Metodología de interoperabilidad basada en un lenguaje de representación unificado

Roy Arturo Mendieta Zuniga

Tesis depositada en cumplimiento parcial de los requisitos para el grado
de Doctor en
Ciencia y Tecnología Informática

Universidad Carlos III de Madrid

Directores:

Prof. Dr. Juan Llorens Morillo

Prof. Dr. Jose María Álvarez Rodríguez

Enero 2021

Esta tesis se distribuye bajo licencia “Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**”.

Dedicatoria

A Valentina Isabel, hija mía: "Per aspera ad astra".

Agradecimientos

Son muchas las personas que de una u otra manera han contribuido a la realización de esta investigación, a continuación una mención a algunas de ellas:

En primer lugar a Dios, porque considero que de Él vienen todas las oportunidades.

A mi esposa Pamela Nicole Simón Sandoval y a mi hija Valentina Isabel Mendieta, por su paciencia, apoyo e inspiración.

A mis padres, que a la distancia, sé que disfrutan de este logro, ellos en gran medida son responsables del deseo de lograr esta tesis.

Al equipo de *The Reuse Company* por todo su apoyo en la realización de esta tesis, por el compañerismo y conocimientos compartidos.

A mis directores Juan Llorens Morillo y José María Álvarez, por compartir tanto conmigo, por su paciencia, liderazgo y cariño, no tengo palabras, de verdad gracias por tanto.

A Eduardo Cibrian por todo el apoyo recibido.

*a todos mil
gracias.*

Contenidos publicados y presentados

Artículos en revistas indexadas

Título:	<i>Enabling System Artefact Exchange and Selection through a Linked Data Layer</i>
Autores:	Jose María Alvarez-Rodríguez, Roy Mendieta, Jose Luis de la Vara, Anabel Fraga y Juan Llorens
Revista:	<i>Journal of Universal Computer Science</i>
Año:	2018
Vol, Issue:	Vol 24, Issue: 12
Factor de impacto:	2018 JCR- Factor de impacto: 0,701, <i>Computer Science, Software Engineering</i> , Q4
DOI/URL:	http://dx.doi.org/10.3217/jucs-024-11-1536
Referencia:	[1]
Contribución:	Parcial
Declaración:	Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título:	<i>Semantic Recovery of Traceability Links between System Artifacts</i>
Autores:	Jose María Alvarez-Rodríguez, Roy Mendieta, Valentín Moreno, Miguel Sánchez Puebla y Juan Llorens
Revista:	<i>International Journal of Software Engineering and Knowledge Engineering</i>
Año:	2020
Vol, Issue:	Vol 30, Issue: 10
Factor de impacto:	2019 JCR- Factor de impacto: 0,886, <i>Computer Science, Software Engineering</i> , Q4
DOI/URL:	https://doi.org/10.1142/S0218194020400197
Referencia:	[2]
Contribución:	Parcial
Declaración:	Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Conferencias

Título:	<i>Towards Effective SysML Model Reuse</i>
Autores:	Roy Mendieta, José Luis de la Vara, Juan Llorens y Jose María Alvarez-Rodríguez
Conferencia:	<i>International Conference on Model-Driven Engineering and Software Development (MODELSWARD)</i>
Fecha:	2017
Lugar:	Oporto, Portugal
URL:	http://hdl.handle.net/10016/24441
Declaración:	Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título:	<i>Assessment of the Quality of Safety Cases: A Research Preview</i>
Autores:	Jose Luis de la Vara, Gabriel Jiménez, Roy Mendieta y Eugenio Parra.
Conferencia:	<i>Requirement Engineering: Foundation for Software Quality (REFSQ)</i>
Fecha:	2019
Lugar:	Alemania
URL:	https://link.springer.com/chapter/10.1007/978-3-030-15538-4_9
Declaración:	Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título:	<i>Integration of Tool Support for Assurance and Certification and for Knowledge-Centric Systems Engineering</i>
----------------	--

Autores: Jose Luis de la Vara, Eugenio Parra , Luis Alonso, Roy Mendieta, Borja Lopez and Jose Alvarez-Rodríguez.

Conferencia: *2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*

Fecha: 2019

Lugar: Alemania

URL: <https://ieeexplore.ieee.org/document/8990220>

Declaración: Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título *Advances in Artefact Quality Analysis for Safety-Critical Systems*

Autores: Eugenio Parra, Luis Alonso, Roy Mendieta y Jose Luis de la Vara.

Conferencia: *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*

Fecha: 2019

Lugar: Berlin, Alemania

URL: <https://ieeexplore.ieee.org/abstract/document/8990290>

Declaración: Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título: *Elevating the meaning of data and operations within the development lifecycle through an interoperable toolchain*

Autores: Jose María Alvarez-Rodríguez , Roy Mendieta Zuñiga y Juan Llorens

Conferencia: *INCOSE International Symposium*
Fecha: 2019
Lugar: Orlando, USA
URL: <https://doi.org/10.1002/j.2334-5837.2019.00652.x>
Declaración: Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título: *Challenges and opportunities in the integration of the Systems Engineering process and the AI/ML Lifecycle*
Autores: Jose María Álvarez-Rodríguez, Roy Mendieta, Valentín Moreno y Juan Llorens
Conferencia: *INCOSE International Symposium*
Fecha: 2019
Lugar: Orlando, USA
URL: <https://doi.org/10.1002/j.2334-5837.2019.00621.x>
Declaración: Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Título: *Enabling system artifacts reuse through the semantic representation of engineering models: a case study of Simulink models*
Autores: Roy Mendieta, Eduardo Cibrián, Jose María Álvarez-Rodríguez y Juan Llorens
Conferencia: *The Fourteenth International Conference on Advances in Semantic Processing SEMAPRO 2020*
Fecha: 2020

Lugar: Nice, Francia

URL: http://www.thinkmind.org/index.php?view=article&articleid=semapro_2020_2_20_30010

Declaración: Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita

Otros méritos de investigación

Proyectos europeos

Título	AMASS- AMASS-ARCHITECTURE DRIVEN MULTICON-CERN ASSURANCE
Entidad financiadora	H2020-ECSEL-2015-1-RIA-TWO-STAGE
Referencia	URL: https://cordis.europa.eu/project/rcn/202642_en.html
Tipo de convocatoria	Europea
Coordinador/	TECNALIA R&I/DR. JOSE LUIS DE LA VARA
Participantes	29 SOCIOS ENTR ELLOS UNIVERSIDAD CARLOS III
Duración	Desde: 01/04/2016 Hasta: 31/03/2019 N° total de meses: 36
Contribución	Durante la ejecución de este proyecto fue necesaria la interconexión entre la plataforma AMASS con la herramienta Requirement Quality Analyzer v18, para el desarrollo de dicha interconexión se utilizaron elementos de la metodología propuesta en esta tesis.

Título	ARROWHEAD TOOLS FOR ENGINEERING OF DIGITALISATION SOLUTIONS (AHTOOLS)
Entidad financiadora	H2020-ECSEL-2018-2-IA-TWO-STAGE
Referencia	826452
Tipo de convocatoria	Europea
Coordinador/	LULUA UNIVERSITY OF TECHNOLOGY,
Investigador Principal	JOSE MARÍA ALVAREZ RODRÍGUEZ
Participantes	88 SOCIOS ENTRE ELLOS UNIVERSIDAD CARLOS III
Duración	Desde: 01/05/2019 Hasta: 31/08/2022 N° total de meses: 39
Contribución	Se utilizó la solución propuesta, para interconectar herramientas de modelado físico que eran necesarias en este proyecto.

Proyectos regionales

Título	MADRID FLIGHT ON CHIP
Entidad financiadora	COMUNIDAD DE MADRID-Proyectos Sinérgicos de I+D (2018)
Referencia	N/A
Tipo de convocatoria	REGIONAL
Coordinador/	SENER
Investigador Principal	JUAN LLORENS (UC3M)
Participantes	SENER, UC3M y otros
Duración	Desde: 01/01/2019 Hasta: 31/12/2021 N° total de meses: 36
Contribución	La metodología de interoperabilidad desarrollada en esta tesis será utilizada para interconectar múltiples herramientas que participan en procesos de ingeniería.

Proyectos nacionales

Título	MADRID FLIGHT ON CHIP
Entidad financiadora	COMUNIDAD DE MADRID-Proyectos Sinérgicos de I+D (2018)
Referencia	N/A
Tipo de convocatoria	REGIONAL
Coordinador/	SENER
Investigador Principal	JUAN LLORENS (UC3M)
Participantes	SENER, UC3M y otros
Duración	Desde: 01/01/2019 Hasta: 31/12/2021 N° total de meses: 36
Contribución	La metodología de interoperabilidad desarrollada en esta tesis será utilizada para interconectar múltiples herramientas que participan en procesos de ingeniería.

Resumen

Los costes económicos asociados a los problemas de interoperabilidad son bastante conocidos, aunque actualmente existen alternativas para mejorar los niveles de interoperabilidad, los desafíos de interoperabilidad siguen vigentes, prueba de ello es la prioridad que da la Unión Europea a reforzar su estrategia de interoperabilidad, esto debido que la misma es fundamental en la nueva estrategia digital de la Comisión Europea.

En el contexto del proceso de ingeniería, la interoperabilidad juega un papel crítico debido a la necesidad intrínseca de conectar diferentes organizaciones, personas, métodos de ingeniería y herramientas provenientes de múltiples disciplinas. El conocimiento incrustado en los diferentes artefactos del sistema debe gestionarse bajo diferentes protocolos de comunicación, elaboración de formatos y lenguajes. Esta situación hace que sea extremadamente difícil unificar los entornos de ingeniería y explotar los datos, la información y el conocimiento generado durante el ciclo de vida del desarrollo.

En este trabajo de investigación, se define, implementa y valida una metodología para unificar el acceso, intercambio y explotación de artefactos del sistema. “*System Representation Language (SRL)*” se especifica como un metamodelo abierto para representar cualquier tipo de contenido de artefactos del sistema. Luego, se presenta una arquitectura basada en conectores SRL como un medio para interconectar herramientas dentro de una cadena de herramientas y para proporcionar capacidades de reutilización de artefactos del sistema, la gestión de la calidad y la trazabilidad. Además, también se describe un proceso iterativo para desarrollar conectores SRL. Una vez definidas las bases teóricas y el marco tecnológico, se realizan algunos experimentos científicos para validar el enfoque presentado como método: 1) para mejorar el grado de interoperabilidad dentro de un entorno de cadena de herramientas y 2) para mostrar su aplicabilidad para brindar funciones que requieran Visión holística del sistema como reutilización, recuperación de trazabilidad o control de calidad.

Finalmente, como resultado técnico de este trabajo de investigación, se introduce la

integración con un conjunto de herramientas comerciales para validar la metodología y la implementación con casos de uso del mundo real.

Palabras clave: Interoperabilidad, representación de conocimientos, desarrollo basado en modelos, gestión de conocimientos, trazabilidad, calidad, verificación y validación.

Abstract

The costs associated with interoperability are well-known. Although it is possible to find different alternatives, the implementation of a interoperability strategy still remains challenging. As an example, the European Union is looking for boosting its interoperability strategy as a driver for the new digital agenda established by the European Commission.

In the context of the engineering process, interoperability plays a critical role due to the intrinsic necessity of connecting different organizations, people, engineering methods and tools coming from multiple disciplines. The embedded knowledge in the different system artifacts must be managed under different communication protocols, formats and languages making. This situation makes extremely difficult to unify the engineering environments and to exploit the data, information and knowledge generated during the development lifecycle.

In this research work, a methodology to unify the access, exchange and exploitation of system artifacts is defined, implemented and validated. The “*System Representation Language (SRL)*” is specified as an open meta-model to represent any type of system artifact content. Then, an architecture based on SRL connectors is presented as a mean to interconnect tools within a toolchain and to provide capabilities for system artifact reuse, quality management and traceability. Furthermore, an iterative process to develop SRL connectors is also described. Once the theoretical definitions and the technological framework is defined, some scientific experiments are conducted to validate the presented approach as a method: 1) to improve the degree of interoperability within a toolchain environment and 2) to show its applicability to provide functions that require a holistic view of the system such as reuse, traceability recovery or quality checking.

Finally, as a technical outcome of this research work, the integration with a suite of commercial tools is introduced to validate the methodology and the implementation with real-world use cases.

Key words:Interoperability, knowledge representation, model-based development, knowledge management, traceability, quality, verification and validation.

Índice

Agradecimientos	V
Contenidos publicados y presentados	VI
Artículos en revistas indexadas	VI
Conferencias	VII
Otros méritos de investigación	X
Proyectos europeos	X
Proyectos regionales	XI
Proyectos nacionales	XI
Resumen	XII
Índice de Tablas	XX
Índice de Figuras	XXII
1 Introducción	1
1.1 Contexto de la investigación	1
1.2 Descripción del problema	2
1.3 Motivación	4
1.4 Hipótesis	6
1.5 Objetivos	7
1.5.1 General	7
1.5.2 Específicos	7

1.6	Metodología	8
1.6.1	Plan de trabajo	9
1.6.2	Estructura del documento	10
2	Estado del Arte	11
2.1	Interoperabilidad	12
2.1.1	Niveles de interoperabilidad	13
2.1.2	Interoperabilidad en la cadena de suministros	17
2.2	Meta Object Facility (MOF)	17
2.3	Trazabilidad	19
2.4	Modelos	21
2.5	Gestión de Conocimiento	23
2.6	Representación del conocimiento	24
2.6.1	Ontologías formales	25
2.6.2	Datashapes	26
2.6.3	Modelo de representación de conocimientos aplicado en la tesis	28
2.7	Métodos de evaluación sobre nivel de madurez de interoperabilidad en las organizaciones	29
2.7.1	El modelo de Niveles de Interoperabilidad de Sistemas de Informa- ción (LISI)	30
2.7.2	Modelo de madurez de interoperabilidad organizacional	33
2.7.3	The Layered Interoperability Score (i-Score)	35
2.7.4	Service Measurement Index	36
2.7.5	Método de evaluación común para estándares y especificaciones (CAMSS)	37
2.7.6	Evaluación de la madurez de la interoperabilidad semántica de un servicio público (SIMAPS)	38
2.8	Trabajos relacionados	38
2.8.1	Trabajos relacionados con interoperabilidad general	38
2.8.2	Trabajos relacionados con interoperabilidad en ingeniería de sistemas	50

2.9	Conclusiones del estado del arte	57
3	Definición de métodos para la interoperabilidad en procesos de ingeniería	61
3.1	Descripción de los conectores SRL	62
3.2	Paradigma del marco de interoperabilidad para procesos de ingeniería . . .	64
3.3	Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería	65
3.3.1	Información fuente	67
3.3.2	Modelos SRL	68
3.3.3	Especificación SRL	70
3.3.4	Interpretación de modelos SRL	71
3.4	Definición del proceso para la implementación de conectores SRL	78
3.4.1	Análisis	79
3.4.2	Preparación y diseño	81
3.4.3	Desarrollo de conector SRL	83
3.4.4	Integración	84
3.4.5	Resumen de metodología de desarrollo de conectores SRL	85
3.5	Conclusiones sobre el marco de interoperabilidad para procesos de ingeniería	87
4	Implementación de métodos para la interoperabilidad en procesos de in- geniería	88
4.1	Arquitectura tecnológica del marco propuesto	88
4.1.1	API de fuentes de información	89
4.1.2	Conector SRL	90
4.1.3	SRL2Cake	91
4.1.4	SRL2RDF	92
4.1.5	CAKE API	93
4.2	Tipos de conectores SRL en función de sus posibilidades tecnológicas	94
4.2.1	Conectores basados en APIs	95
4.2.2	Conectores basados en lectura de archivos	96

4.2.3	Conexiones SRL	97
4.2.4	Ejemplo de integración de metodología de interoperabilidad con herramientas TRC	98
4.3	Descripción de casos de uso de implementación del marco de interoperabilidad para procesos de ingeniería en herramientas	100
4.3.1	Detalle de fabricantes que demandan integración de sus herramientas bajo el marco propuesto	100
4.3.2	Caso de uso industrial del marco de interoperabilidad para procesos de ingeniería en la herramienta Knowledge Manager	103
4.3.3	Caso de uso industrial de implementación del marco de interoperabilidad para procesos de ingeniería en la herramienta Traceability Studio	104
4.3.4	Caso de uso industrial del marco de interoperabilidad para procesos de ingeniería en la herramienta RQA - QUALITY Studio	106
4.4	Conclusiones sobre implementación de métodos para la interoperabilidad en procesos de ingeniería	108
5	Experimentación: evaluación y análisis de resultados	110
5.1	Descripción de la validación	111
5.2	Consideraciones generales	111
5.3	Experimento sobre evaluación del grado de interoperabilidad en un entorno de ingeniería	113
5.3.1	Objetivo	114
5.3.2	Diseño	114
5.3.3	Ejecución del experimento	116
5.3.4	Análisis de resultados del experimento	118
5.3.5	Validación Hipótesis H1	118
5.4	Experimento sobre reutilización de modelos lógicos y físicos	119
5.4.1	Objetivo	119
5.4.2	Diseño	119

5.4.3	Ejecución del experimento	120
5.4.4	Análisis de resultados del experimento	126
5.4.5	Validación Hipótesis H2	126
5.5	Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas	127
5.5.1	Objetivo	128
5.5.2	Diseño	128
5.5.3	Ejecución del experimento	130
5.5.4	Análisis de resultados del experimento	135
5.5.5	Validación Hipótesis H3	142
5.6	Validación industrial del marco de interoperabilidad para procesos de ingeniería	142
6	Conclusiones y futuras líneas de investigación	144
6.1	Conclusiones científicas	144
6.2	Conclusiones tecnológicas	146
6.3	Futuras líneas de investigación	147
6.3.1	Futuras líneas de investigación científicas	147
6.3.2	Futuras líneas de investigación tecnológicas	148
	Bibliografía	149

Índice de Tablas

1-1	Detalle de cumplimiento de objetivos por capítulo de la tesis.	8
2-1	Ejemplo MOF para SRL, UML y OSLC.	29
2-2	“Organizational interoperability maturity model for c2” [3].	34
3-1	Descripción de los elementos del metamodelo SRL	72
3-2	Ejemplo de equivalencia entre elementos de Simulink y SRL.	73
3-3	Resumen de metodología de desarrollo de conector SRL.	86
4-1	Equivalencia entre elementos SRL y elementos CAKE.	91
4-2	Equivalencia entre componentes OWL y SRL.	93
4-3	Listado de fabricantes que demandan integración de sus herramientas al marco propuesto en el ecosistema de TRC.	100
4-4	Detalle de conectores SRL actualmente implementados KM, RQA y Traceability Studio.	102
5-1	Detalle de validación por hipótesis.	112
5-2	Tabla de niveles de evaluación de métricas de precisión y Recall [4].	112
5-3	Matriz utilizada definir los elementos necesarios para el cálculo de Precisión, Recall y F_1	113
5-4	Matriz de nivel de sofisticación de interoperabilidad (IM).	114
5-5	Criterios para definición del valor numérico de LV.	115
5-6	Matriz de sofisticación de interoperabilidad sin utilizar la solución propuesta IM_x	117

