades como consistencia y satisfactibilidad. En caso de detectarse inconsistencias o limitaciones en la representación, se realiza un proceso de refinamiento del LEL, retroalimentando las etapas previas. Este ciclo iterativo continúa hasta alcanzar una representación consistente, dando lugar a un conjunto de conocimiento de dominio validado. De este modo, el método no solo sistematiza la construcción del LEL, sino que incorpora mecanismos formales de verificación que fortalecen su calidad estructural.

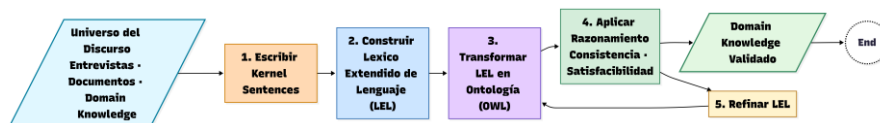


Fig. 4 - Proceso de creación del glosario

El abordaje propuesto consiste en una sucesión de cinco pasos principales:

1. A partir del UofD, reducir los términos de interés a kernel sentences.
2. Construcción del LEL a partir de las kernel sentences, tomando las como base para definir axiomas.
3. Transformación del LEL en una ontología formal utilizando un metamodelo que representan los símbolos del LEL.
4. Aplicación de inferencias como soporte al análisis de requisitos y organización de la base de conocimiento del dominio, habilitando la retroalimentación de nuevos símbolos, donde el reasoner verifica la consistencia semántica de las definiciones.
5. Refinamiento del LEL, con la consistencia chequeada en el paso anterior se pone en evidencia los símbolos que necesitan de ajuste, sea descriptivo o sea semántico modificando o creando nuevos axiomas.

Cada paso utiliza como entrada el resultado del paso anterior, reutilizando el conocimiento de forma incremental, como se muestra en la Figura Fig. 4.

Este proceso establece un flujo determinista que inicia con la abstracción del lenguaje natural, continúa con su estructuración semántica y culmina con una validación lógica formal del vocabulario antes de la etapa de especificación de requisitos.

4.1 Escribir Kernel Sentences

En la fase de elicitación, el conocimiento del dominio se obtiene a partir del Universo de Discurso (UofD), compuesto típicamente por entrevistas, normativas y manuales operativos. Estas fuentes suelen presentar ambigüedades propias del lenguaje natural, tales como oraciones subordinadas, elipsis, voz pasiva o jerga implícita. Si este discurso se incorpora directamente a un modelo conceptual, existe un alto riesgo de introducir inconsistencias semánticas.

Para reducir este problema, el método propone un preprocesamiento lingüístico basado en la extracción de Oraciones Nucleares (Kernel Sentences). Esta técnica descompone el discurso complejo en afirmaciones simples, activas y declarativas con la estructura morfosintáctica: [Sujeto] + [Verbo] + [Objeto].

Ejemplo 4. – Ejemplo Oración original

“El ticket de embarque será emitido por el sistema automatizado.”

Ejemplo 5. – Ejemplo Oración nuclear

“El sistema automatizado emite el ticket de embarque.”

4.2 Construir el Léxico Extendido de Lenguaje (LEL)

A partir del conjunto de Kernel Sentences se construye el Léxico Extendido del Lenguaje (LEL). El ingeniero de requisitos identifica los símbolos relevantes del dominio y define para cada uno sus atributos fundamentales: (i) Noción (denotación); describe qué es el símbolo, (ii) Impacto (connotación); describe cómo el símbolo actúa o cómo el sistema actúa sobre él. Durante la redacción del LEL se aplican dos principios metodológicos: el Principio de Circularidad, que promueve el uso de símbolos ya definidos en el propio léxico, y el Principio de Vocabulario Mínimo, que limita la introducción de términos externos al dominio. El resultado es un léxico estructurado que captura el conocimiento del dominio de forma coherente y autocontenida.

4.3 Transformar el LEL en Ontología (OWL)

Por medio de la *Meta-ontología*, con el objetivo de construir su base de conocimiento consistente en esta etapa del método, basta crear los individuos de una ontología de dominio como queda representado en el Listing 3, utilizando el metamodelo del glosario describiendo los símbolos LEL como individuos dentro de las estructuras discutidas.