5-7	Matriz de sofisticación de interoperabilidad utilizando la solución propuesta IM _y	117
5-8	Resultados de la recuperación de modelos lógicos con SRL, Papyrus e IBM Rhapsody [1].	121
5-9	Resultados de la recuperación de modelos físicos con SRL y Simulink [1]. .	123
5-10	Consultas para recuperación de modelos lógicos [1].	124
5-11	Consultas para recuperación de modelos físicos [1].	125
5-12	Listado de requisitos de EIRENE utilizados para realizar la experimentación [2].	130
5-13	Elementos de la estructura de desglose de productos en Rational Rhapsody [2].	136
5-14	Métricas de Precision, Recall y F1 resultado de la ejecución de algoritmos de descubrimiento de trazas, usando búsquedas basadas en texto y búsquedas basadas en semántica, sobre elementos de la especificación de requisitos contra otros elementos de la misma especificación de requisitos [2]. . . .	137
5-15	Prueba de hipótesis estadística, la prueba t de dos muestras que asume varianzas desiguales para comparar los procesos de descubrimiento de enlaces de trazabilidad basados en texto y basados en conceptos para métricas de precisión [2].	139
5-16	Prueba de hipótesis estadística para la métrica Recall, la prueba t de dos muestras que asume varianzas desiguales para comparar los procesos de descubrimiento de enlaces de trazabilidad basados en texto y basados en conceptos [2].	139
5-17	Prueba de hipótesis estadística para la métrica F1, la prueba t de dos muestras que asume varianzas desiguales para comparar los procesos de descubrimiento de enlaces de trazabilidad basados en texto y basados en conceptos [2].	140
5-18	Precision, Recall an F1 for Scenario 1 [2].	140

Índice de Figuras

1-1	Ejemplo NASA <i>Safety Critical Environment</i> [5].	4
1-2	Hoja de ruta de digitalización de la ingeniería [6].	6
1-3	Fases para el desarrollo de la tesis.	9
2-1	Niveles de interoperabilidad [7].	14
2-2	Arquitectura de metadatos de MOF [8].	18
2-3	Ejemplo de contexto de trazabilidad [9].	21
2-4	Áreas semánticas de UML [10].	22
2-5	Modelo de representación de información de RSHP [11].	27
2-6	Modelo de madurez de LISI [12].	31
2-7	Atributos PAID [13].	33
2-8	Atributos SMI [14].	36
2-9	Relación entre las soluciones CAMSS y sus componentes publicados [15]. .	37
2-10	Modelo conceptual de EIF [16].	40
2-11	Enfoque metodológico para la definición de EISB y revisión de la situación [17].	42
2-12	Estructura del Marco de Interoperabilidad Empresarial [18].	44
2-13	Conceptos y relaciones de la especificación básica de OSLC [19].	51
2-14	Arquitectura funcional y servicios centrales para la gestión de conocimien- tos basados en la especificación OSLC-KM [20].	52
2-15	Alineación MoSSEC con otros estándares ISO STEP [21].	54
2-16	Entorno integrado de MBSE [22].	56
2-17	Descripción general de LOTAR [23].	57

3-1	Concepto inicial del marco propuesto.	63
3-2	Ejemplo interoperabilidad entre herramientas sin aplicar el marco propuesto.	65
3-3	Ejemplo interoperabilidad entre herramientas aplicando el marco propuesto.	66
3-4	Arquitectura conceptual del marco propuesto.	66
3-5	Ejemplo de representación de información textual en SRL, utilizando un patrón para a generación de grafos semánticos.	68
3-6	Ejemplo de diagrama de Simulink [24].	69
3-7	Metamodelo SRL.	73
3-8	Diseño de Mapeable Elements.	74
3-9	Interacción entre Mapeable Elements y conectores SRL.	75
3-10	Arquitectura de Mapeable Elements.	78
3-11	Metodología de desarrollo de conectores SRL.	79
4-1	Arquitectura tecnológica del marco propuesto.	89
4-2	Conector SRL a Rhapsody.	90
4-3	Arquitectura de SRL2RDF.	92
4-4	Ecosistema de herramientas de TRC.	94
4-5	Arquitectura de conector SRL nativo.	96
4-6	Arquitectura de conector SRL basado en la lectura de archivos.	97
4-7	Ejemplo de conexiones a Simulink.	98
4-8	Integración SRL con herramientas TRC.	99
4-9	Estructura de una ontología en el ámbito de KM [190].	103
4-10	Arquitectura de integración de conectores SRL en KM.	105
4-11	Arquitectura de integración de conectores SRL en Traceability Studio.	107
4-12	Arquitectura de integración de conectores SRL en RQA.	108
5-1	Precisión para cada caso de prueba y enfoque.	138

1 Introducción

“Atreveos: el progreso solamente se logra así”

Víctor Hugo

Este capítulo pretende servir de introducción a la investigación desarrollada en esta tesis. En el mismo se presenta la estructura y planificación que se ha seguido.

Se presenta el contexto en que se realizó la tesis, en donde se detallan los pormenores de las líneas de investigación con las cuales se alinea esta tesis, así como también la estructura organizacional del grupo de investigación y su relación con la industria.

Adicionalmente se detalla tanto la descripción del problema al que se pretende dar solución como la motivación a realizar la investigación, se sustenta la necesidad de dar solución al problema todavía vigente alineado con los objetivos estratégicos de la Unión Europea.

También se definen la hipótesis, objetivo general y objetivos específicos de la investigación, se especifica una tabla que pretende ubicar al lector sobre que capítulo da cumplimiento a que objetivo específico. Se presenta la metodología que se utilizó para llevar a cabo el desarrollo de la tesis, el plan de trabajo que consiste en los principales hitos a cumplir.

1.1. Contexto de la investigación

La presente tesis se desarrolla dentro del grupo Knowledge Reuse Group (KR) de la Universidad Carlos III de Madrid. KR mantiene un contrato con la empresa The Reuse

Company (TRC) que implica el desarrollo del I+D+i por parte de KR en áreas de interés para TRC, siendo esta última un Spin-off del grupo KR, las líneas de investigación de KR están perfectamente alineadas con las necesidades de I+D+i de TRC [25].

Algunas de las líneas de investigación en las que se centra KR, son: la reutilización, representación y recuperación del conocimiento, ingeniería de software e ingeniería de sistemas [26]. Dentro de las áreas de enfoque del KR se encuentran las siguientes:

- Representación y del conocimiento: desarrollo herramientas, estándares para la representación y gestión de conocimientos entre otros.
- Reutilización y plataformas.
- Análisis de redes sociales: desarrollo de plataformas para el aprovechamiento de datos en web social.
- Tecnologías del lenguaje humano: modelado y comprensión del lenguaje.

Siendo TRC una empresa que produce tecnologías para el apoyo a los procesos de trazabilidad, recuperación, gestión del conocimiento, calidad en artefactos de ingeniería, entre otros, la interoperabilidad es un área de mucho interés, debido a la gran cantidad de herramientas que actualmente existen en el mercado para la generación artefactos de información que intervienen en las distintas etapas de la ingeniería de sistemas.

En el contexto descrito anteriormente, es donde se lleva a cabo la presente tesis, la colaboración existente entre TRC y KR permitió el acceso a tecnologías que tienen aproximadamente 20 años en el mercado, apoyando diversos procesos de ingeniería, para clientes en distintos países y de diferentes dominios.

1.2. Descripción del problema

Con el desarrollo de sistemas complejos y la capacidad de las compañías de trabajar en distintos entornos, incluso en distintas ubicaciones geográficas, la cadena de suministros se ha convertido en una red, debido a la cantidad de actores que actualmente se involucran en los distintos procesos de ingeniería de sistemas [27]. Este nuevo contexto, implica la necesidad de procesamiento de múltiples tipos de información con el propósito

1.2 Descripción del problema

de llevar a cabo una cooperación eficaz entre las distintas fuentes de conocimiento.

Contar con la capacidad de entender todo el conocimiento generado en un determinado entorno de producción es un desafío en sí mismo, considerando la gran cantidad de herramientas y conocimiento que fluye durante todo el proceso, la diversidad de formatos de información, protocolos de acceso a esta información así como el ecosistema de tecnologías que conforman cualquier proceso industrial, representa una necesidad vigente y sobre todo muy compleja, es necesario sobrepasar no solamente las barreras tecnológicas que permitan la comunicación entre las fuentes de información sino también contar con un entendimiento semántico del conocimiento, para mejorar la eficiencia en los procesos. Según [28], este entendimiento consiste en la habilidad de dos o más componentes de software de cooperar independientemente de sus lenguajes, interfaces y plataformas de ejecución.

En otras palabras, los sistemas como fabricas o plantas, constan de varios miles de activos, que a menudo son proporcionados por diferentes organizaciones [29]. Estos activos interactúan entre sí para actuar como un sistema unificado. Para ello, los dispositivos y sistemas involucrados deben contar con la capacidad de intercambiar y utilizar mutuamente la información que se ha intercambiado [30]. Una de las dificultades a las que se enfrentan las organizaciones es la falta de este intercambio y utilización de información antes mencionado, con el propósito de gestionar y progresar en sus respectivas actividades.

La Figura 1-1 muestra un entorno de desarrollo de software para *Safety Critical* [31,32] dentro de La Administración Nacional de Aeronáutica y el Espacio (NASA), en donde se puede apreciar que, en las distintas fases del proceso de desarrollo, se ven involucradas múltiples herramientas.

Estas herramientas generan artefactos de conocimiento que contribuyen a los procesos de desarrollo de software, lograr un entendimiento entre todo este ecosistema permitiría automatizar muchas de las actividades a las cuales estas fuentes de información contribuyen, asimismo realizar procesos de trazabilidad, reutilización de artefactos, transformación, entre otros, lo cual permitiría hacer los procesos más eficientes y consistentes.

Figura 1-1: Ejemplo NASA *Safety Critical Environment* [5].

1.3. Motivación

Uno de los aspectos claves de la estrategia digital de la Comisión Europea es la interoperabilidad [33]. Según un comunicado de febrero de 2020 [34] sobre la estrategia del futuro digital de la Unión Europea, la interoperabilidad es esencial para un mayor aprovechamiento de los datos. El comunicado dice literalmente que “Una estrategia reforzada de interoperabilidad de los gobiernos de la UE programada para fines de 2021 será una acción clave para garantizar la coordinación y estándares comunes para servicios y flujos de datos seguros y sin fronteras del sector público” [34].

En vista de lo anterior, las investigaciones realizadas sobre mejoras en la interoperabilidad están alineadas con la visión de la UE.

Adicionalmente, los problemas de interoperabilidad pueden representar costes económicos muy fuertes en las diferentes industrias, por ejemplo, el Departamento Industria y Comercio de los Estados Unidos de América en 2004, estimó que problemas de interoperabilidad representaron un costo de US 15.8 billones [35]. *The West Health Institute* en 2013 estimó pérdidas potenciales de US 30 billones de pérdidas por año, debido a problemas de interoperabilidad en segmentos del Healthcare en Estados Unidos de América [36]. En 2015 PwC estimó que Salud Digital podría ahorrarse 99 billones en costes a la *European*

Union Gross Domestic Product (EUGDP) con mejoras en la interoperabilidad [37].

Otra de las motivaciones a contribuir con las mejoras en la interoperabilidad obedece a la necesidad industrial que nace del hecho de que el desarrollo de la actividad industrial implica el uso de una variedad de herramientas en sus procesos, para gestionar grandes cantidades de datos de ingeniería que se generan a lo largo del ciclo de su actividad [38–41]. Las herramientas individuales suelen estar diseñadas para realizar tareas de ingeniería específicas y se basan en formatos de datos específicos. Esto genera problemas al intentar automatizar tareas de ingeniería que no son compatibles con una herramienta en particular y que requieren datos de varias herramientas [42]. Por lo tanto, lograr un entendimiento entre las distintas herramientas que intervienen en los procesos de las organizaciones, contribuyen a incrementar las posibilidades de automatización de los procesos, estos procesos pueden involucrar trabajo colaborativo de múltiples dominios de la información [43–45].

Finalmente, el Consejo Internacional en Ingeniería de Sistemas (INCOSE) que es una organización dedicada al desarrollo de la ingeniería de sistemas tanto a nivel académico como profesional, mediante el *workshop Future of Systems Engineering (FuSE)*, definió los conceptos, Inteligencia Artificial para Ingeniería de Sistemas (AI4SE) e ingeniería de sistemas para inteligencia artificial (SE4AI) para describir una evolución que está teniendo lugar en la comunidad de ingeniería de sistemas. En [6] se presenta una visión a cinco (5) años que refleja la evolución en la ingeniería de sistemas en la que se adoptará un enfoque basado en el uso de datos y la colaboración mediante modelos, como habilitadores de la actividad de ingeniería. La Figura 1-2 presenta una hoja de ruta que refleja la evolución de la digitalización de la ingeniería. Como se puede apreciar a medida que se avanza en la hoja de ruta, se requiere mayor integración e interpretación de la información.

Partiendo del estado actual, es necesario lograr una integración a nivel de datos, posteriormente es necesario evolucionar esta integración y llevarla hasta un nivel de interpretación semántica, en donde es posible utilizar el conocimiento obtenido de la integración lograda en etapas previas, esto es imperativo para lograr que los datos se vuelvan un habilitador de la actividad de ingeniería. En vista de lo anterior, la interoperabilidad

Figura 1-2: Hoja de ruta de digitalización de la ingeniería [6].

es fundamental para lograr esta visión ya que como se ha mencionado anteriormente el conocimiento se encuentra disperso entre distintas fuentes de información (formatos, protocolos, etc.).

1.4. Hipótesis

Según la problemática abordada en la sección 1.3, en la línea de investigación que se aborda en esta tesis, se plantean la siguiente hipótesis:

Hipótesis 1 (H1): *Mediante una representación unificada de conocimiento es posible mejorar la interoperabilidad.*

Hipótesis 2 (H2): *Mediante la posibilidad de tener un mecanismo de interoperabilidad se puede mejorar la selección de artefactos.*

Hipótesis 3 (H3): *Mediante la posibilidad de tener un mecanismo de interoperabilidad se puede mejorar el descubrimiento de trazas.*

1.5. Objetivos

Los objetivos de la presente tesis doctoral se han dividido en dos partes, por un lado, el objetivo general, donde se presenta de forma resumida el alcance de la investigación presentada en la tesis y la segunda parte que corresponde a los objetivos específicos, en donde se detallan los objetivos que se pretenden alcanzar.

1.5.1. General

Proponer una metodología de interoperabilidad basada en una representación unificada del conocimiento aplicable a cualquier fuente de información:

1.5.2. Específicos

1. Determinar los niveles selección de artefactos basados en un mecanismo de interoperabilidad.
2. Determinar los niveles de descubrimiento de traza basados en un mecanismo de interoperabilidad.
3. Estudiar, analizar y valorar los marcos de interoperabilidad disponibles.
4. Proponer un lenguaje de representación de conocimientos que permita la representación unificada de cualquier fuente de información.
5. Desarrollar y reutilizar la tecnología necesaria para procesar los elementos de conocimiento.
6. Realizar pruebas y validaciones a los componentes implementados para determinar la validez de la metodología propuesta.
7. Difundir el conocimiento generado y las tecnologías implementadas en las comunidades científicas e Industriales.

La Tabla 1-1 muestra cómo se realizará el cumplimiento de los objetivos específicos por cada capítulo del documento.

Tabla 1-1: Detalle de cumplimiento de objetivos por capítulo de la tesis.

Objetivo(s)	Capítulo(s)
3	Estudio y valoración de las alternativas de interoperabilidad presentadas en el Capítulo 2.
4, 5	En los Capítulos 3 y 4, se presenta SRL como alternativa para la representación de los diferentes artefactos de ingeniería, también se propone una arquitectura tecnológica que incluye el detalle de tecnologías desarrolladas y reutilizadas.
1, 2 y 6	Evaluación y análisis de resultados al marco propuesto, en el Capítulo 5.
7	Detalle de publicaciones realizadas durante la actividad doctoral. Proyectos nacionales y europeos relacionados con la tesis en el apartado VI.

1.6. Metodología

El desarrollo de la tesis se llevó a cabo como un proceso estructurado en el cual se busca dar solución a un problema concreto. El problema consiste en tres hipótesis de las que parten objetivos concretos, que se pretenden alcanzar a lo largo del desarrollo de la investigación.

Para el desarrollo de la tesis se definieron una serie de pasos con el propósito de estructurar de forma eficiente el trabajo necesario. Para la definición estos pasos, se consideraron métodos propuestos por [46].

La Figura 1-3 muestra los 5 fases que abarcaron el desarrollo la tesis:

- **Problema:** Se seleccionó un problema en sintonía con las líneas de investigación del Grupo de Knowledge Reuse (KR). Una vez seleccionado el problema se plantearon una serie de preguntas con el propósito de enfocar la investigación, ¿Es posible resolver el problema?, ¿existen alternativas en la comunidad científica para dar solución al problema propuesto?, ¿El problema propuesto sigue siendo un problema

Figura 1-3: Fases para el desarrollo de la tesis.

en la actualidad?.

- **Revisión bibliográfica:** una vez el problema estuvo puntualizado se procedió a hacer una revisión bibliográfica de las posibles soluciones al problema de la investigación, se realizó una valoración de las alternativas propuestas.
- **Hipótesis:** se planteó la hipótesis de investigación y los objetivos generales y específicos que se desarrollarían durante la tesis.
- **Definición de Solución:** se procedió a la definición de una solución al problema planteado, reutilizando elementos identificados en la etapa de la revisión bibliográfica, con componentes nuevos desarrollados durante la investigación.
- **Evaluación:** Se realizó la evaluación de la solución propuestas mediante la elaboración de varios casos de estudio.

1.6.1. Plan de trabajo

El plan de trabajo de la elaboración de esta tesis consiste en tres hitos principales:

1. **Desarrollo de investigación:** en esta etapa se definió el alcance de la investigación, se valoraron los trabajos relacionados, se identificaron elementos a reutilizar en la propuesta final.
 - Definición y alcance.
 - Investigación y experimentación inicial.
 - Elaboración de publicación inicial y presentación en conferencia.
2. **Validación.**

- Elaboración de casos de estudio.
 - Desarrollo de experimentación detallada.
 - Publicación en al menos dos revistas *Journal Citation Reports (JCR)*.
3. Escritura de documento de tesis.
- Escritura de tesis doctoral.
 - Lectura tesis.