Listing 3. – Símbolo LEL Pasajero, usando la Meta-Ontología[18]

```
:AirlineLELModel a lel:LELModel .  
:Pasajero_LEL a lel:LELSymbol ;  
  lel:symbolName "Pasajero"^^xsd:string ;  
  lel:hasType :subjectType ;  
  lel:hasNotion :Pasajero_Notion ;  
  lel:hasImpact :Pasajero_Impact .  
:AirlineLELModel lel:hasSymbol :Pasajero_LEL .
```

```

:Pasajero_Notion a lel:Notion ;
  lel:text "Persona física que ha adquirido un Billete Aéreo, obteniendo el derecho contractual a ser transportada en una Aeronave entre Aeropuertos, de acuerdo con las condiciones definidas por la Compañía Aérea y la Clase de Tarifa."@es;
  lel:referencesSymbol :Aeronave_LEL, :BilleteAereo_LEL, :ClaseDeTarifa_LEL, :CompaniaAerea_LEL .

:Pasajero_Impact a lel:Impact ;
  lel:text "Adquisición del Billete Aéreo y Pago de la Tarifa Aérea. Elección de la Clase de Tarifa. Realización del proceso de Check-in. Embarque en el Vuelo. Ocupación de un Asiento asignado en la Aeronave. Gestión y Entrega de Equipaje al Miembro de Operación."@es;
  lel:referencesSymbol :Aeronave_LEL, :Asiento_LEL, :BilleteAereo_LEL, :ClaseDeTarifa_LEL, :Equipaje_LEL, :MiembroDeOperacion_LEL, :TarifaAerea_LEL, :Vuelo_LEL .

```

4.4 Aplicar razonamiento garantizando consistencia y satisfactibilidad.

El tercer paso consiste en aplicar inferencias sobre la ontología.

Una vez poblada la TBox con axiomas y la ABox con escenarios simulados, se ejecuta un motor de razonamiento (por ejemplo, HerMiT).

Los principales servicios de inferencia son:

(i) Verificación de Satisfactibilidad

Comprueba que las clases no sean conjuntos vacíos producto de contradicciones.

Ejemplo: Si un Objeto intenta ejecutar un verbo cuyo dominio es exclusivo de Sujetos, el sistema detecta inconsistencia automáticamente.

(ii) Verificación de Consistencia Global

Evalúa si: $K \neq \perp$

Es decir, que la base de conocimiento no deriva contradicción lógica.

4.5 Refinar el Léxico Extendido de Lenguaje LEL.

Una vez formalizado en OWL, el conocimiento capturado en el Universo del Discurso (UoFD) pasa a representarse como una base de conocimiento semántica que permite verificar la consistencia del vocabulario del dominio. A medida que el léxico evoluciona, los expertos pueden proponer nuevos símbolos o modificar los existentes, reflejando así el entendimiento progresivo del dominio.

Antes de incorporar estos cambios al modelo, se realiza una validación automática basada en las restricciones definidas en la ontología. Este proceso permite evaluar si las nuevas definiciones son coherentes con la estructura del metamodelo.

Como resultado de esta validación, pueden presentarse dos situaciones. Por un lado, si la propuesta es consistente, el símbolo es incorporado al léxico, ampliando el vocabulario del dominio. Por otro lado, si se detectan inconsistencias, el modelo identifica el conflicto, permitiendo que los ingenieros de requisitos y los expertos revisen y ajusten la definición propuesta.

De esta manera, el proceso de construcción del LEL se desarrolla de forma iterativa, integrando retroalimentación continua y mecanismos de validación que contribuyen a preservar la coherencia del modelo a lo largo de su evolución.

5 Prueba de Concepto

La implementación del método propuesto requiere una infraestructura capaz de representar formalmente el vocabulario del dominio, instanciar las reglas lógicas definidas en el metamodelo y ejecutar mecanismos de inferencia automática. Esta sección describe la arquitectura experimental utilizada para demostrar la viabilidad del enfoque.

Es importante destacar que las tecnologías empleadas constituyen únicamente el soporte de implementación del método. La contribución científica de este trabajo reside en el proceso sistemático de validación temprana del vocabulario del dominio, mientras que las herramientas utilizadas permiten operacionalizar dicho proceso en un entorno automatizado.

5.1 Fundamentos Tecnológicos

La solución se basa en estándares de la Web Semántica definidos por el W3C. En particular, se utiliza el lenguaje OWL para representar formalmente el metamodelo del LEL mediante clases, propiedades y restricciones. Esta representación permite estructurar el vocabulario del dominio de manera explícita y habilitar su validación automática mediante herramientas de razonamiento.

5.2 Instanciación del caso de estudio

El metamodelo fue aplicado a un caso de estudio en el dominio de una línea aérea comercial. A partir de entrevistas con expertos, se identificaron símbolos como Pasajero, Aeronave y Vuelo, los cuales fueron modelados en la ontología junto con sus nociones, impactos y relaciones [18].

La representación ontológica permite verificar automáticamente la coherencia de las definiciones. Por ejemplo, si un analista asigna una acción a una entidad que no corresponde a su tipo semántico, el modelo detecta la inconsistencia, facilitando su corrección.

5.3 Escalabilidad

Para asegurar la escalabilidad en dominios extensos, la base de conocimiento puede modularizarse en ontologías separadas, desacoplando el metamodelo LEL para favorecer su mantenibilidad. Aunque esta prueba de concepto empleó Protégé en configuración standalone, los escenarios industriales y colaborativos exigen resolver desafíos de versionado, sincronización y rendimiento del razonamiento. Esto requerirá la adopción de arquitecturas semánticas distribuidas (ej. Servicios que usan Framework Apache Jena y repositorios RDF) capaces de soportar mayores volúmenes de conocimiento y procesos complejos.

6 Trabajos relacionados

En el ámbito de la extracción y modelado automatizado del conocimiento, Antonelli, Delle, Adorno et al. (2023) proponen un enfoque para obtener modelos conceptuales a partir de especificaciones escritas en lenguaje natural mediante técnicas de Procesamiento de Lenguaje Natural (NLP) e Inteligencia Artificial (IA)[16]. El método utiliza algoritmos probabilísticos para identificar entidades y relaciones en textos no estructurados. No obstante, este enfoque parte de la existencia previa de especificaciones redactadas por analistas. Además, los modelos de lenguaje utilizados en IA presentan una naturaleza probabilística que puede introducir ambigüedades en la extracción semántica.

La propuesta presentada en este trabajo se sitúa en una etapa del proceso de elicitación. En lugar de extraer modelos conceptuales a partir de textos extensos, el método utiliza Kernel Sentences para estructurar el discurso y transforma las definiciones del Léxico Extendido del Lenguaje (LEL) en una representación basada en Lógica de Descripción (DL). De esta manera, la validación del vocabulario del dominio se realiza mediante inferencia lógica, lo que permite verificar la consistencia de los símbolos durante su definición.

Por otra parte, la literatura sobre Ontology Driven Software Development (ODSD), representada por Pan, Staab, Aßmann, Ebert y Zhao (2011), propone el uso de ontologías como artefacto central a lo largo de todo el ciclo de vida del software, desde la captura de requisitos hasta el diseño y la generación de código. Este enfoque busca proporcionar una semántica común para los diferentes artefactos del desarrollo. Sin embargo, su adopción puede requerir conocimientos avanzados en ingeniería ontológica por parte de los equipos de desarrollo [9]. En contraste, el método propuesto en este trabajo utiliza la ontología de manera limitada al proceso de validación del vocabulario. La ontología (TBox y ABox) se emplea como un mecanismo de verificación para detectar inconsistencias en las definiciones del LEL, mientras que los analistas continúan trabajando con representaciones en lenguaje natural estructurado.

Litvak, Antonelli y Rossi (2018) presentaron un proceso focalizado en identificar y resolver conflictos semánticos durante la construcción colaborativa del LEL, utilizando equipos distribuidos [6]. El estudio aborda el problema humano y de gestión en equipos modernos, reconociendo explícitamente que la definición del vocabulario genera discrepancias de visión. La resolución depende de revisiones manuales, listas de verificación (checklists) y el consenso humano. Vemos como oportunidad, proponer una técnica lógico-matemática que ofrezca un apoyo determinista sobre la existencia de una contradicción en las definiciones. Diferenciador de la propuesta: A diferencia de los métodos expuestos, nuestra propuesta utiliza de manera nativa el meta-modelo ontológico para inyectar una semántica formal estricta directamente en el proceso de elicitación del vocabulario. Al separar los conceptos (TBox) de los hechos temporales (ABox), logramos una verificación determinista, automática e incremental de las definiciones del dominio antes de redactar cualquier requerimiento.

En síntesis, mientras que Antonelli et al. abordan la extracción automática de modelos a partir de documentos y Pan et al. proponen un enfoque basado en ontologías a lo largo del ciclo de vida del software, el método presentado introduce un mecanismo

de validación durante la construcción del léxico del dominio. El metamodelo ontológico basado en Lógica de Descripción, LEL y AOM se utiliza para verificar la consistencia del vocabulario durante la elicitación, proporcionando una base formal para las fases posteriores de especificación y modelado.

7 Conclusiones

La calidad de un sistema de software depende en gran medida de la precisión del conocimiento capturado durante las primeras etapas del proceso de desarrollo [3][14]. Este trabajo demuestra que el conocimiento del Universo de Discurso (UofD), expresado en lenguaje natural, puede representarse mediante una estructura formal basada en ontologías. A través de la formalización del Léxico Extendido del Lenguaje (LEL), el enfoque propuesto permite validar la consistencia del vocabulario del dominio mediante mecanismos de verificación automática. De este modo, la validación del conocimiento durante la elicitación de requisitos puede apoyarse tanto en revisiones humanas como en mecanismos automatizados.