1.6.2. Estructura del documento

El presente documento está estructurado en los capítulos que a continuación se detallan:

- Capítulo 1. Introducción: En este capítulo se presenta el contexto en el cual se realizó la investigación, así como también la descripción del problema, hipótesis, objetivos y la metodología utilizada para llevar a cabo la presente tesis.
- Capítulo 2. Estado del arte: Se presenta investigaciones relacionadas con la tesis, así como también los conceptos sobre los cuales se enmarca el trabajo desarrollado.
- Capítulo 3. Definición de métodos para la interoperabilidad en procesos de ingeniería: Se presenta una arquitectura conceptual y una metodología para la implementación del marco propuesto.
- Capítulo 4. Implementación de métodos para la interoperabilidad en procesos de ingeniería: Se presenta una arquitectura tecnológica y casos de uso industriales utilizando el marco propuesto.
- Capítulo 5. Experimentación: evaluación y análisis de resultados: Se detallan los experimentos realizados para validar el marco propuesto.
- Capítulo 6. Conclusiones y futuras líneas de investigación: Se describen las conclusiones y hallazgos identificados durante la realización de esta tesis, así como posibles líneas futuras de investigación derivadas del trabajo presentado.

2 Estado del Arte

“Conócete, acéptate, supérate”

San Agustín

En este capítulo se recopila el estado del arte de los temas relacionados con la tesis doctoral, adicionalmente se hace una valoración con el propósito de establecer la diferencia entre el marco de interoperabilidad propuesto respecto a las alternativas identificadas en los trabajos relacionados.

Este capítulo comienza con la presentación y el estudio de la interoperabilidad, que es el tema en el que se centra esta investigación, se estudian los distintos tipos y definiciones, así como diversos frameworks de interoperabilidad que se utilizan en la comunidad científica.

Adicionalmente se definen diversos conceptos que inspiraron el marco propuesto, como ser Meta-Object Facility (MOF), Ontologías, Datashapes, entre otros. Se proponen los enfoques desde donde se aborda la reutilización de conocimientos de cara al marco propuesto en esta tesis.

Debido a que una de las premisas de esta tesis consiste en la representación unificada del conocimiento, se dedica una sección entera al estudio de lenguajes de representación de conocimientos que fueron revisados durante el desarrollo del marco propuesto. El análisis de estos lenguajes se llevó a cabo con el objetivo de proponer una alternativa que permitiera la representación unificada de cualquier elemento de conocimiento, independientemente de su formato, fuente o dominio.

Asimismo, se analizaron diferentes metodologías de evaluación de los niveles de interoperabilidad con el propósito de seleccionar un método de evaluación del marco de

interoperabilidad propuesto. Se valoraron diversos trabajos relacionados con la interoperabilidad tanto a nivel de interoperabilidad general, como distintas propuestas para la interoperabilidad en la ingeniería de sistemas. Se concluye el capítulo con unas conclusiones en donde se puntualizan las diferencias del marco propuesto respecto a los trabajos relacionados valorados a lo largo del presente capítulo.

2.1. Interoperabilidad

Según [47], una de las dificultades de la discusión acerca de la interoperabilidad es la ausencia de una definición clara de lo que significa interoperabilidad, de forma general en este trabajo se define la interoperabilidad como algo que denota la capacidad de un sistema, producto o servicio para comunicarse y funcionar con otros sistemas, productos o servicios (técnicamente diferentes).

Según [28] *“la interoperabilidad consiste en la capacidad de dos o más componentes de software para cooperar a pesar de las diferencias en el lenguaje, la interfaz y la plataforma de ejecución”*.

Según IEEE [48], *“la interoperabilidad es la capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información que se ha intercambiado”*.

Dentro de una cadena de suministros la interoperabilidad se define como *“la información previamente compartida en una variedad de formas, incluidas las conversaciones en papel y telefónicas, así como las diferentes comunicaciones electrónicas que se llevan a cabo entre los actores involucrados en un proceso de una cadena de suministros”* [49].

Existen otras definiciones de interoperabilidad adicionales [50–52] a las ya presentadas, pero la mayoría de estas definiciones coinciden en cuanto a aspectos como el intercambio de información y la capacidad de uso de esta información en sistemas complejos, esto puede ser un verdadero desafío por la cantidad de información que se puede necesitar gestionar.

En lo referente al intercambio de la información, este intercambio debe ser con sentido, es decir para poder tener capacidad de uso de la información es necesario entenderla [53]. Esto implica que esta comunicación de la que se habla como una de las premisas

2.1 Interoperabilidad

fundamentales de la interoperabilidad, debe tener un componente de comprensión por el receptor de la información, lo cual implica que más allá de una comunicación básica entre dos fuentes de información, debe haber un lenguaje que pueda ser producido por un emisor y entendido por el receptor.

2.1.1. Niveles de interoperabilidad

Según la teoría lingüística, el lenguaje debe contar con semántica, sintaxis y pragmática. Estos aspectos permiten el entendimiento entre la comunicación que se da entre sujetos, la sintaxis se refiere a los símbolos que representan la información, en el caso de la semántica se refiere al significado de la información, en cuanto a la pragmática se refiere al contexto en el que se da la comunicación. [54,55].

Considerando que la interoperabilidad implica un entendimiento entre el emisor y receptor, la comprensión de la información debe contar con aspectos de la teoría lingüística. En primer lugar, esta comunicación debe permitir la transmisión de información, es decir que los mensajes fluyan desde el emisor hacia el receptor de forma independiente de su ubicación física.

El segundo paso en esta comunicación es la estandarización del mensaje, esto se refiere a que el mismo, debe estar codificado de tal forma que pueda ser procesado tanto para su emisión como para su lectura, por ejemplo, información representada en diferentes formatos como *JavaScript Object Notation (JSON)*, *Extensible Markup Language (XML)*, texto, tabular, etc.

Adicionalmente debe existir un entendimiento más profundo de la información, esta suele ser la parte más compleja de lograr. Este entendimiento se refiere al significado de la información, con el propósito de ser integrado en procesos del receptor de la información.

En base a lo anterior, para efectos de esta tesis, se consideran los niveles de interoperabilidad propuestos por European Telecommunications Standards Institute (ETSI), los cuales consisten en niveles: técnico, sintáctico, semántico y organizacional [7,56–58].

La Figura 2-1 muestra los distintos niveles de interoperabilidad en forma de círculos concéntricos que reflejan los distintos niveles de interoperabilidad. Mientras más cerca

se esté del centro, más básica es la interoperabilidad. La complejidad va creciendo en función de los niveles que se alejan del centro.

Figura 2-1: Niveles de interoperabilidad [7].

En [7] los autores proponen la utilidad para determinar la no interoperabilidad entre dos entidades comunicantes. Es posible resumir la no interoperabilidad en las siguientes repuestas a cualquiera de las entidades comunicantes: ¿Dónde estás?, ¿Qué dijiste?, ¿Por qué hiciste eso?. La respuesta a estas preguntas no solamente nos permite en términos generales determinar si existe una interoperabilidad global que funciona de la forma en que se espera, sino también el nivel de interoperabilidad que no funciona de forma adecuada.

La pregunta ¿dónde estás? planteada entre las entidades comunicadoras implica la incapacidad de interactuar así sea de forma temporal entre dos entidades, esto se refiere a la interoperabilidad técnica, lo que significa que los protocolos o infraestructura que soporta la comunicación entre entidades podría ocasionar un funcionamiento inadecuado a nivel de toda la operación de interoperabilidad. Esto podría deberse a muchas causas, protocolos mal implementados, fallas en la arquitectura, necesidad de ajustes para lograr los niveles de disponibilidad necesarios, esto en función de la sensibilidad de los procesos involucrados entre las unidades comunicantes.

2.1 Interoperabilidad

En caso de ¿qué dijiste?, esto se refiere a sintaxis, es decir, si la comunicación es correcta y existen problemas de lectura de información entre entidades comunicantes probablemente se deba a problemas de sintaxis, esto debido a la apertura que tienen los lenguajes de intercambio de información como puede ser JSON y XML, RDF, etc. Probablemente los metamodelos sobre los cuales los formatos son generados cambien en el emisor y no en el receptor o viceversa, lo cual podría ocasionar incapacidad de procesar la sintaxis de la información [59].

La pregunta ¿por qué hiciste eso? es más compleja, esto revela la incapacidad de procesar significado del conocimiento transmitido entre entidades comunicantes, esto claramente implica la necesidad de procesar elementos inesperados, en este caso si los protocolos de procesamiento de la información semántico están bien diseñados existirá la posibilidad de procesar incluso aquellos aspectos inesperados.

Interoperabilidad técnica

Cuando se habla de interoperabilidad técnica, generalmente se refiere a elementos de hardware o software que permiten la intercomunicación entre máquinas, este tipo de interoperabilidad generalmente se refiere a la comunicación, en todos los elementos físicos necesarios para que esta comunicación se lleve a cabo, como ser protocolos e infraestructura [7,60,61].

La selección de protocolos y definición de infraestructura debe estar condicionada por las capacidades de las organizaciones, así como la sensibilidad del conocimiento proporcionado por las entidades comunicantes. Este nivel de interoperabilidad es el más básico y únicamente se refiere a la capacidad de comunicación entre entidades.

Interoperabilidad sintáctica

Se refiere a la gramática utilizada para codificar la información entre entidades comunicantes, esta gramática debe ser consensuada entre emisor y receptor y permite la lectura del contenido de la información incluyendo su estructura [7,62,63]. Existen muchas alternativas para proporcionar gramática a la información y que posibilitan la codificación

de la misma. Por lo general estos estándares son muy abiertos y dependiendo de la información aun cuando se seleccione un estándar es necesario contar con un metamodelo que permita garantizar la estructura de la información, por ejemplo, archivos XML con sus respectivos esquemas XSD.

Por lo anterior al momento de definir la gramática en este nivel de interoperabilidad se debe diseñar también aspectos como metamodelos necesarios sobre los protocolos y estándares a utilizar a nivel de interoperabilidad sintáctica.

Interoperabilidad semántica

Generalmente se asocia con el significado de la información. Este nivel de interoperabilidad suele ser más complicado ya que significa que existe un entendimiento común entre las personas sobre el significado de la información que se intercambia [7].

Lograr una interpretación semántica de la información puede implicar la interacción de múltiples elementos [57]. En algunos casos es necesario contar con ontologías, etiquetados en la información, taxonomías, meta modelos de información entre otros. Aun y cuando existen múltiples propuestas para abordar la interoperabilidad semántica [64–67], ésta permanece como un reto para las organizaciones hoy en día [68–70].

Interoperabilidad organizacional

De acuerdo con [7], interoperabilidad organizacional es la capacidad de las organizaciones para comunicarse y transferir de manera efectiva datos (información) (significativos) a pesar de que pueden estar utilizando una variedad de sistemas de información diferentes en infraestructuras muy diferentes, posiblemente a través de diferentes regiones geográficas y culturas. La interoperabilidad organizacional depende de una interoperabilidad técnica, sintáctica y semántica exitosa.

Lograr la interoperabilidad organizacional continúa siendo un desafío debido a la complejidad de los aspectos organizacionales [71–73].

2.1.2. Interoperabilidad en la cadena de suministros

La necesidad de cooperación entre los diferentes procesos e información de las organizaciones, ha llevado a la implantación de bases de datos centralizadas y cooperativas como los Sistemas de planificación de recursos empresariales (ERP), que permiten la gestión eficiente de los diferentes procesos involucrados internamente en una empresa [74]. En sistemas complejos donde se requiere un gran volumen de interacción entre elementos, se vuelve complejo implementar las interfaces entre los diferentes nodos y hacerlo de forma no estandarizada podría llevar a una mayor complejidad [75,76]. Una forma de lograr cadenas de herramientas bien integradas es a través de herramientas acopladas del mismo proveedor. Sin embargo, el problema con este enfoque es que en algunos casos no existe ningún proveedor que pueda proporcionar todas las herramientas necesarias [77].

2.2. Meta Object Facility (MOF)

Meta Object Facility (MOF) proporciona un marco de gestión de metadatos abierto e independiente de la plataforma y un conjunto asociado de servicios de metadatos para permitir el desarrollo, la interoperabilidad de modelos y sistemas basados en metadatos. Los ejemplos de sistemas que usan MOF incluyen herramientas de modelado y desarrollo, sistemas de almacenamiento de datos, repositorios de metadatos, etc. [78].

En [8], se presenta el marco tradicional para el metamodelado el cual se basa en una arquitectura con cuatro capas. Estas capas se describen convencionalmente de la siguiente manera:

1. La capa de objeto de usuario se compone de la información que se desea describir. Esta La información generalmente se conoce como “datos”.
2. La capa del modelo se compone de metadatos que describen información. Los metadatos se agregan informalmente como modelos.
3. La capa del metamodelo se compone de las descripciones (es decir, meta-meta-datos) que definen la estructura y la semántica de los metadatos. Los meta-meta-datos son informales agregado como metamodelos. Un metamodelo también pue-

Figura 2-2: Arquitectura de metadatos de MOF [8].

de considerarse como un “lenguaje” para describir diferentes tipos de datos.

4. La capa metametamodelo se compone de la descripción de la estructura y semántica de meta-meta-data. En otras palabras, es el “lenguaje” para definir diferentes tipos de metadatos.

La Figura 2-2 muestra la arquitectura que ofrece el metamodelado utilizando MOF, tiene una estructura piramidal que inicia con la capa M0 en donde no se tiene ningún entendimiento de la información, a medida que se va escalando en las diferentes capas se logra una mejor interpretación de la información.

La arquitectura de metadatos de MOF tiene algunas características importantes que la distinguen de las arquitecturas de metamodelado anteriores:

El modelo MOF está orientado a objetos y admite construcciones de metamodelado que están alineadas con (aunque un poco más simple que) las construcciones de modelado de objetos de UML [8]. El modelo MOF se describe a sí mismo. En otras palabras, el modelo MOF se define formalmente utilizando sus propias construcciones de metamodelado [8].

La naturaleza autodefinida del modelo MOF tiene algunas consecuencias importantes [8]:

1. Ayuda a validar las construcciones de metamodelado del MOF. Como el mode-

lo MOF puede describirse a sí mismo, debería ser adecuado para describir otros metamodelos de complejidad similar.

2. Permite que las interfaces y el comportamiento del MOF se definan mediante la aplicación de la asignación de MOF IDL (Lenguaje de Definición de Interfaz) al modelo MOF. Esto proporciona uniformidad de semántica entre objetos computacionales que representan modelos y metamodelos. También significa que cuando se define un nuevo mapeo tecnológico, las Interfaces de Programación de Aplicaciones (API) para administrar metamodelos en ese contexto también se definen implícitamente.
3. Proporciona una base arquitectónica para futuras extensiones y modificaciones del modelo MOF.

MOF ha sido utilizado en múltiples casos de estudio [79–83]. Por su naturaleza, es ideal para gestionar los diferentes artefactos de información de forma independiente de su dominio, ya que muchos de estos artefactos son modelos en sí mismos, uno de los más prominentes ejemplos son los modelos *Unified Modeling Language*(UML) y *Systems Modeling Language*(SysML).

2.3. Trazabilidad

Según [9], la trazabilidad en su forma más fundamental, consiste en relacionar datos que se encuentran almacenados dentro de artefactos de algún tipo, junto con la capacidad de examinar esta relación. La capacidad de lograr la trazabilidad depende, por tanto, de la creación de enlaces navegables entre datos almacenados dentro de artefactos que de otro modo estarían desconectados [9].

El valor de la trazabilidad radica en las numerosas actividades de ingeniería de sistemas, que la información proporcionada a través de tales interrelaciones puede permitir, como ser análisis de impacto, análisis de cobertura, análisis de dependencia, etc. [9,84–86]. La trazabilidad puede proveer visibilidad en los aspectos requeridos de desarrollo de sistemas [87–89].

Los componentes básicos de la trazabilidad descritos por [9] son:

- **Artefacto de traza:** Consisten en unidades de datos trazables que se refieren a datos o marcas residuales del proceso de desarrollo de software y sistemas que sean susceptibles de ser trazados. El término puede aplicarse a un requisito o una especificación completa de requisitos. El término general “artefacto” es utilizado definir tanto al objeto en su conjunto como a cualquier delimitación interna en el mismo. Lo que esto significa es que la granularidad de un artefacto de seguimiento no está predeterminada y puede que ni siquiera sea consistente en un proyecto en particular. Es esta incertidumbre en la granularidad de los artefactos traza que pueden conducir a muchos problemas en el establecimiento y utilizando la trazabilidad en la práctica [9].
- **Enlace de traza:** Es una asociación entre dos artefactos, uno que comprende el artefacto de origen y otro el artefacto de destino. Esta definición implica que el enlace cuenta con direccionalidad, la cual permite la posibilidad de dar seguimiento a la información partiendo desde la fuente [9].
- **Traza:** Puede definirse de dos formas: (1) traza(sustantivo), que consiste en un artefacto fuente, un artefacto de destino y un enlace de seguimiento que asocia los dos artefactos. (2) Traza (Verbo): El acto de seguir un enlace de traza desde un artefacto de origen a un artefacto de destino [9].
- **Trazabilidad:** Consiste en la posibilidad de crear, mantener y utilizar trazas. El desafío de llevar a cabo la trazabilidad se debe a la necesidad de contar con la posibilidad de adquirir, representar, almacenar y recuperar cada elemento de información [9].

En la Figura 2-3, se presenta un contexto de trazabilidad, en donde se detallan diversos artefactos que están relacionados entre sí. Como se puede ver, estos artefactos son de distinta naturaleza y han sido generados en distintas fases.

Figura 2-3: Ejemplo de contexto de trazabilidad [9].

2.4. Modelos

Según [90] “un modelo es un conjunto coherente de elementos formales que describen algo (por ejemplo, un sistema, banco, teléfono o tren) construido para algún propósito que es susceptible de una forma particular de análisis”, como:

- Comunicación de ideas entre gente y máquinas.
- Análisis de completitud.
- Análisis de condición de carrera.
- Generación de casos de prueba.
- Viabilidad en términos de indicadores como costo y estimación.
- Estándar.
- Transformación en implementación.

El uso de modelos en el diseño de sistemas de ingeniería complejos es una tradición de muchos años que es casi tan antigua como la ingeniería [91].

Existe evidencia de que era una práctica estándar en la Antigua Grecia. Vitruvio, arquitecto romano que vivió en el siglo I a.C., analiza el uso de modelos en el diseño de edificios y maquinaria de diversa índole. En el siglo XV, Galileo introdujo la noción de modelos matemáticos formales que aumentaron significativamente la confiabilidad del

diseño de ingeniería [91].

El propósito principal de los modelos de ingeniería es ayudarnos a comprender los aspectos interesantes de un sistema complejo antes de pasar por el gasto y el esfuerzo de construirlo [92]. Un buen modelo puede ayudar a arrojar luz sobre las características de un sistema donde hay incertidumbre sobre los requisitos o sobre la idoneidad de una solución propuesta, particularmente cuando la complejidad es tal que el razonamiento humano sin ayuda es insuficiente [91, 93]. Además de la mitigación de riesgos, Otro propósito fundamental de los modelos de ingeniería es comunicar ideas de diseño a las distintas partes interesadas en el sistema [94, 95].

Uno de los lenguajes de modelado más populares es UML [96–98], que es un lenguaje de modelado basado en el paradigma de objetos. Fue ideado como una consolidación de conceptos probados y se perfeccionó durante el espectacular aumento del interés en las tecnologías orientadas a objetos a finales de 1980.

La Figura 2-4 muestra las áreas semánticas de UML.

Figura 2-4: Áreas semánticas de UML [10].

UML ha sido ampliamente adoptado para distintas aplicaciones [99–104] y es compatible con la mayoría de los principales proveedores de herramientas de modelado. También se ha incorporado como parte esencial de los planes de estudio de ciencias de la compu-

tación e ingeniería en universidades de todo el mundo y en diversos programas de formación profesional. Por último, pero no menos importante, las comunidades académicas y de investigación y es en sí mismo objeto de importantes investigaciones [105].

UML también ha ayudado a crear conciencia general sobre el valor del modelado de software como un medio para hacer frente a la complejidad del software moderno. Aunque esta técnica de gran utilidad es casi tan antigua como el software en sí (los diagramas de flujo y las máquinas de estados finitos son ejemplos tempranos de modelado de software), la mayoría de los practicantes han tardado en aceptarla como algo más que una pequeña ayuda. Debido a que esta sigue siendo la actitud dominante, los métodos de desarrollo basados en modelos están encontrando una gran resistencia [105].

Dada la popularidad de UML en la ingeniería de software y teniendo en cuenta las necesidades de los ingenieros de sistemas, *Object Management Group (OMG)* decidió buscar UML para la ingeniería de sistemas. En marzo de 2003, el OMG emitió una solicitud de propuestas para una versión personalizada de UML adecuada para la ingeniería de sistemas, que respalda el modelado de una amplia gama de sistemas, que pueden incluir *hardware, software*, datos, personal, procedimientos e instalaciones. Solo hubo una presentación de tecnología a la solicitud, que fue del grupo SysML, que proponía un lenguaje basado en UML 2.0 [106].

2.5. Gestión de Conocimiento

“La gestión del conocimiento facilita la captura, implementación, acceso y reutilización de conocimiento” [107], las organizaciones se enfocan en la creación de sistemas de gestión del conocimiento que provean conocimiento explícito [108]. Según David Snowden, el conocimiento explícito “es reutilizable de manera coherente y repetible, en un sistema informático. El procedimiento documentado de un taller de lecciones aprendidas, los comentarios escritos de un economista que examina un conjunto de datos financieros. Son conocimientos explícitos que utilizamos para respaldar o tomar decisiones y ejercer nuestro juicio” [107],

En [109] se describe el proceso de reutilización de conocimientos con las siguientes etapas:

- Capturar o documentar el conocimiento: Puede suceder cuando por ejemplo equipos virtuales o comunidades generan archivos sobre su comunicación electrónica informal que puede ser buscada posteriormente.
- Empaquetar el conocimiento para su reutilización: Es el proceso de seleccionar, limpiar, estructurar, formatear o indexar documentos con un esquema de clasificación.
- Distribuir el conocimiento: Puede ser pasivo, como publicar un boletín informativo o llenar un repositorio para que los usuarios naveguen, o activo, como impulsar el conocimiento a través de alertas electrónicas para aquellos que lo necesitan.
- Reutilización del conocimiento: Se ha dicho que implica tanto el recuerdo (que la información se ha clasificado, en qué ubicación, bajo qué índice o esquema de clasificación) como el reconocimiento (que la información satisface las necesidades de los usuarios, así como la aplicación real del conocimiento) [110,111].

Dentro de la gestión de conocimientos, la reutilización es fundamental debido a que posibilita a las empresas a aprovechar el contenido de sus sistemas de gestión del conocimiento [112–116]. Es con la reutilización en nuevas situaciones de toma de decisiones que las empresas crean valor a partir del contenido de conocimiento existente [117–119]. Por lo tanto, las organizaciones se preocupan por los factores relacionados con facilitar la reutilización en sus sistemas [120–122]. Considerando la importancia que tiene la gestión del conocimiento como herramienta para hacer los procesos de las organizaciones más eficientes, existen modelos que ayudan a integrar estrategias de gestión del conocimiento [123,124], esto derivado de las necesidades de gestión que actualmente están vigentes en las organizaciones [125–128].

2.6. Representación del conocimiento

Según [129] es posible entender mejor la representación del conocimiento en términos de papeles que desempeña la misma, algunos de los papeles son:

En primer lugar una representación del conocimiento se refiere a la definición de un sustituto, así como la definición de acuerdos de uso común, es decir, acuerdos que todos deben entender y procesar de la misma forma. Por último, es un medio de expresión humana, es decir, un lenguaje en el que decimos cosas sobre el mundo.

Como lo menciona [130], la gestión de conocimientos se basa en una estrategia adecuada de representación de conocimientos. En base a lo anterior y considerando la necesidad de reutilizar el conocimiento entre distintos actores que intervienen en el ciclo de desarrollo de la Ingeniería de Sistemas, la representación de conocimiento como base para interoperar y reutilizar conocimiento se volvió una necesidad importante. .

Actualmente existen múltiples alternativas para representar conocimiento [131–133]. [130] desarrolla un debate entre ontologías formales versus Datashapes, en donde se menciona la evolución que han tenido las ontologías formales desde la perspectiva de la Web Semántica, así como también la fuerza que han cobrado los Datashapes como ser *Open Services for Lifecycle Collaboration (OSLC)* en sus distintos Resource Shapes, incluso se menciona la posibilidad de tener enfoques híbridos.

2.6.1. Ontologías formales

Según [134] “una ontología es una especificación explícita de una conceptualización” [134]. El término está tomado de la filosofía, donde una ontología es una explicación sistemática de la existencia. Adicionalmente, es necesario definir convenciones en tres niveles: Formato del lenguaje de representación, protocolo de comunicación del agente y especificación del contenido del conocimiento compartido [134] afirma que las propuestas de formatos estándar de representación del conocimiento y los lenguajes de comunicación de los agentes son independientes del contenido del conocimiento que se intercambia o comunica. Para establecer acuerdos sobre el conocimiento, tales como supuestos compartidos y modelos del mundo, las ontologías pueden jugar un papel de especificación de software. Actualmente, existen varias alternativas de representación de conocimiento basado en otologías [135–139].

2.6.2. Datashapes

En [130] se define un Datashape como un recurso, es decir, un metamodelo, que describe el contenido y las limitaciones de otros recursos. Adicionalmente [140] hace una diferencia entre los estándares *Web Ontology Language (OWL)* y *Resource Description Framework (RDF)* provistos por *World Wide Web Consortium (W3C)* para describir vocabulario y ontologías formales, pero en este trabajo los autores explican que estos estándares están definidos para la inferencia de nuevo conocimiento y no para otras necesidades como ser especificar restricciones en los datos RDF, si bien RDF permite validar gráficos contra un conjunto de condiciones [141]. Los autores proponen OSLC (ver en sección 2.8.2) para satisfacer esta necesidad, lo cual junto con la discusión desarrollada por [130] deja entrever la relevancia que tienen en la gestión del conocimiento los Datashapes al momento de realizar interoperabilidad.

Un modelo de representación de información basado en relaciones: RSHP

Uno de los principales objetivos para mejorar la eficiencia de la ingeniería de software ha sido encontrar formas de administrar conjuntamente diferentes tipos de información y activos de software para permitir la reutilización de la información. Se hace necesario encontrar un modelo de representación de información que sea capaz de hacer frente a los diferentes tipos de artefactos presentes en los diferentes procesos de ingeniería, de forma independiente de su formato o estructura [11]. Ésta es la finalidad del modelo de representación de información RSHP.

RSHP, consiste en los siguientes siete elementos principales, los cuales se utilizan para representar la información, considerando la sintaxis, semántica y pragmática [11]. A continuación se detallan los componentes de RSHP:

1. **Artifact:** Se define como un contenedor de conocimientos, puede ser utilizado como nodo dentro de una relación, o simplemente ser una especificación de un elemento de conocimiento (KE) [11].
2. **Term:** Permite representar un concepto semántico, en lenguaje natural, a este elemento se le pueden agregar tipologías para especificar la sintaxis del lenguaje, por

2.6 Representación del conocimiento

Figura 2-5: Modelo de representación de información de RSHP [11].

ejemplo, nombres, verbos, etc [11].

3. Relationship: Es el componente principal en el modelo RSHP y permite la definición de grafos para todo tipo de artefactos de información, estas relaciones pueden conectar dos KEs [11].
4. Knowledge Element: Es la mas pequeña unidad de elementos de conocimiento y está relacionada con la ocurrencia de un término o artefacto [11].
5. Metaproperty: Representa meta información de tipo clave/valor acerca de un artefacto [11].
6. Term Tag: Es la clasificación sintáctica de un término [11].
7. Cluster: Es una agrupación semántica que permite establecer pragmática a la terminología [11].

RSHP permite representar cualquier tipo de conocimientos en forma de grafos semánticos, de forma independiente de su formato, o naturaleza. RSHP cuenta con tecnología que ofrece servicios de indexación trazabilidad, gestión de conocimientos, entre otros [142]. Estas posibilidades han permitido la realización de investigaciones sobre la reutilización de artefactos de ingeniería como ser modelos lógicos [143, 144] y físicos [145].

2.6.3. Modelo de representación de conocimientos aplicado en la tesis

Para efectos de esta tesis se ha desarrollado un *Resource Shape* denominado *System Representation Language (SRL)* [130], implementado en base a trabajos previos [1, 20]. En la sección 3.3.3 se presenta el detalle del SRL.

Aun cuando existen otros estándares para representar conocimientos, se seleccionó SRL como lenguaje único de representación de conocimientos, debido a la simplicidad al momento de representar los diferentes artefactos de ingeniería de forma independiente de su formato. Adicionalmente APIs en versiones C# y Java, están disponibles componentes para serializar los objetos de SRL a: (1) RDF, (2) RSHP y (3) OWL. El propósito de estos componentes es transformar los artefactos representados en los tres estándares antes detallados, esto permite la integración con tecnologías que funcionan bajo los estándares antes mencionados.

En el caso de RSHP la transformación del conocimiento representado en SRL permitió la integración con las tecnologías que subyacen sobre las APIs de RSHP [142], las cuales tecnologías para reutilización, trazabilidad, transformación, análisis de calidad, entre otros.

Posicionamiento de SRL respecto a MOF

En la Tabla 2-1 se presenta un ejemplo que pretende posicionar y comparar diferentes lenguajes de representación y modelado de información respecto a MOF.

En la primera columna se presentan los distintos niveles de MOF: M0, M1, M2 y M3. En la segunda columna se describe el modelo UML de acuerdo con cada una de las capas

2.7 Métodos de evaluación sobre nivel de madurez de interoperabilidad en las organizaciones

Tabla 2-1: Ejemplo MOF para SRL, UML y OSLC.

	UML	SRL	SRL- MOF UML	OSLC	OSLC-SRL
M3	Class	Class	Class	Property, resource, etc.	Property, resource, etc.
M2	Class, Attribute, Interface	Artifact, Term, Property, Type	Class, Attribute, Interface	Core	Core
M1	Patient, Doctor	SRL- Requirements, SRL - Logical Models, Etc	Artifact, Term, Property, Type	Requirement, Test Case, etc.	SRL- Requirements, SRL - Logical Models, Etc
M0	Patient Juan, Doctor Ernesto	[The]→Artifact [System]→Artifact	[The]→Artifact [System]→Artifact	Instancia de OSLC Requirement, Instancia OSLC Test Cases	[The]→Artifact [System]→Artifact
		Requirement: "The system shall fly"	Requirement: "The system shall fly"	Requirement: "The system shall fly"	Requirement: "The system shall fly"

de MOF. En la tercera columna se detallan el SRL, en la cuarta columna se fusionan el SRL bajo la representación MOF de UML, las dos últimas columnas presentan el OSLC y una fusión de OSLC y SRL.

2.7. Métodos de evaluación sobre nivel de madurez de interoperabilidad en las organizaciones

La interoperabilidad se está convirtiendo en un factor clave para el éxito en organizaciones o empresas por lo tanto, la interoperabilidad entre sistemas, requiere, una atención considerable para ser evaluada y mejorada continuamente [146]. Existen modelos de madurez, estándares, metodologías y pautas que pueden ayudar a una organización, una empresa, o más generalmente un sistema, mejorando la forma en que opera con otros y por lo tanto lograr los objetivos de interoperabilidad deseados. En los siguientes apartados se detallan algunos de los modelos más importantes para medir los niveles de madurez en las organizaciones en lo que se refiere a la interoperabilidad.

2.7.1. El modelo de Niveles de Interoperabilidad de Sistemas de Información (LISI)

Fue desarrollado por el Grupo de Trabajo C4ISR del Departamento de Defensa de EE. UU., y considera cinco niveles crecientes de sofisticación con respecto a la interacción del sistema y la capacidad del sistema para intercambiar y compartir información y servicios, cada nivel superior representa un aumento demostrable en las capacidades sobre el nivel anterior de la interacción sistema a sistema [12].

LISI define un conjunto de “niveles” de interoperabilidad cada vez más sofisticados o maduros. Cada nivel representa una caracterización específica de varios elementos y el conjunto asociado de capacidades presentes para fomentar la interoperabilidad [12]. La Figura 2-6, muestra el modelo de madurez LISI.

A continuación se detallan los distintos niveles de madurez que propone LISI:

- **Nivel 0 Interoperabilidad aislada en un entorno manual:** Abarca la amplia gama de sistemas aislados o autónomos. No se permite ni se encuentra disponible ninguna conexión electrónica directa, por lo que la única interfaz entre estos sistemas es mediante el cambio de clave manual o mediante medios comunes extraíbles. La fusión de información, si la hubiera, se realiza fuera de línea el responsable de la toma de decisiones por otros medios automatizados [12].
- **Nivel 1 Interoperabilidad conectada en un entorno peer-to-peer:** Los sistemas pueden conectarse electrónicamente y proporcionar alguna forma de intercambios simples. Estos sistemas tienen una capacidad limitada, por lo general pasan tipos de datos homogéneos, como voz, correo electrónico de “texto” simple o archivos gráficos fijos como imágenes GIF o TIFF entre estaciones de trabajo. Permiten a los tomadores de decisiones intercambiar información unidimensional, pero tienen poca capacidad para fusionar información para respaldar la toma de decisiones [12].
- **Nivel 2 Interoperabilidad funcional en un entorno distribuido:** Los sistemas residen en redes locales que permiten que los conjuntos de datos pasen de un sistema a otro. Proporcionan intercambios de medios cada vez más complejos. Están pre-

2.7 Métodos de evaluación sobre nivel de madurez de interoperabilidad en las organizaciones

LEVEL (Environment)			Interoperability Attributes				
			P	A	I	D	
Enterprise Level (Universal)	4	c	Multi-National Enterprises	Interactive (cross applications)	Multi-Dimensional Topologies	Cross-Enterprise Models	
		b	Cross Government Enterprise			Enterprise Model	
		a	DoD Enterprise	Full Object Cut & Paste			
Domain Level (Integrated)	3	c	Domain Service/Agency Doctrine, Procedures, Training, etc.	Shared Data (e.g., Situation Displays Direct DB Exchanges)	WAN	DBMS	
		b		Group Collaboration (e.g., White Boards, VTC)		Domain Models	
		a		Full Text Cut & Paste			
Functional Level (Distributed)	2	c	Common Operating Environment (e.g., DII-COE Level 5) Compliance	Web Browser	LAN	Program Models & Advanced Data Formats	
		b		Basic Operations Documents Briefings Pictures & Maps Spreadsheets Databases			
		a	Program Standard Procedures, Training, etc.	Adv. Messaging Message Parsers E-mail w/Attachments	NET		
Connected Level (Peer-to-Peer)	1	d	Standards Complaint (e.g., JTA)	Basic Messaging (e.g., Unformatted Text E-mail w/o Attachments)	Two Way	Basic Data Formats	
		c		Data File Transfer			
		b	Security Profile	Simple Interaction (e.g., Telemetry, Remote Access Text Chatter, Voice, Fax)	One Way		
		a					
Isolated Level (Manual)	0	d	Media Exchange Procedures	N/A	Removable Media	Media Formats	
		c	Manual Access Controls		Manual Re-entry	Private Data	
		b					
		a					
		o					NO KNOWN INTEROPERABILITY

Figura 2-6: Modelo de madurez de LISI [12].

sentes modelos de datos formales (lógicos y físicos). Sin embargo, en general, solo se acepta el modelo de datos lógicos en todos los programas y cada programa define su propio modelo de datos físicos. Los datos son generalmente heterogéneos y pueden contener información de muchos formatos simples fusionados, como una imagen con una superposición de anotaciones. Los tomadores de decisiones pueden compartir información fusionada entre sistemas o funciones [12].

- **Nivel 3 Interoperabilidad basada en dominios en un entorno integrado:** Los sistemas pueden conectarse a través de redes de área amplia (WAN) que permiten

que varios usuarios accedan a los datos. La información de este nivel se comparte entre aplicaciones independientes. Existe un modelo de datos basado en el dominio (lógico y físico) que se comprende, acepta e implementa en un área funcional o grupo de organizaciones que comprende un dominio. Al utilizar modelos de datos de dominio acordados, los sistemas ahora deben ser capaces de implementar reglas y procesos comerciales para facilitar las interacciones directas de base de datos a base de datos, como las necesarias para admitir servidores de replicación de bases de datos. Las aplicaciones individuales en este nivel pueden compartir repositorios de datos centrales o distribuidos. Los sistemas de este nivel apoyan la colaboración en grupo en productos de información fusionados. La toma de decisiones está respaldada por información fusionada de un dominio localizado [12].

- **Nivel 4 Interoperabilidad basada en empresas en un entorno universal:** Los sistemas son capaces de operar utilizando un espacio de información global distribuido en múltiples dominios. Varios usuarios pueden acceder e interactuar con datos complejos simultáneamente. Los datos y las aplicaciones se comparten por completo y se pueden distribuir en este espacio para apoyar la fusión de información. Son posibles formas avanzadas de colaboración (el concepto de oficina virtual). Los datos tienen una interpretación común independientemente de su forma y se aplican a toda la empresa. Se reduce la necesidad de aplicaciones redundantes y funcionalmente equivalentes, ya que las aplicaciones se pueden compartir tan fácilmente como los datos en este nivel. La toma de decisiones tiene lugar en el contexto de, y es facilitada por, la información de toda la empresa que se encuentra en este espacio de información global [12].

Como se muestra en la Figura 2-7 LISI categoriza los diversos aspectos de la interoperabilidad de los sistemas de información en términos de cuatro atributos completos, estrechamente interrelacionados: Procedimientos, Aplicaciones, Infraestructura y Datos (PAID). Son los facilitadores de la interoperabilidad y los distintos niveles de madurez. Individualmente, estos atributos son como piezas de un rompecabezas y pueden representarse como un círculo completo cuya “Circunferencia” abarca todo el ámbito de los

2.7 Métodos de evaluación sobre nivel de madurez de interoperabilidad en las organizaciones

problemas de interoperabilidad [13].

Figura 2-7: Atributos PAID [13].

2.7.2. Modelo de madurez de interoperabilidad organizacional

Fue desarrollado en 1998, como una extensión del modelo LISI a capas más abstractas de soporte de comando y soporte de control. La Tabla 2-2 muestra el modelo de madurez de interoperabilidad organizacional en detalle [3].