La técnica puede interpretarse como una aplicación del paradigma Shift-Left a la calidad semántica. En lugar de detectar inconsistencias en etapas posteriores del desarrollo, el método permite analizar el vocabulario del dominio desde la fase de elicitación, incorporando validaciones tempranas que contribuyen a mejorar la coherencia del modelo.

El enfoque presentado aún no ha sido evaluado en contextos industriales ni en entornos colaborativos con múltiples analistas y expertos del dominio. Asimismo, este trabajo no aborda la transformación automática de Kernel Sentences en definiciones del LEL. En la presente investigación, la construcción del léxico se realizó de forma manual con el objetivo de validar el modelo conceptual y los mecanismos de verificación propuestos.

Sin embargo, se puede afirmar que el método propuesto, basado en un metamodelo ontológico, proporciona mecanismos de validación formal para el LEL, permitiendo verificar la consistencia del vocabulario del dominio desde etapas tempranas del ciclo de vida del software.

Referencias

1. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
2. Breitman, K. K., & Leite, J. C. S. P. (2003). Ontology as a Requirements Engineering Product. *Proceedings of the 11th IEEE International Requirements Engineering Conference (RE'03)*, 309–319. IEEE Computer Society.
3. Brooks, F. P. (1987). No silver bullet: Essence and accidents of software engineering. *Computer*, 20(4), 10–19.
4. Guo, W., Li, J., & Jiang, Z. (2021). Automatically detecting the conflicts between software requirements based on finer semantic analysis. *arXiv preprint arXiv:2103.02255*.

5. Leite, J. C. S. P., & Franco, A. P. M. (1993). A strategy for conceptual model acquisition. *Proceedings of the First IEEE International Symposium on Requirements Engineering*, 243–246. IEEE.
6. Litvak, C., Antonelli, L., & Rossi, G. (2018). Improving the identification of conflicts in collaborative requirements engineering. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 872–877. IEEE.
7. Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL Query Language for RDF. W3C Recommendation. W3C.
8. Leite, J. C. S. P. (1996). Léxico extendido del lenguaje y escenarios del sistema nacional para la obtención de pasaportes. Documento de trabajo, Universidad de Belgrano, Buenos Aires.
9. Pan, J., Staab, S., Aßmann, U., Ebert, J., & Zhao, Y. (2011). *Ontology Driven Software Development*. Springer.
10. Antonelli, L., Rossi, G., & Oliveros, A. (2016). A Collaborative Approach to Describe the Domain Language through the Language Extended Lexicon. *Journal of Object Technology*, 15(3), 3:1–3:27. doi:10.5381/jot.2016.15.3.a3.
11. Gómez-Pérez, A., Corcho, O., & Fernández-López, M. (2004). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer.
12. Maedche, A., Motik, B., Stojanovic, L., Studer, R., & Volz, R. (2003). *Ontologies for Enterprise Knowledge Management*. IEEE Computer Society.
13. Antoniou, G., & van Harmelen, F. (2004). *A Semantic Web Primer* (2nd ed.). MIT Press. 13
14. IEEE. (2026). *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Disponible en: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
15. Hein, A. M. (2010). "Identification and Bridging of Semantic Gaps in the Context of Multi-Domain Engineering", *Abstracts of the 2010 Forum on Philosophy, Engineering & Technology*, pp. 57-5
16. Antonelli, L., Delle V., J. & Adorno, M. & Ballestero, Letizia & Cecconato, Santiago & Fernandez, Alejandro & Maclen, Gregorio & Maltempo, Giuliana & Mattei, Juliana & Tanevitch, Luciana & Torres, Diego. (2023). An Approach to Extract a Conceptual Model from Natural Language Specifications Using NLP and IA. 10.29327/1298356.26-12.
17. Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley
18. Alcântara, M. (2026). *OntoLEL Airline Example*. GitHub repository. Available at: <https://github.com/classeasoft/research-ontolel-airline>
19. Stanford Center for Biomedical Informatics Research, Protégé Ontology Editor and Knowledge Acquisition System. Available: <https://protege.stanford.edu>
20. Fowler, M. (1997). *Analysis Patterns: Resusable Object Models*. Addison-Wesley.
21. G. Hadad, J. Doorn, and G. Kaplan, "Explicitar Requisitos del Software usando Escenarios," in *Proceedings of the 12th Workshop on Requirements Engineering (WER'09)*, 2009