- **Nivel 0 - Independiente:** La interacción entre organizaciones independientes se describe en este nivel de interoperabilidad. Incluye las organizaciones que normalmente podrían trabajar sin ninguna interacción, excepto la proporcionada por el contacto personal. Este nivel también contiene las organizaciones que no comparten propósitos u objetivos comunes, pero que pueden necesitar interoperar en escenarios sin precedentes [3].
- **Nivel 1 - Ad hoc:** Solo algunos de los marcos organizativos están en una posición que podría apoyar acuerdos ad hoc en este nivel de interoperabilidad. Se proporciona un conjunto de pautas para explicar cómo se llevará a cabo la interoperabilidad, pero básicamente no se han planificado arreglos específicos. En este nivel de interoperabilidad organizacional, existen algunos objetivos generales compartidos, sin embargo, las aspiraciones de las organizaciones individuales tienen prioridad

Tabla 2-2: "Organizational interoperability maturity model for c2" [3].

	Preparedness	Understanding	Command Style	Ethos
Unified	Complete - normal day-today working	Shared	Homogeneous	Uniform
Combined	Detailed doctrine and experience in using it	Shared comms and shared knowledge	One chain of command and interaction with home org	Shared ethos but with influence from home org
Collaborative	General doctrine in place and some experience	Shared comms and shared knowledge about specific topics	Separate reporting lines of responsibility overlaid with a single command chain	Shared purpose; goals, value system significantly influenced by home org
Ad hoc	General guidelines	Electronic comms and shared information	Separate reporting lines of responsibility	Shared purpose
Independent	No preparedness	Communication via phone etc	No interaction	Limited shared purpose

y las organizaciones siguen siendo totalmente distintas [3].

- **Nivel 2 - Colaborativo:** El nivel de interoperabilidad organizacional colaborativo es donde existen marcos reconocidos para respaldar la interoperabilidad y se reconocen los objetivos compartidos y los roles y responsabilidades se asignan como parte de las responsabilidades en curso, sin embargo, las organizaciones siguen siendo distintas [3].
- **Nivel 3 - Integrado:** Es aquel en el que hay sistemas de valores y objetivos compartidos, un entendimiento común y una preparación para interoperar [3].
- **Nivel 4 - Unificado:** es aquella en la que los objetivos de la organización, los sistemas de valores, la estructura / estilo de mando y las bases de conocimiento se comparten en todo el sistema. La organización está interoperando de forma continua [3].

2.7.3. The Layered Interoperability Score (i-Score)

En [147], los autores presentan un método para la medición de la interoperabilidad de redes complejas de sistemas no homogéneos, el cual proporciona una medida numérica única de qué tan bien interoperan los sistemas. La metodología primero define una medición de referencia de la interoperabilidad para una red no homogénea de sistemas, ya que se utilizan dentro de un escenario operativo o “hilo”. La metodología luego define la puntuación teórica óptima de interoperabilidad y propone algunas heurísticas que se pueden utilizar para mejorar la interoperabilidad general de la red.

El i-Score consta de los siguientes pasos:

- Diagramar el hilo operacional y definir el conjunto de sistemas de soporte: Este paso consiste en definir una serie de actividades que se requiere sean soportadas por los sistemas a evaluar, estas actividades pueden ser definidas mediante un diagrama de actividades SysML o UML.
- Crear una matriz de interoperabilidad: Se define una matriz en donde el vértice de esta, consiste en el conjunto de conexiones dirigidas entre sistemas a evaluar y el valor que corresponde al hilo operativo soportado.
- Calcular el i-Score: Con la información de la matriz se calcula el i-Score, que consiste en la puntuación de interoperabilidad que se pretende maximizar.
- Determinar el i-Score óptimo: El i-Score óptimo consiste en la puntuación máxima posible a la luz de las limitaciones físicas y operativas.
- Determinar la brecha de interoperabilidad: Consiste en determinar cuánta mejora de interoperabilidad se puede realizar considerando las limitaciones físicas y operativas en el hilo operativo.
- Realizar análisis de interoperabilidad: En base a los cálculos anteriores, se identifican posibles áreas de mejora de la interoperabilidad.

El i-Score permite identificar puntos de mejora en los niveles de interoperabilidad y es sumamente útil para la toma de decisiones, considera limitaciones técnicas y operacionales, por lo que es un indicador realista de las posibilidades de mejora.

2.7.4. Service Measurement Index

En [14] se presenta un framework estándar para medición que incluye siete (7) características principales que engloban un grupo de atributos por cada una de estas características, actualmente el framework cuenta con un total de 51 atributos los cuales se muestran en la Figura 2-8.

Figura 2-8: Atributos SMI [14].

A continuación se detallan las principales categorías que conforman el framework SMI:

- Responsabilidad: Atributos que se utilizan para la medición de propiedades relacionadas con la organización del proveedor de servicios.
- Agilidad: Atributos que indican el impacto del servicio sobre las habilidades del cliente para cambiar de estrategia dirección o tácticas con la mínima interrupción.
- Garantía: Atributos que indican el impacto de un servicio sobre la capacidad del cliente para cambiar de estrategia con una interrupción mínima.

2.7 Métodos de evaluación sobre nivel de madurez de interoperabilidad en las organizaciones

- Finanzas: El costo monetario en que incurre el cliente por el servicio.
- Desempeño: Características y funciones que proveen los servicios, en esta categoría se incluye la interoperabilidad.
- Seguridad: Atributos que indican la efectividad del servicio en cuanto al control de acceso y la privacidad de la información.
- Usabilidad: La facilidad con la que el servicio puede ser utilizado por los clientes.

2.7.5. Método de evaluación común para estándares y especificaciones (CAMSS)

CAMSS es una guía europea que permite la realización de la evaluación y selección de estándares para proyectos de administración electrónica. Puede utilizarse como referencia al construir una arquitectura TIC y para justificar la selección de estándares y especificaciones. CAMSS proporciona un conjunto de recursos que facilitan la evaluación, asimismo es un referente a la hora de construir una arquitectura y permite justificar la elección de estándares y especificaciones en términos de necesidades de interoperabilidad y requisitos. CAMSS está totalmente alineado con el Reglamento europeo de normalización 1025/2012 [15].

La Figura 2-9 muestra los elementos de CAMSS disponibles en cada uno de los componentes publicados.

Figura 2-9: Relación entre las soluciones CAMSS y sus componentes publicados [15].

CAMSS puede considerarse un método de evaluación transparente y colaborativo, ya que está disponible públicamente. Esta herramienta está respaldada por especificaciones

funcionales que se han apoyado en esta guía [148]. Entre otras cosas la evaluación se enfoca en la apertura de estándares y especificaciones, así como también contiene criterios para garantizar la solidez y consistencia.

2.7.6. Evaluación de la madurez de la interoperabilidad semántica de un servicio público (SIMAPS)

SIMAPS es una herramienta que permite medir la madurez de interoperabilidad de servicios públicos digitales desde el punto de vista de interoperabilidad semántica [149].

Posibilita la evaluación, mejora y consideración de aspectos clave de interoperabilidad, observando el contexto de: Prestación de servicios (D) y consumo de servicios (C). SIMAPS evalúa: Especificaciones de interoperabilidad semántica de los datos, el conocimiento enviado y recibido; capacidades de interoperabilidad semántica que permiten el envío y la recepción de conocimiento, adicionalmente es totalmente compatible con la última versión del Marco Europeo de Interoperabilidad (EIF) [149].

2.8. Trabajos relacionados

En esta sección se realiza una revisión de los trabajos relacionados con la presente investigación, se valoran los aportes realizados por otros autores y se establece una diferencia entre el marco propuesto respecto a éstas. Lo anterior con el propósito de presentar de forma explícita la originalidad de los aportes que se obtienen del marco de interoperabilidad propuesto, respecto a los trabajos relacionados.

2.8.1. Trabajos relacionados con interoperabilidad general

En este apartado se presentan diversas propuestas para lograr interoperabilidad de forma genérica, a nivel empresarial, entre otros, los trabajos valorados en este apartado constituyen marcos de interoperabilidad aplicados a ramas distintas a la ingeniería de sistemas.

Nuevo marco europeo de interoperabilidad (EIF)

El EIF es un marco de trabajo genérico propuesto por la Comisión Europea, tiene como propósito brindar orientación, a través de un conjunto de recomendaciones, a las administraciones públicas sobre cómo mejorar la gestión de sus actividades de interoperabilidad, establecer relaciones entre organizaciones, agilizar los procesos que respaldan los servicios digitales de un extremo a otro y garantizar que la legislación existente y nueva no comprometa los esfuerzos de interoperabilidad [16, 150].

La estructura de EIF consta de un (1) modelo conceptual, cuatro (4) niveles de interoperabilidad, doce (12) principios subyacentes y cuarenta y siete (47) recomendaciones. A efectos del EFI, la interoperabilidad es la capacidad de las organizaciones para interactuar hacia objetivos mutuamente beneficiosos, que implican el intercambio de información y conocimiento entre estas organizaciones, a través de los procesos comerciales que apoyan, mediante el intercambio de datos entre sus sistemas de TIC.

Los principios de interoperabilidad propuestos por EIF, son aspectos fundamentales de comportamiento para encaminar acciones de interoperabilidad, los mismos están divididos en 4 categorías:

1. Estableciendo el contexto para acciones sobre la interoperabilidad a nivel EU.
2. Principios fundamentales de interoperabilidad.
3. Principios relacionados con las necesidades genéricas de los usuarios y sus expectativas.
4. Principios fundamentales de la cooperación entre administraciones públicas

EIF propone cuatro niveles de interoperabilidad los cuales son:

- **Interoperabilidad Legal:** Consiste en la gestión de los marcos legales que se ven involucrados en los procesos de interoperabilidad. Lograr la interoperabilidad legal podría implicar ajustes a las legislaciones vigentes entre los actores involucrados en los procesos de interoperabilidad, incluso el desarrollo de nuevas legislaciones.
- **Interoperabilidad Organizacional:** Se refiere a lograr la integración del conocimiento obtenido producto de los procesos de interoperabilidad dentro de los procesos de negocio de los involucrados.

- **Interoperabilidad Semántica:** Esta se encarga de que el formato del conocimiento involucrado en los procesos de interoperabilidad sea preciso, consensuado, bien diseñado y con las capacidades necesarias para satisfacer las necesidades de transmisión del conocimiento, adicionalmente asegura que el significado de los datos e información intercambiados se conservan y comprenden a través de los intercambios entre las partes. En el EIF, la interoperabilidad semántica cubre aspectos tanto semánticos como sintácticos.
- **Interoperabilidad Técnica:** Esto implica aplicaciones, infraestructura, protocolos de comunicación, servicios de interconexión, servicios de integración de datos entre otros.

En la Figura 2-10 se muestra el modelo conceptual de EIF, el cual se fundamenta en realizar la “Interoperabilidad por diseño”, es decir diseñar los componentes siguiendo el modelo propuesto por EIF. El modelo promueve la reutilización como motor de la interoperabilidad, reconociendo que los servicios públicos europeos deben reutilizar la información y los servicios que ya existen y pueden estar disponibles en diversas fuentes dentro o fuera de los límites organizativos de las administraciones públicas.

Figura 2-10: Modelo conceptual de EIF [16].

EIF es un marco de trabajo muy completo orientado a promover la comunicación entre administraciones públicas Europeas, todo esto mediante un grupo de principios y recomendaciones, tiene en cuenta los aspectos legales y tecnológicos para el desarrollo de una interoperabilidad que sea beneficiosa para todos los involucrados.

2.8 Trabajos relacionados

La propuesta de [16, 150] es compatible con el trabajo de esta tesis, ya que EIF está concebido para reutilizar los mecanismos de interoperabilidad y servicios existentes dentro de los estados que intervienen en los procesos de interoperabilidad, La metodología de esta tesis podría fortalecer los procesos de interoperabilidad que se realizan a nivel sintáctico y semántico, lo cual vendría a apoyar la implementación del EFI.

Sistematización del conocimiento sobre interoperabilidad: la base de la interoperabilidad empresarial como ciencia

En [17], los autores realizan un análisis a diferentes trabajos académicos relacionados con la interoperabilidad empresarial. Los autores remarcan la necesidad de lograr una recopilación de conceptos y aspectos a considerar para la futura compilación de un cuerpo de conocimiento denominado “Interoperability Body of Knowledge (IBoK)”, que según los autores aun se encuentra en estado embrionario, estos esfuerzos han buscado organizar y agregar información de fuentes muy fragmentadas y dispares, y con diferentes granularidades de detalle, distintos orígenes epistemológicos, campos académicos separados, etc.

Los autores se enfocan en estandarizar y darle un tono formal al estudio de la Interoperabilidad Empresarial, algo que se denomina Enterprise Interoperability Science Based (EISB) [17], esto debido a la diversidad de estudios en diferentes dominios sobre la interoperabilidad, estandarizar la diversidad de conceptos, estructuras y modelos puede ser una tarea sumamente complicada.

En la Figura 2-11 se presenta el enfoque propuesto por los autores para llevar a cabo la definición del EISB partiendo de la revisión de los trabajos relacionados con la Interoperabilidad Empresarial. Este enfoque consiste en la ejecución paralela de dos procesos que se comunican entre sí con el propósito de asegurar la ejecución y alineación correcta de las especificaciones, contenidos, procesos, etc. que deben ser incluidos en el EISB.

El trabajo presentado por [17] presenta un intento por alinear los distintos aportes que existen a nivel académico relacionados con la interoperabilidad empresarial, se intenta estandarizar los niveles de interoperabilidad, así como también otros conceptos base, pero

Figura 2-11: Enfoque metodológico para la definición de EISB y revisión de la situación [17].

a diferencia del trabajo propuesto por esta tesis no se pretende proporcionar una metodología para construir procesos de interoperabilidad entre fuentes. Por lo que el trabajo propuesto en esta tesis podría enriquecer una posible definición de EISB, considerando que esta metodología ofrece mecanismos conceptuales y tecnológicos para lograr interoperabilidad a nivel semántico.

Marco de Interoperabilidad Empresarial

En [18], se propone un enfoque basado en barreras que permite identificar y estructurar el conocimiento. El marco propuesto se define a partir de conceptos desarrollados en algunos marcos y modelos existentes [16,151], centrándose en aquellos conceptos más relevantes para definir la interoperabilidad empresarial.

La Figura 2-12 muestra el marco de interoperabilidad propuesto por [18]. Según [18], la categorización de las barreras se puede definir de la siguiente forma: 1)Conceptuales: Se preocupan por las diferencias sintácticas y semánticas (Interoperabilidad sintáctica y semántica) de la información a intercambiar. 2)Tecnológicas: Se refieren a la incompatibilidad de las tecnologías(Interoperabilidad técnica) de la información y las organizacionales las cuales se relacionan con la definición de responsabilidad (¿quién es responsable de

2.8 Trabajos relacionados

qué?) y autoridad (¿quién está autorizado para hacer qué?) así como la incompatibilidad de las estructuras organizacionales.

La segunda dimensión de interoperabilidad, representa los niveles de interoperabilidad:

- **Datos:** Consiste en compartir información de bases heterogéneas, y que además pueden residir en diferentes máquinas con diferentes sistemas operativos y sistemas de gestión de bases de datos.
- **Servicios:** Es responsable de identificar, componer y hacer funcionar conjuntamente diversas aplicaciones (diseñadas e implementadas de forma independiente).
- **Procesos:** Tiene como objetivo hacer que varios procesos funcionen juntos: Un proceso define la secuencia de los servicios (funciones) de acuerdo con unas necesidades específicas de una empresa. En una empresa en red, también es necesario estudiar cómo conectar los procesos internos de dos empresas para crear un proceso común.
- **Negocios:** Se refiere a trabajar de manera armonizada a nivel de organización y empresa a pesar de, por ejemplo, los diferentes modos de toma de decisiones, métodos de trabajo, legislaciones, cultura de la empresa y enfoques comerciales, etc. para que se puedan desarrollar negocios entre empresas

La interoperabilidad de los datos: Se refiere a hacer trabajar juntos diferentes modelos de datos y lenguajes de consulta, los cuales se basan en [151].

La tercera dimensión (enfoques de interoperabilidad) se agrega al marco bidimensional. Esta tercera dimensión permite categorizar el conocimiento y las soluciones relacionadas con la interoperabilidad empresarial de acuerdo con las formas de eliminar diversas barreras de interoperabilidad.

Por ejemplo, PSL (Lenguaje de especificación de proceso) contribuye a eliminar la barrera conceptual (tanto sintáctica como semántica) a nivel de proceso a través de un enfoque unificado. Con respecto a la técnica de anotación semántica, si la anotación y el mapeo se realizan utilizando una ontología de referencia predefinida, entonces es un enfoque unificado. Por otro lado, si la anotación y el mapeo se realizan mediante nego-

ciación sobre la marcha, entonces es un enfoque federado.

Figura 2-12: Estructura del Marco de Interoperabilidad Empresarial [18].

El marco de trabajo propuesto por [18] proporciona una aproximación global para gestionar la interoperabilidad empresarial y enfoque basado en la identificación de barreras para definir el dominio de investigación de la interoperabilidad empresarial, pero recalca que existen dimensiones adicionales como ser: 1) fases de ingeniería. 2) tipos de conocimientos, así como la necesidad de definir métricas de interoperabilidad. En vista de lo anterior se recalca la falta de gestión de los tipos de conocimientos a interoperar, por esta razón el marco propuesto en esta tesis es compatible con la propuesta de [18], como una alternativa para gestionar los tipos de conocimiento, que es una de las dimensiones adicionales identificadas por [18].

Towards a classification framework for interoperability of enterprise applications

En [152], se propone un marco de trabajo en el cual se pretenden clasificar distintas técnicas de interoperabilidad. Adicionalmente, se estudian los diferentes modelos de ma-

durez.

Se detallan dos ejemplos de clasificación:

1. El primero basado en UEMML (unified enterprise modelling language) [153, 154], que es un formato de intercambio, pero podría considerarse como un lenguaje de modelado empresarial, que puede especificar los procesos empresariales en su entorno. Este formato se propone como alternativa de clasificación a un nivel operacional. La interoperabilidad entre aplicaciones que utilizan UEMML como lenguaje fundamental necesita, procesos de transformación de modelos y alineación semántica.
2. El segundo basado en IEC6224 [155] en donde se propone como alternativa para clasificar la interoperabilidad en un nivel más operativo.

El resultado que se obtiene de la clasificación utilizando ambos enfoques (UEMML Y IEC6224), refleja la necesidad de alinear las transformaciones a un nivel más semántico. Adicionalmente, en este trabajo se intenta de manera objetiva evaluar la dificultad de integrar y desarrollar algunos trabajos de interoperabilidad dentro de entornos empresariales.

En [152] únicamente se llevan a cabo dos clasificaciones y en las conclusiones del paper se especifica la necesidad de crear más clasificaciones con el propósito de evaluar de forma más objetiva el marco propuesto.

Architectures for enterprise integration and interoperability Past, present and future

En [156], se hace un repaso de diferentes propuesta para fortalecer la interoperabilidad. La propuesta está planteada en dos partes, en la primera se analizan las iniciativas iniciales para gestionar la interoperabilidad desde una perspectiva de arquitecturas y cuerpos de conocimientos. La segunda parte es un repaso general de propuestas realizadas a partir del año 2000, en donde se enfatiza la evolución de las nuevas propuestas como alternativas para la gestión de interoperabilidad en donde las nuevas propuestas han reflejado una evolución de arquitecturas para

integración empresarial a marcos de trabajo de interoperabilidad, Europa ha destacado mucho en estos esfuerzos.

Según [156], las iniciativas más recientes se han centrado principalmente en marcos para interoperabilidad. La investigación no logra definir lo que debería ser una arquitectura “ideal” para desarrollar sistemas empresariales interoperables.

Los autores concluye con una crítica a las alternativas actuales de interoperabilidad, remarcando que éstas se enfocan en una arquitectura “ideal” que se debería poner el foco en una arquitectura simple basada en una ontología en la que se definieran los conceptos relaciones y propiedades arquitecturales, adicionalmente menciona la necesidad de definir y llegar a acuerdos con lenguajes de representación y técnicas para evaluar dichos lenguajes, adicionalmente definir patrones de diseño y arquitectura enfocados en el diseño de la interoperabilidad. Por último se especifica que aun cuando existe una diversidad de marcos de trabajo y propuestas arquitecturales con distintas tecnologías a nivel de IT, falta según el artículo un nivel arquitectural superior de interoperabilidad que englobe y busque soluciones más generales a los problemas.

An interoperability model for ultra large scale systems

[157] propone como alternativa al problema de interoperabilidad desde la perspectiva del intercambio y comprensión de información entre dos elementos, una solución basada en dos partes. En la primera parte, se presenta un modelo de madurez para la interoperabilidad de sistemas de escala ultra grande, utilizando el nivel de interoperabilidad para definir el nivel de madurez del mismo. En la segunda parte, al proponer un marco, los autores pretenden incrementar la interoperabilidad en sistemas de ultra gran escala en base a los niveles de madurez definidos en la primera parte.

Según [157], para mejorar la interoperabilidad de los sistemas a gran escala, se deben utilizar las tecnologías disponibles que cumplan con los requisitos de interoperabilidad. Por lo tanto, el primer paso en la presentación del marco de inter-

operabilidad es identificar las tecnologías disponibles sobre la base de las cuales se puede fundar el marco de interoperabilidad. Generalmente, para tener un sistema interoperable o aumentar el nivel de madurez de interoperabilidad de un sistema y alcanzar el estado de interoperabilidad deseado, se deben emplear los enfoques disponibles.

Los autores proponen evaluar el nivel de madurez inicial y posteriormente seleccionar tecnologías que apoyen aquellos procesos que requieran una mejora en los niveles de interoperabilidad. Los autores, inicialmente, detallaron las tecnologías de interoperabilidad, posteriormente para cada dominio de interoperabilidad se seleccionó la tecnología adecuada para cumplir con los requisitos de sistemas de ultra gran escala. Finalmente, los autores presentan una tecnología apropiada para cubrir todos los subparámetros presentes en el modelo de madurez de interoperabilidad definido en la primera parte.

Semantic interoperability in eGovernment initiatives

[158] presenta la problemática de la interoperabilidad dentro del mundo del electronic business. La necesidad de permitir las interacciones entre diferentes herramientas y fuentes de conocimiento no solo de una empresa, sino también entre las distintas organizaciones.

El paper hace un repaso de diferentes frameworks aplicables a e-business y presenta el reto que implica la interoperabilidad en el área del software, debido a la rápida evolución que se observa en este mundo adicionalmente se propone la necesidad de hacer tecnologías y marcos de trabajo que sean fáciles de mantener y adaptar a los constantes cambios tecnológicos.

Respecto a los e-business interoperability frameworks se constata que deben tener en cuenta los estándares existentes para estructurar la información y asegurar la consistencia evitando la redundancia. Dentro de esta área, el paper muestra que se han realizado avances en algoritmos que ayuden a resolver el problema de la interoperabilidad. Se han creado data mappings para crear relaciones semánticas y

adaptadores entre herramientas.

Interoperability frameworks and enterprise architectures in e-government initiatives in Europe and the United States

En este trabajo [159] presenta la interoperabilidad como un reto para el e-government que apareció sobre 1980-1990. Prueba de la relevancia de esta problemática es que la interoperabilidad aparece dentro de las estrategias de agenda de diferentes gobiernos. En el artículo se hace un repaso a los frameworks utilizados para intentar minimizar el impacto de la interoperabilidad, dentro de los gobiernos de Estados Unidos y de la Unión Europea.

Según [159], la interoperabilidad dentro del e-governance se ha dividido en dos fases. Una primera fase es la creación y definición de estándares para representar la información de un mismo modo y que esto ayude a ser transmitida y utilizada en diferentes propósitos, y una segunda fase que se centre en la arquitectura para facilitar y alinear la interoperabilidad entre distintas administraciones.

An Overview of Semantic Interoperability Ontologies and Frameworks for IoT

En [160], se puntualiza la importancia de lograr interoperabilidad a nivel de Internet de las Cosas(IoT), los autores realizan una revisión a las alternativas propuestas para lograr la interoperabilidad en IoT, adicionalmente se presentan los desafíos y las soluciones actuales de la interoperabilidad semántica a nivel de IoT. También analiza las limitaciones de las soluciones actuales y se recomienda un camino a seguir.

El trabajo se centra en la interoperabilidad semántica y sus desafíos en el contexto de IoT, se presentan los siguientes desafíos de la interoperabilidad semántica:

- **Modelado e intercambio de datos:** Se refiere la posibilidad de modelar y representar datos de una manera estandarizada, comprensible para las diferentes partes interesadas en el entorno de IoT, considerando que los datos se pueden modelar utilizando un modelo semántico que describe el significado

de sus instancias. El modelado semántico proporciona facilita la interoperabilidad entre diferentes sistemas y aplicaciones en IoT.

- Anotación semántica: Se refiere a detalles proporcionados sobre los datos que se intercambian, como ser de dónde provienen, quien los envía y su relación con su entorno, entre otros.
- Descubrimiento semántico de fuentes, datos y servicios: Se refiere a la necesidad de proveer la búsqueda y el descubrimiento de recursos, entidades y servicios. Adicionalmente se requieren consideraciones de eficiencia energética para los procesos de descubrimiento, dado que los dispositivos de IoT cuentan con recursos limitados.
- Razonamiento e interpretación semántico: La posibilidad de contar con capacidades de razonamiento semántico, es esencial para garantizar el descubrimiento de recursos, por lo que existe la necesidad de mecanismos de procesamiento y razonamiento entre los datos de IoT.
- Representación del conocimiento y ontologías relacionadas: Se refiere a problemas de interoperabilidad ocasionados por la incompatibilidad entre metodologías de representaciones del conocimiento. Se propone el uso de especificaciones de la web semántica para representar formalmente el conocimiento.

Finalmente los autores concluyen que a pesar de contar con ontologías y otras alternativas de interoperabilidad, la visión de IoT aún no se ha logrado. La interoperabilidad de IoT aún se encuentra en su fase inicial, ya que “en las aplicaciones de IoT existentes, los dispositivos no son o son poco interoperables entre sí, ya que sus datos se basan en formatos propietarios y no utilizan términos o vocabulario comunes para describir datos de IoT” [161].

2.8.2. Trabajos relacionados con interoperabilidad en ingeniería de sistemas

A continuación se presentan trabajos relacionados con la interoperabilidad a nivel de ingeniería de sistemas, los trabajos valorados en este apartado constituyen alternativas a resolver problemas de interoperabilidad industrial.

Open Services for Lifecycle Collaboration

La iniciativa Open Services for Lifecycle Collaboration (OSLC) proporciona un conjunto de especificaciones de servicios web, para productos, servicios y otras herramientas que apoyan las distintas fases en el desarrollo de software y el ciclo de vida de productos [19, 162].

Estas especificaciones permiten la integración entre herramientas que contribuyen a la gestión del ciclo de vida de las aplicaciones (ALM) y la gestión del ciclo de vida del producto (PLM). Cada parte del ciclo de vida o dominio tiene su propio grupo y especificación, por ejemplo, gestión de cambios, gestión de calidad, estimación y medición, entre otras. Cada una de las especificaciones de dominio se basa en esta especificación principal. En la Figura 2-13 se presentan la estructura básica de los Servicios de OSLC, estos son accesibles a través de un recurso que describe los servicios ofrecidos. Cada servicio puede proporcionar fábricas para la creación de recursos, capacidades de consulta y diálogos de interfaces de usuario delegadas para permitir a los clientes crear y seleccionar recursos.

OSLC-KM: A knowledge management specification for OSLC-based resources

En [20] se presenta una especificación basada en OSLC, RSHP y RDF como alternativa a la necesidad de gestionar grandes cantidades de conocimientos generados en los distintos procesos de ingeniería de sistemas. Esta especificación propone aprovechar estándares existentes y permitir la gestión de conocimiento dentro del contexto de Ingeniería de Sistemas. La combinación de RDF con un lenguaje de representación universal como RSHP, permiten superar problemas conocidos de RDF como alternativa para la representación

2.8 Trabajos relacionados

Figura 2-13: Conceptos y relaciones de la especificación básica de OSLC [19].

de conocimientos, tales como: (1) falta de vocabularios para representar cualquier tipo de información, (2) falta de servicios nativos comunes como indexación y recuperación de conocimientos. Estas brechas se superan mediante la combinación de RDF con RSHP.

De acuerdo con los autores, es posible abordar el desafío de proporcionar una capa común y estandarizada de gestión de conocimientos en la disciplina de Ingeniería de Sistemas, mediante el uso de servicios basados en RDF y OSLC como interfaz de entrada salida y aprovechando un lenguaje de representación de conocimientos con un enfoque orientado a la industria como RSHP. Adicionalmente en este trabajo se puntualiza la importancia de contar con un mapeo entre cualquier vocabulario RDF y RSHP, ya que actualmente existen múltiples servicios públicos disponibles en RDF. En la Figura 2-14 se presenta un sistema basado en OSLC-KM desarrollado por los autores. En primer lugar se proponen dos posibles entradas, las primeras basadas servicios OLSC y RDF, y las segundas basadas en recursos representados mediante servicios OSLC-KM. En ambos casos, se define una etapa de razonamiento opcional para inferir nuevo conocimiento que puede agregar valor al sistema.

Después de eso, si el RDF está codificado usando el vocabulario OSLC KM, se lleva a cabo un proceso de validación para garantizar que el conocimiento cumpla con el voca-

bulario definido. Por otro lado, si el RDF proviene de un servicio externo, se aplican las reglas de mapeo para generar una representación en su correspondiente RSHP. Finalmente, se realiza un proceso de indexación semántica, en donde se indexa el modelo RSHP en el repositorio de conocimientos del sistema.

El lado derecho de la figura también describe la arquitectura funcional de un conjunto de servicios básicos que funcionan sobre un repositorio de conocimientos. En conclusión, “las tres claves principales de esta arquitectura radican en: 1) la posibilidad de importar y exportar cualquier pieza de Datos codificados en RDF; 2) reutilizar datos / información / conocimiento a través de un conjunto de servicios básicos basados en un repositorio de conocimientos del sistema y 3) permitir a los usuarios utilizar una consulta en lenguaje natural interfaz o incluso RSHP para buscar artefactos” [20].

Figura 2-14: Arquitectura funcional y servicios centrales para la gestión de conocimientos basados en la especificación OSLC-KM [20].

ISO 10303 Standard for the Exchange of Product model data (STEP)

En una Norma ISO, documentada como “ISO 10303 Industrial Automation Systems - Product Data Representation and Exchange, Standard for the Exchange of Product Model

2.8 Trabajos relacionados

Data (STEP)". Tiene como propósito, convertirse en un estándar mundial para la representación e intercambio de datos de productos, para lo cual proporciona un mecanismo capaz de describir los datos del producto a lo largo de su ciclo de vida, de forma completamente independiente de cualquier método de implantación individual [163, 164].

STEP es utilizado para el intercambio de información sobre productos, procesos de fabricación, ensamblaje, especialmente cuando se utilizan herramientas de Diseño Asistido por Ordenador (CAD), Ingeniería Asistida por Ordenador (CAE) o Fabricación Asistida Por Ordenador (CAM). Si bien STEP tiene el potencial de convertirse en una tecnología de información clave en una amplia gama de sectores industriales, el grueso de su documentación y la complejidad de su estructura de datos puede plantear problemas a los usuarios en todos los niveles. En otras palabras, no se conoce hasta qué punto los usuarios industriales y los desarrolladores de software podrán adoptar y explotar el potencial de STEP para aumentar la productividad. Esta incertidumbre se debe en parte a la complejidad del estándar [164].

La parte de STEP que se utiliza más ampliamente en la actualidad es el protocolo de aplicación AP 203 ('Diseño controlado por configuración'). Esto se refiere a la transferencia de modelos de forma de producto, estructura de ensamblaje e información de control de configuración (por ejemplo, control de versiones de piezas, estado de lanzamiento, datos de autorización, etc.) [164].

ISO 10303 (STEP) Modelling and Simulation information in a collaborative Systems Engineering Context (MoSSEC)

La información de modelado y simulación son de suma importancia, sobre todo al momento de realizar validaciones o justificar decisiones de diseño a lo largo del ciclo de vida de un producto, MoSSEC tiene como objetivo, proporcionar un estándar para compartir información de forma independiente de la tecnología utilizada para modelado y simulación [21].

La Edición 1 de MoSSEC, está planificada como un nuevo protocolo de aplicación dentro de STEP(ver 2.8.2) y se basa en la tecnología STEP. Por lo anterior, MoSSEC encaja con

los demás estándares ISO STEP. En la Figura 2-15, se presentan las agrupaciones cubiertas por los objetos comerciales de MoSSEC. Éstos proporcionan el contexto de ingeniería de sistemas para los elementos indicados (por ejemplo, Análisis estructural, Geometría 3D) [21].

Adicionalmente, debido a que LOTAR(ver 2.8.2) utiliza protocolos de aplicación de STEP, los cuales son compatibles con MoSSEC, es posible contar con un mapeo directo entre los objetos de negocio de MoSSEC y los estándares utilizados por LOTAR. Lo que significa que los datos de MoSSEC se pueden convertir automáticamente en los estándares utilizados por LOTAR.

Figura 2-15: Alineación MoSSEC con otros estándares ISO STEP [21].

En conclusión, MoSSEC pretende contribuir a la mejora de la toma de decisiones para productos complejos, así como también compartir el contexto de ingeniería de sistemas (quién, qué, dónde, cuándo, cómo, por qué) de datos de modelado y simulación entre equipos / dominios internos. MoSSEC cuenta con el respaldo de socios industriales (por ejemplo, Airbus, Rockwell Collins, Boeing, BAE Systems) y proveedores (por ejemplo, Eurostep, Dassault Systèmes, MSC Software, Siemens) [165].

INCOSE model-based systems engineering (MBSE) initiative

Es un enfoque emergente en la Ingeniería de Sistemas y puede describirse como la solicitud formalizada de principios, métodos, lenguajes y herramientas de modelado a todo el ciclo de vida de sistemas grandes, complejos, interdisciplinarios y sociotécnicos [22]. INCOSE define MBSE “como una iniciativa que pretende la aplicación formalizada del modelado para respaldar los requisitos del sistema, las actividades de diseño, análisis, verificación y validación que comienzan en la fase de diseño conceptual y continua a lo largo del desarrollo y las fases posteriores del ciclo de vida” [166].

En particular, se espera que MBSE reemplace el enfoque centrado en documentos que han practicado los ingenieros de sistemas en el pasado y que influya en la práctica futura de la ingeniería de sistemas al estar completamente integrado en la definición de procesos de ingeniería de sistemas [166, 167].

MBSE, tiene como objetivo facilitar las actividades de Ingeniería de Sistemas a través del desarrollo de un modelo coherente y unificado. Las principales ventajas potenciales de este enfoque incluyen una mejor comunicación entre las partes interesadas y los miembros del equipo, así como una verdadera comprensión compartida del dominio, una mejor captura de conocimientos, precisión e integridad del diseño sin desconexiones entre las representaciones de datos, adicionalmente se logran mejoras en los niveles de trazabilidad de la información y reutilización de artefactos [22].

Los diferentes aspectos de MBSE como ser, estándares, formalismos, métodos, herramientas de modelado, aplicaciones, etc., se pueden considerar como dimensiones interactivas que deben trabajar juntas para lograr los principales resultados, que es un entorno MBSE capaz de conducir a un sistema exitoso. Este éxito se mide por el cumplimiento de las expectativas de las partes interesadas. En la Figura 2-16 se presenta esta visión integrada.

LOTAR

Consiste en un modelo de referencia basado en ISO 14721 “Sistema de información de archivo abierto (OAIS)”. Los documentos para el estándar se publican en Europa, es-

Figura 2-16: Entorno integrado de MBSE [22].

pecíficamente en la Norma Europea (EN) 9300-xxx bajo ASD-STAN y como el Estándar Aeroespacial Nacional (NAS) 9300-xxx en los EE.UU [23]. Dicho modelo ha sido desarrollado por “LOTAR International que es un proyecto llevado a cabo por Fabricantes de Equipo Original (OEMs) líderes, proveedores de soluciones en la industria aeroespacial y de defensa bajo los auspicios conjuntos de la Asociación de Industrias Aeroespaciales de América (AIA), la Asociación de Industrias Aeroespaciales y de Defensa de Europa para la Estandarización (ASD-STAN)” [23].

El objetivo de LOTAR International es desarrollar, probar, publicar y mantener estándares para el archivo a largo plazo (LTA) de productos digitales y datos técnicos, como datos 3D CAD / CAM y PDM. Estos estándares constituyen el Estándar Lotar y definirán los procesos auditables de archivo, gestión y recuperación, preservando la intención de ingeniería durante todo el ciclo de vida del producto [23].

La Figura 2-17, presenta una descripción de los distintos componentes del estándar Lotar, en donde los mismos se clasifican en las siguientes partes:

- Partes específicas del dominio de datos: Contienen requisitos y definiciones para tipos específicos de datos relacionados con el archivo y la recuperación a largo plazo. Hasta ahora existen descripciones para datos de modelos CAD 3D y datos de gestión de datos de productos (PDM).
- Partes de proceso común: Definen los pasos principales del proceso de archivo y recuperación de datos de estructura de producto y modelo 3D de una manera de-

2.9 Conclusiones del estado del arte

Figura 2-17: Descripción general de LOTAR [23].

tallada y común.

- Partes básicas: Proporcionan una descripción general sobre la información esencial sobre el archivo y la recuperación a largo plazo de datos de estructura de producto y modelo 3D.

Finalmente, respecto a los estándares específicos de dominio utilizados por LOTAR, “LOTAR no define nuevos estándares de modelos de información de productos, pero hace referencia a los protocolos de aplicación STEP ISO 10303. Cuando se identifican las limitaciones de estos estándares, el proyecto LOTAR contribuye a las mejoras de estos estándares. Ejemplo: Contribuciones de LOTAR al proyecto AP 242 y a la armonización para PDM entre AP 242 y AP 239 PLCS” [23].

2.9. Conclusiones del estado del arte

A continuación se detallan conclusiones identificadas después de realizar una valoración a los trabajos relacionados:

- La necesidad de estandarizar los conceptos que subyacen en la definición de interoperabilidad sigue vigente y representa un desafío, proponer un cuerpo de conocimientos formal que unifique las definiciones, herramientas, metodologías de

interoperabilidad podría ser una contribución importante, en [17] se puntualiza esta necesidad y se proponen algunas ideas, que podrían enfocar los esfuerzos por dar respuesta a las necesidades académicas e industriales en lo referente a la interoperabilidad.

- Los trabajos valorados en el presente capítulo, proporcionan alternativas interesantes pero con enfoques concretos, para satisfacer necesidades de interoperabilidad puntuales. En los mismos se proponen el uso de tecnologías disponibles para satisfacer las diferentes necesidades de interoperabilidad abordadas, en los casos en donde se proponen marcos de interoperabilidad de aplicación general, el marco propuesto podría enriquecer estos marcos, apoyando las necesidades de interoperabilidad semántica.
- En ninguno de los trabajos relacionados se propone el uso de un lenguaje común como una alternativa para mejorar los niveles de interoperabilidad, y con el propósito de representar cualquier artefacto de ingeniería. Lo más cercano a la utilización de entornos neutrales, es el uso de los estándares de STEP que ofrecen metodologías y tecnologías maduras, pero tienen la debilidad de ser sumamente complejos de aplicar [164], lo que implica altos niveles de esfuerzo para ser aplicados en ambientes heterogéneos, la mayor parte del antecedente de Step se centra en modelos CAM y CAD salvo MoSSEC que incluye información contextual de simulación.
- En el caso de MBSE, el marco propuesto puede apoyar la visión global del mismo, ya que permite la obtención de una formalización de las diferentes fuentes de información. Adicionalmente la simplicidad al momento de construir la representación de los diferentes artefactos de ingeniería contribuyen a reducir el costo de integrar el marco propuesto dentro de los procesos de una organización.
- Debido a que, para poder llevar a cabo la trazabilidad, es necesario contar con la posibilidad de adquirir, representar, almacenar y recuperar cada elemento de información [9]. El marco propuesto proporciona una alternativa a realizar las operaciones anteriores, ya que se basa en la gestión de una representación unificada, que puede ser gestionada en cualquier momento basándose en un repositorio de cono-

2.9 Conclusiones del estado del arte

cimiento en donde estas representaciones pueden ser almacenadas y recuperadas según se requiera.

Como se aprecia en las conclusiones presentadas anteriormente, a pesar de que contar con múltiples alternativas para abordar los problemas de interoperabilidad, en ninguna de las soluciones propuestas se encontró una alternativa que se basara en una representación unificada y que permitiera su aplicación sobre de cualquier fuente de información. El marco propuesto ofrece tanto una arquitectura conceptual como tecnológica para abordar la interoperabilidad a cualquier fuente de información accesible, sin importar su formato, protocolo, etc. adicionalmente, la representación unificada propuesta en esta tesis, contribuye a mejorar los procesos de trazabilidad y reutilización de conocimientos.

En vista de lo anterior a continuación se detallan algunos aspectos que no han sido abordados por trabajos anteriores y a los que la presente tesis pretende dar solución:

- Representación de cualquier tipo de información y estandarización de formalización: La posibilidad de representar cualquier tipo de información ya sea estructurada o no estructurada, no fue abordada por ninguna de las propuestas (salvo OSLC-KM [20] que es una de las bases del marco propuesto), las opciones de representación unificadas de información obedecían a necesidades de migración entre versiones de herramientas o compatibilidad entre herramientas como es el caso de STEP, en casos como MoSSEC y MBSE se abordó el tema desde un enfoque de unificación de información de modelado y simulación para MoSSEC y la necesidad de formalizar el conocimiento y los modelos como alternativa para el apoyo de las actividades de Ingeniería con una visión de abandonar el enfoque de documentos en el caso de MBSE. El marco propuesto ofrece la posibilidad de representar cualquier tipo de información. Adicionalmente, si bien es cierto OSLC-KM [20] ofrece la posibilidad de representar diversos artefactos, dentro del mismo no se aborda el problema de la estandarización de estos modelos de representación, es decir que utilizando OSLC-KM para representar información por ejemplo de máquinas de estados, las representaciones de dos fuentes de información pueden ser diferentes, lo cual ocasionaría dificultades al momento de interoperar entre herramientas

aun cuando éstas estén representadas por el mismo lenguaje, en el caso del marco propuesto este problema se mitiga con el uso de los `MappeableElements` (ver sección 3.3.4), cabe resaltar que OSLC-KM es un trabajo utilizado en esta tesis (Ver apartado 2.6.3).

- Interoperabilidad a nivel de operaciones: La posibilidad de invocar funcionalidad que está disponible en las fuentes de información es un aspecto interesante al momento de abordar la interoperabilidad, el marco propuesto permite en caso de que las APIs proporcionadas por los fabricantes, ofrezcan la posibilidad invocar funcionalidad, utilizar esta información, por ejemplo ejecutar la función `CheckModel` ofrecida por IBM Rational Rhapsody sobre modelos SysML que existen en MagicDraw.
- Servicios nativos de indexación, transformación y recuperación: En ninguno de los trabajos relacionados se presenta la posibilidad de contar con servicios de indexación, transformación y recuperación nativos, en el caso del marco propuesto en esta tesis, considerando que en su arquitectura tecnológica incluye un repositorio de conocimiento sobre las tecnologías TRC, es posible contar con un repositorio de indexación, recuperación, así como también la posibilidad de realizar operaciones de trazabilidad Verificación y Validación entre otros.

3 Definición de métodos para la interoperabilidad en procesos de ingeniería

“Solamente aquel que construye el futuro tiene derecho a juzgar el pasado”

Friedrich Nietzsche

En el caso de objeto de estudio de esta tesis, se propone la aplicación de un lenguaje unificado para mejorar la interoperabilidad de distintas fuentes de información, esto con el propósito de lograr la comunicación entre las mismas, abstrayéndose de los protocolos, formatos y fuentes.

La aplicación de un lenguaje unificado para representar múltiples artefactos de conocimiento, presenta diversos retos como ser:

1. Acceso a la información: Para poder definir una representación formal del conocimiento, es necesario acceder al mismo, entenderlo y representarlo en un lenguaje unificado, esto puede constituirse como un verdadero reto en aquellas plataformas que gestionan el conocimiento de forma muy propietaria y en donde los proveedores de herramientas no ofrecen tecnologías de interoperabilidad muy maduras.
2. Estandarización en modelos de representación: El problema de la representación del conocimiento radica en que es posible representar el mismo artefacto de diferentes formas, lo cual ocasiona un problema grave al momento de procesar las

formalizaciones, ya que un mismo artefacto podría tener múltiples representaciones.

Considerando los problemas antes mencionados, en este capítulo se propone marco de interoperabilidad inspirado en MOF que se denomina conectores SRL, donde es posible la creación de modelos auto-contenidos, es decir que los distintos metamodelos pueden ser interpretados, gobernados y estandarizados.

Adicionalmente, se describe el lenguaje de representación del conocimiento propuesto, se presenta una especificación detallada del modelo de clases, así como también de cada elemento que compone el modelo de clases, ejemplos de representación de diversos tipos de información, como la estructurada y no estructurada. Se presentan una metodología para el desarrollo del marco propuesto. Por ultimo se presentan conclusiones respecto al marco de interoperabilidad propuesto.

3.1. Descripción de los conectores SRL

Para proponer una solución de interoperabilidad que permita la integración de múltiples herramientas, en la Figura 3-1: Se presenta la idea inicial en donde se muestran los conceptos principales que derivaron en el diseño de la arquitectura conceptual presentada en la sección 3.3. En primera instancia, fue necesario considerar el lenguaje de representación, que contara con características como: simplicidad, flexibilidad, adaptabilidad, esto en aras de integrar la mayor cantidad de herramientas posibles de una forma simple.

Por otro lado, una de las necesidades más importantes es la de estandarizar las representaciones de los contenidos, es decir, garantizar que si dos fuentes generan el mismo tipo de conocimientos y este contenido es representado con un lenguaje único, esta representación sea la misma, o en su defecto totalmente compatible, esto con el propósito de evitar complicaciones que se tienen actualmente en lenguajes como UML, en donde las herramientas que producen UML tienen problemas de interoperabilidad entre ellas [168].

Otro aspecto relevante para considerar, es la posibilidad de acceder a la información fuente, se debe tener en cuenta como se menciona anteriormente, que no siempre se con-

3.1 Descripción de los conectores SRL

Figura 3-1: Concepto inicial del marco propuesto.

tará con tecnología para interoperar con las herramientas que generan la información fuente. Por otro lado, si las tecnologías de interoperabilidad lo permiten se debe poder invocar a funcionalidades que ofrecen las herramientas sobre las cuales se desarrollarán los diferentes conectores SRL, por ejemplo, si se desarrolla un conector SRL a Simulink utilizando las APIs de Matlab y éstas permiten invocar funcionalidades de simulación, el conector debe ofrecer esta funcionalidad a cualquier plataforma que consuma el conector SRL.

Contar con representaciones del conocimiento, estándares y bajo un lenguaje de representación unificado, permite realizar una gestión del conocimiento en términos de reutilización de las diferentes representaciones de conocimientos extraídas de las fuentes de información y de forma independiente de su formato original, adicionalmente la posibilidad de gestionar trazas, desde la creación manual de trazas a la posibilidad de construcción de algoritmos de sugerencias automáticas de trazas, las posibilidades de aplicar transformaciones al conocimiento, es decir generación de conocimiento nuevo a partir de las representaciones obtenidas de la lectura del conocimiento obtenido mediante los diferentes conectores SRL, asimismo utilizar artefactos representados para actividades de validación y verificación.

Considerando los elementos anteriores se proponen las arquitecturas conceptual y tecnológica de del marco de interoperabilidad, denominado “conectores SRL”, adicionalmente se propone como lenguaje de representación unificado el SRL, utilizando como base los trabajos propuestos por [1, 11, 20].

3.2. Paradigma del marco de interoperabilidad para procesos de ingeniería

La idea que subyace en el marco de interoperabilidad definido en esta tesis, consiste en crear un entorno neutral que permita realizar transformaciones de cualquier fuente de información a este entorno neutral y viceversa. Este entorno neutral debe contar con capacidades de gestión de contenido y operaciones. Para explicar mejor esta idea, en la Figura 3-2 se presenta un entorno de tres (3) herramientas o fuentes de información, todas las herramientas de este entorno están conectadas, es decir existe una interoperabilidad total, partiendo de este supuesto se puede apreciar lo siguiente:

- Cada herramienta cuenta con dos conexiones por fuente de información, es decir la primera desde la herramienta de origen hacia el destino y una segunda desde el destino hacia el origen.
- Por cada conexión se pretenden compartir artefactos (información), accesos y operaciones.
- Se cuenta con un total de conectores de $n(n-1)$.

Considerando lo anterior, el costo para para agregar una nueva herramienta al entorno descrito en la Figura 3-2, sería de $2(n)$ si se quisiera conservar un entorno totalmente conectado. Es decir que en caso de agregar una herramienta D, se tendrían que desarrollar conectores individuales desde la nueva herramienta a las herramientas existentes, lo que en el caso del ejemplo implica el desarrollo de 6 conectores nuevos.

en la Figura 3-3 se presenta el mismo ejemplo, pero considerando marco de interoperabilidad propuesto, como un entorno neutral, en este caso no existen conexiones directas entre las herramientas, existe únicamente conexión desde la herramienta hacia el entorno

3.3 Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

Figura 3-2: Ejemplo interoperabilidad entre herramientas sin aplicar el marco propuesto.

neutral. Considerando el ejemplo de la Figura 3-3 se puede apreciar lo siguiente:

- Cada herramienta únicamente cuenta con conectividad hacia el entorno neutral.
- Al igual que en el ejemplo presentado en la Figura 3-2, se pretenden compartir artefactos (información), accesos y operaciones.
- Se cuenta con un total de conectores de $n(n-1)$.

La diferencia principal radica en el caso de agregar una nueva herramienta a este entorno, bajo el marco propuesto, la integración de una Herramienta D en el ejemplo, implica únicamente el desarrollo de dos conectores, es decir desde la Herramienta D hacia el entorno neutral y desde el entorno neutral hacia la herramienta. Esto reduce considerablemente los costes de integración de nuevas fuentes de información al entorno.

3.3. Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

La Figura 3-4 muestra la arquitectura conceptual de los conectores SRL con sus respectivas capas, se proponen cuatro (4) capas, dentro de las cuales se enmarcan los elementos que deberán proporcionar los mecanismos para representar de forma auto contenida el conocimiento de las distintas fuentes de información sobre las que se desarrollarán los

3 Definición de métodos para la interoperabilidad en procesos de ingeniería

Figura 3-3: Ejemplo interoperabilidad entre herramientas aplicando el marco propuesto.

conectores SRL.

Figura 3-4: Arquitectura conceptual del marco propuesto.

En la Figura 3-4 se presentan los elementos que conforman los conectores SRL, se

3.3 Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

utilizan cuatro (4) capas, que son las que proporcionan los elementos necesarios para interoperar entre las herramientas fuentes. En la capa Modelos SRL, se muestran modelos, lo cual constituyen representaciones formales obtenidas de la interpretación semántica de las fuentes de conocimiento que se encuentran en el nivel de las fuentes de información. Dentro de la capa Especificación SRL, están los elementos de SRL, que constituyen todos los componentes necesarios para representar conocimientos, la posibilidad de formalizar una representación que exprese la semántica, la sintaxis y la pragmática del conocimiento obtenido de un proceso de interpretación semántica y posterior representación de la información fuente. Los Mappable Elements son los elementos que gobiernan como se utiliza la simbología disponible en la Especificación SRL. Es decir, son los componentes responsables de generar la representación del conocimiento, éstos permiten crear los modelos de forma estándar, garantizan que el tipo de conocimiento se represente de la misma forma, independientemente de su fuente.

De esta forma teóricamente, es posible construir un módulo completo de interoperabilidad que permita gestionar conocimiento desde las fuentes de información, construyendo modelos estandarizados, mediante el lenguaje de representación SRL, todo gracias a la abstracción de los artefactos de conocimientos en Mappable Elements.

3.3.1. Información fuente

Los procesos de ingeniería de sistemas pueden involucrar una gran cantidad de herramientas. Éstas, generan conocimiento con diversidad de formatos, características y propósitos, los cuales contribuyen de alguna manera a los procesos de ingeniería. Las grandes cantidades de información vuelven la reutilización

Desde la perspectiva de los conectores SRL, la capa “Información Fuente” se refiere al conocimiento contra el que se pretende interoperar, se constituyen todas aquellas fuentes de información, es decir la realidad, el contenido en sus formatos y protocolos originales, por ejemplo, los archivos slx de Simulink, o los archivos rpy de Rhapsody. Este conocimiento puede ser estructurado, como ser modelos, diagramas, planos, diseños y no estructurado como archivos de Excel, txt o Word. Cualquier información independiente

del formato, que se considere relevantes dentro de los procesos de ingeniería, conforma esta realidad que debe ser modelada. Este modelado se lleva a cabo siempre y cuando el conocimiento a modelar pueda ser accedido, ya sea a nivel de APIs o incluso haciendo procesos de interpretación semántica sobre los archivos originales que almacenan la información.

Durante el proceso de ingeniería de sistemas en una cadena de suministros, son muchas las herramientas que interactúan en las diferentes fases de la misma.

3.3.2. Modelos SRL

En esta capa se realiza una representación de la información utilizando el SRL (ver Tabla 3-1 y Figura 3-7) para modelar el contenido de una fuente de información mediante grafos semánticos. SRL se basa en la premisa de que “todo conocimiento puede ser representado mediante grafos”. Es posible incluso representar información textual haciendo uso de patrones como se muestra en la Figura 3-5

Figura 3-5: Ejemplo de representación de información textual en SRL, utilizando un patrón para a generación de grafos semánticos.

En el caso de la información estructurada, su representación mediante SRL es más natural, ya que los modelos en la mayoría de los casos son grafos con tipologías y características definidas. La Figura 3-6. muestra un ejemplo de un modelo de Simulink.

Este modelo con las tipologías de nodos y relaciones es un grafo en sí mismo, por ende, es muy sencillo de representar utilizando SRL. Al momento de modelar en SRL

mo los puertos, incluso estos bloques pueden ser subsistemas. Teniendo en cuenta que los bloques contienen información relevante y que incluso puede ser el caso de que estos bloques contengan modelos completos, es mejor representar cada bloque como un *artifact*.

3.3.3. Especificación SRL

En base a los trabajos [1, 20], se propone SRL como alternativa de representación unificada de conocimiento, para gestionar el conocimiento que se produce en las diferentes fases de los procesos de ingeniería. Esto constituye la capa “Especificación SRL” de la metodología de interoperabilidad el cual es el lenguaje utilizado para modelar la realidad.

Previamente se han desarrollado *resource shapes*, para representar diversos tipos de conocimientos, como ser: requisitos [169], activos [170], calidad [171], casos de prueba [172], cambios [173] y métricas de estimación y medición [174], entre otros. Asimismo, OMG propone la especificación *Open Lifecycle for Collaboration for Model Base Systems Engineering (OSLC-MBSE)* [175] para promover modelos de sistema. Aun así existe tipos de conocimiento para los cuales no se cuenta con especificaciones concretas para su representación, como ser patrones, tablas, etc. En vista de lo anterior se propone SRL como alternativa para representar cualquier tipo de artefacto de conocimientos. SRL se basa en la representación del conocimiento mediante grafos semánticos como premisa para la representación de cualquier tipo de artefacto. Este es un método muy utilizado [176–180], debido a que proporcionan información estructurada eficaz y han sido recursos cruciales para varias aplicaciones inteligentes, incluida la búsqueda web [181–183].

SRL proporciona entre otras cosas un entorno sencillo de aplicar, con una gramática muy flexible y en continua evolución, durante la realización de esta tesis, se han realizado mejoras, enriqueciendo sus propiedades nativas y capacidades debido a las necesidades de representación que han ido surgiendo, a partir de la heterogeneidad de artefactos que se han ido representando a lo largo de la actividad doctoral. Asimismo, se ha comprobado la efectividad del enfoque de representación basado en los grafos semánticos. Esto junto a la facilidad de aplicación, han sido factores sumamente beneficiosos en lo que se

3.3 Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

refiere al tiempo y esfuerzo que se requiere para la aplicación del marco de interoperabilidad propuesto, el hecho de aplicar especificaciones complejas para representación puede representar un verdadero desafío en cuanto al tiempo y esfuerzo que se requiere para su aplicación.

Los principales elementos de SRL son los componentes *Artifact* y *Relationship*. Un *Artifact* es un componente que permite representar nodos, contener otros componentes ya sea por composición o agregación, los artefactos pueden contener *Relationships*, *Properties* (información que se refiere única y exclusivamente a contenido, como ser color=rojo, etc.) y *Metaproperties* (información que no necesariamente es contenido como ser fecha de modificación, usuario que modificó, etc.). También es posible que un *artifact* apunte a un *Term* (elemento de vocabulario de dominio).

Los *Terms* pueden contar con una tipología que permite indicar su etiqueta sintáctica, como ser: verbo, nombre, abreviación, etc. De igual forma estos términos pueden pertenecer a grupos semánticos (Tipo), lo que permite la definición de contexto a la terminología, es decir pueden definirse un grupo semántico de tipo *System*, que cuando se refiera a al dominio ferroviario tenga un significado y cuando se refiera al dominio aeroespacial otro. por último, es posible definir relaciones n-arias, y especificar cardinalidad entre sus nodos. La Figura 3-7 muestra el diagrama de clases UML con el metamodelo SRL, este metamodelo está disponible en forma de una biblioteca de clases de .Net y un proyecto de Java, ambos están disponibles para descarga en [185].

La Tabla 3-1 muestra el detalle de los elementos que conforman el metamodelo de SRL.

A continuación se presenta un ejemplo de equivalencia entre los componentes de Simulink y SRL3-2.

3.3.4. Interpretación de modelos SRL

Uno de los problemas más difíciles de afrontar al momento de representar el conocimiento utilizando *resource shapes*, es la posibilidad de representar el mismo conocimiento de múltiples formas. Al no existir una única forma de representar el conocimiento, la mis-

Tabla 3-1: Descripción de los elementos del metamodelo SRL

Clase de SRL	Descripción
<i>Artifact</i>	Un elemento contenedor, permite agrupar términos, relaciones, propiedades, metapropiedades y otros <i>artifacts</i> , este elemento permite definir la tipología de los <i>artifacts</i> . Dentro de los elementos contenidos en los <i>artifacts</i> es posible definir un orden que facilite la sintaxis del conocimiento representado.
<i>Type</i>	Permite la definición de tipología a los <i>Terms</i> , <i>Artifacts</i> y <i>Relationships</i> .
<i>Term</i>	Se puede definir como una gramática que se puede dotar de sintaxis y pragmática con la clase de <i>Type</i> .
<i>Relationship</i>	Permite establecer relaciones entre elementos, estas relaciones pueden ser binarias o n-arias y facilitar la definición de cardinalidad.
<i>Data</i>	Permite la representación del contenido de la información utilizando propiedades de tipo clave / valor, por ejemplo: Color = Rojo o velocidad 100, esta clase facilita la representación de propiedades incluso cuando tienen más de un valor, por ejemplo, velocidad = 100 Kms, En este último caso, hay dos valores, la velocidad y la unidad de medida de esta velocidad.
<i>Metadata</i>	Tiene una estructura similar a la de la clase de datos, pero con la diferencia de que esta clase está concebida para almacenar solo metainformación y no información referente al contenido, por ejemplo: Última fecha de modificación.

3.3 Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

Figura 3-7: Metamodelo SRL.

Tabla 3-2: Ejemplo de equivalencia entre elementos de Simulink y SRL.

Elemento de Simulink	Elemento SRL
Modelo	Artifact
Bloque	Artifact
Tipo de bloque	Artifact Type
Nombre del bloque	Artifact Name, noun term
Propiedades del bloque	Metaproperties
conexión de tipo línea	RSHP

ma realidad podría ser modelada de formas diferentes aun con el mismo *resource shape*. Con el propósito de gobernar estas representaciones al menos en aquellos *artifacts* que tienen la misma tipología, se han desarrollado los SRL Mappable Elements. Estos consisten en una abstracción del conocimiento común entre distintas fuentes a un nivel más general, es decir este elemento permite estandarizar los diferentes modelos SRL, por ejemplo, requisitos, en la realidad pueden existir requisitos en Rhapsody, Matlab, DOORS, etc. Un Mappable Element de requisitos garantiza una única representación de SRL de forma independiente de la herramienta que originó dicho requisito.

Figura 3-8: Diseño de Mapeable Elements.

Por ejemplo, mediante los Mapeable Elements es posible garantizar que las máquinas de estado de Simulink se representen exactamente de la misma forma que las máquinas de estado de Rhapsody y por ende transferir conocimientos de una plataforma a otra sería posible siempre y cuando las APIs de las herramientas permitan la creación de nuevo conocimiento. En la Figura 3-8 se muestra cómo se utilizan los Mapeable Elements dentro de la arquitectura de representación de información, las fuentes de información crean instancias de los Mapeable Elements y gestionan su información.

Interacción de elementos Mapeable Elements dentro de los conectores SRL

La Figura 3-9 presenta la interacción que tienen los Mapeable Elements dentro de los conectores SRL. Cada Conector SRL cuenta con un componente de parseo, el cual es el encargado de interactuar con la APIs de interoperabilidad de los fabricantes. Este com-

3.3 Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

ponente debe tener la capacidad de identificar los tipos de conocimiento, instanciar el Mapeable Element que corresponde a cada tipo de conocimientos cuando se estén realizando operaciones de lectura de la fuente de información, en otras palabras se realiza una transformación del conocimiento que se obtiene mediante las APIs de interoperabilidad y que está representado en el modelo de clases de estas APIs, a Mapeable elements, posteriormente se retorna el modelo SRL que proporcionan los Mapeables Elements instanciados.

Figura 3-9: Interacción entre Mappeable Elements y conectores SRL.

La invocación a las capacidades de interoperabilidad en los conectores SRL desde las plataformas que hacen uso de los mismos, se hace a través de las distintas funciones y operaciones que ofrecen de forma nativa todos los conectores SRL, estas funciones y operaciones únicamente aceptan y ofrecen como parámetros de entrada y salida modelos SRL, la primera operación que realiza el Conector SRL cuando una de sus funciones es invocada, es utilizar los parámetros recibidos para construir a partir de ellos los objetos Mappeable Elements que corresponden, estos posteriormente son leídos son enviados al componente de parseo para su respectivo tratamiento dependiendo de la operación invocada, por ejemplo si se pretende crear una máquina de estados completa con todos sus estados, transiciones y parámetros, una vez invocada desde afuera del Conector SRL la función correspondiente a creación de Workproducts, esta función creará una instancia del Mappeable Element con el modelo recibido y posteriormente la enviará al componente de parseo, dicho componente, hará la transformación de Mappeable Elements al Metamodelo que se maneja en la fuente de información.

El proceso anterior garantiza que siempre y cuando en el Conector SRL se reciba un modelo SRL con la información suficiente para construir el Mapeable Element correspondiente, esta información podrá ser procesada de forma independiente de su origen, en otras palabras, si construimos una máquina de estados en Visio utilizando Mapeable Elements, es posible replicar exactamente la misma máquina de estados e, Rhapsody, Magic Draw, Simulink o cualquier otra herramienta.

Arquitectura de Mapeable Elements

En la Figura 3-10 se presenta la arquitectura que deben tener los Mapeable Elements, se detallan las capas con sus respectivos componentes, esta arquitectura es conceptual, en ella se definen componentes que se consideran importantes a la hora de utilizar los Mapeable Elements, la mayoría son opcionales, salvo la capa de representación y la Capa de Transformación específicamente el componente de generación a SRL, ya que de acuerdo al marco de interoperabilidad propuesto en esta tesis, la gestión de los Modelos en SRL debe ser realizada siempre mediante los Mapeable Elements, a continuación se detallan los elementos que conforman la arquitectura presentada en la Figura 3-10:.

1. Capa de representación: Se refiere a todas las propiedades que permiten realizar la representación de la información, es utilizando estas propiedades que el conocimiento obtenido de las fuentes es formalizado:
 - Relaciones: Este componente permite definir un set de relaciones binarias o n-arias con su respectiva tipología. Los nodos de estas relaciones deben ser objetos de tipo Mapeable Element.
 - Ocurrencias: Esto permite establecer un subelemento, pero a nivel de agregación, es se pretende ofrecer la posibilidad de representar la ocurrencia de un elemento sin necesidad de establecer pertenencia, la ocurrencia debe ser un elemento de tipo Mapeable Element.
 - Sub elementos: Se refiere a una colección de subelementos, esta propiedad permite definir composición entre elementos.
 - Propiedades nativas: Estas están relacionadas directamente con la naturaleza

3.3 Arquitectura conceptual del marco de interoperabilidad para procesos de ingeniería

de la información que se representa en el Mappable Element, por ejemplo en un requisito, una propiedad nativa puede ser el texto del requisito, otras pueden ser, el texto del requisito en formato RTF, o en formato HTML.

- Otras propiedades: Se refieren a información que no necesariamente es de carácter obligatorio, por ejemplo, algún atributo añadido a los metamodelos de SysML por el usuario.

2. Capa de transformación: Se refiere a las transformaciones que se realizan, utilizando la información de la capa de representación, es en esta capa en donde se definen serializaciones como ser Json o XML, adicionalmente esta capa es la que permite la generación de los modelos SRL a partir de capa de representación.

- Generación Modelo SRL: Esta propiedad permite generar una formalización en lenguaje SRL de la información definida en la Capa de Representación, estos modelos se deben generar siempre de forma uniforme y a demanda.
- Generación a otros Modelos: Este componente está pensado para permitir generar la información en otros metamodelos que sean de interés, se deja a nivel conceptual para que la persona que desarrolle Mappable Elements basándose en el marco propuesto en esta tesis considere de interés.
- Este componente se refiere a la posibilidad de serializar el contenido de los Mappable Elements a diferentes formados como ser XML y Json.

3. GUI: En esta capa se integran las GUIs en formato de controles de usuario que permitan la ejecución de información puntual o la construcción de elementos.

- Gestión de información: Este componente está pensado para definir Interfaces de usuario para hacer operaciones CRUD sobre el contenido de los Mappable Elements, es decir aquí se debería definir una GUI que sepa cómo crear nuevo conocimiento o modificar conocimiento existente, representado en el Mappable Element.
- Ejecución: En caso de que el Mappable Element tenga la posibilidad de ser ejecutado(ej: compones simulables, funciones matemáticas, entre otros) este componente está pensado para ofrecer la interfaz de usuario que permita la

realización de esa ejecución.

Figura 3-10: Arquitectura de Mapeable Elements.

3.4. Definición del proceso para la implementación de conectores SRL

El desarrollo de un conector SRL debe llevarse a cabo de forma secuencial, siguiendo diferentes etapas que faciliten la forma en que cada conector debe ser desarrollado, a continuación se presenta en la Figura 3-11 las fases de la “Metodología de Desarrollo de conectores SRL”.

Como se muestra en la Figura 3-11, la metodología para el desarrollo de los conectores SRL se divide en cuatro etapas: Análisis, Preparación y diseño, Desarrollo de conector

3.4 Definición del proceso para la implementación de conectores SRL

Figura 3-11: Metodología de desarrollo de conectores SRL.

y finalmente Integración, estas etapas son secuenciales e iterativas, en cada iteración es posible ir enriqueciendo más el conector, agregando nuevas capacidades de gestión de artefactos y funcionalidades que ofrece la fuente de información.

Esto significa que es posible ir incrementando los niveles de interoperabilidad en los conectores de acuerdo a las necesidades. En los siguientes apartados se detallan cada una de las fases con sus respectivos procesos y herramientas.

3.4.1. Análisis

Esta es la fase inicial, en ella se estudian las tecnologías de interoperabilidad así como la tipología de información que se gestiona en la fuente con la que se pretende interoperar, se analiza a detalle las capacidades y limitaciones de interconexión así como aspectos concernientes a compatibilidad tecnológica, gestión de la información entre otras cosas. Adicionalmente se determinan las funcionalidades que provee la fuente y podría ser relevante exponer en el conector, con el propósito de ser utilizada por cualquier plataforma

que instancie el conector SRL.

Entradas

A continuación, se listan las entradas necesarias para el inicio de la fase de análisis:

1. Kit de Desarrollo de Software (SDK): Es necesario contar con las herramientas que proveen los fabricantes para extender o invocar funcionalidades de las herramientas que producen el conocimiento sobre el que se requiere interoperar, es necesario estudiar y evaluar las posibilidades que ofrecen las tecnologías contenidas en los SDK, entender su uso en aras de desarrollar los componentes necesarios en el Conector SRL, lo que en la mayoría de los casos implica transformaciones o manipulación de los modelos de objetos ofrecidos por los distintos SDKs y los componentes de SRL.
2. Documentos de contenidos gestionados por la herramienta: Toda la documentación que describa el conocimiento gestionado por las herramientas es fundamental, con el propósito de determinar el conocimiento gestionado por las fuentes y que serán expuestos en el conector SRL .
3. Licencia de pruebas: Es necesario, previo al inicio del desarrollo de los componentes de los conectores SRL, contar con un entorno de desarrollo en el que se tenga la herramienta fuente sobre la que se desarrollará el conector. Esto es necesario para poder realizar el desarrollo haciendo uso de los SDK proporcionados por los fabricantes.

Herramientas

Durante el proceso de análisis se cuenta con ciertas herramientas que facilitan la identificación de los elementos críticos a identificar en las distintas tecnologías de interoperabilidad:

1. Check list con funciones principales a identificar en los SDK (conexión, creación, modificación y recuperación de elementos, etc.).

3.4 Definición del proceso para la implementación de conectores SRL

Procesos

El detalle de procesos que se deben llevar a cabo durante esta fase es el siguiente:

1. Estudiar, analizar y entender las posibilidades que ofrece el SDK (conexión, creación, modificación, recuperación y eliminación de elementos).
2. Desarrollar códigos de prueba para las operaciones del SDK.
3. Determinar los distintos artefactos de conocimiento gestionados por la fuente, con su respectivo modelo de objetos.

Salidas

1. Mecanismos de interoperabilidad seleccionados y sustentados, en este punto se debe hacer una selección en caso de que el SDK ofrezca múltiples posibilidades de interoperabilidad, por ejemplo, dos APIs, una Rest y una DLL de .Net, en este caso la salida implicaría, previo al estudio de la posibilidades que ofrece el SDK, que tecnología es la que es más adecuada para utilizar desde la perspectiva del conector SRL.
2. Informe incluidos códigos de prueba con las funcionalidades necesarias para hacer la gestión del conocimiento en la fuente utilizando el SDK (altas, bajas, modificaciones, eliminaciones, conexión, operaciones).
3. Informe con la tipología de información gestionada por la fuente.

3.4.2. Preparación y diseño

En esta etapa se desarrolla uno de los componentes más importantes del conector SRL, tiene como propósito tanto el desarrollo o modificación de los Mappable Elements (ver sección 3.3.4) en función de los elementos de conocimiento identificados en la fase anterior, es decir, supóngase que se está desarrollando un conector para la gestión de requisitos en Teamcenter, es posible que ya se cuente con Mappable Elements desarrollados para conectores SRL anteriores para la gestión de requisitos, en esta fase es necesario garantizar que ese Mappable Element existente, cumpla con lo necesario para almacenar la

información de los requisitos de Teamcenter, el cual es un software de gestión de ciclo de vida de los productos (PLM) [184], si el mismo no cuenta con los atributos necesarios es para la gestión de los requisitos de Teamcenter, será necesario ajustarlo para que además de soportar los conectores anteriores, dicho Mapeable Element soporte los atributos y características del nuevo conector, en caso de no tener Mapeable Element para requisitos, en esta etapa se desarrollará y posteriormente será utilizado cuando se desarrollen conectores SRL que requieran representar requisitos.

Adicionalmente se desarrolla un componente de parseo que es el que interactúa con el SDK de la fuente, este componente es la base de la siguiente etapa que es cuando se desarrolla el conector SRL. En otras palabras, las funciones y métodos del conector realizarán invocaciones a este componente de parseo para interactuar con la fuente.

Entradas

Las entradas necesarias para el inicio de la fase de análisis son las siguientes:

1. Mecanismos de interoperabilidad seleccionados y sustentados, en este punto se debe hacer una selección en caso de que el SDK ofrezca múltiples posibilidades de interoperabilidad, por ejemplo, dos APIs, una Rest y una DLL de .Net, en este caso la salida implicaría, previo al estudio de la posibilidades que ofrece el SDK, que tecnología es la que es más adecuada para utilizar desde la perspectiva del conector SRL.
2. Informe incluidos códigos de prueba con las funcionalidades necesarias para hacer la gestión del conocimiento en la fuente utilizando el SDK (altas, bajas, modificaciones, eliminaciones, conexión, operaciones).
3. Informe con la tipología de información gestionada por la fuente.
4. Repositorio Mapeable Elements existentes.

Herramientas

1. SDK para el desarrollo de Mapeable Elements.

3.4 Definición del proceso para la implementación de conectores SRL

Procesos

A continuación, el detalle de procesos que se deben llevar a cabo durante esta fase:

1. Analizar los Mapeable Elements existentes, con el propósito de validar que existen todos los Mapeable Elements necesarios para gestionar la tipología de información gestionada en la fuente.
2. Desarrollar o modificar Mapeable Elements para la gestión de información fuente.
3. Desarrollar componente *parsing* que realizará las invocaciones a la herramienta fuente desde las llamadas desarrolladas en el conector SRL.

Salidas

1. Mapeable Elements para el soporte de la información gestionada por la fuente.
2. Componente parsing con sus respectivos test unitarios.

3.4.3. Desarrollo de conector SRL

En esta fase se realiza el desarrollo tecnológico de conectores SRL [185]. implica la gestión de contenido, invocación de operaciones, gestión de la configuración, seguridad y trazabilidad y todas las capacidades que serán incluidas en el desarrollo tecnológico.

Adicionalmente esta interfaz permite la creación de Interfaz Gráfica de Usuario (GUIs) para: Configuración de conexión a la fuente, edición y modificación y visualización de información. Estas GUIs deben ser desarrolladas en forma de controles de usuario que serán invocados dinámicamente por la aplicación que haga uso de los conectores SRL.

Entradas

Las entradas necesarias que sirven como punto de partida para la realización de la fase de análisis son las siguientes:

1. Mapeable Elements para el soporte de la información gestionada por la fuente.
2. Componente parsing con sus respectivos test unitarios.

Herramientas

1. SDK de SRL con las clases, interfaces, extensiones y utilitarios para el desarrollo de conectores SRL.

Procesos

A continuación el detalle de procesos que se deben llevar a cabo durante esta fase:

1. Implementar la interface ISource2SRL, con las funciones definidas de acuerdo al alcance del conector SRL, con sus respectivos test unitarios.

Salidas

1. Conector SRL basado en la interface ISource2SRL.

3.4.4. Integración

Esta es la última fase en donde se pretende integrar el conector SRL desarrollado en la fase anterior en la plataforma que consumirá dicho conector, la mayoría de las veces que un nuevo conector SRL se integra en un consumidor(que puede ser un API o una aplicación concreta), basta con instanciar la clase del nuevo conector SRL, Los conectores además de contar con las funciones para la ejecución de interoperabilidad, puede proporcionar información como ser nombre, fabricante del conector, descripción, logo, entre otros.

Entradas

A continuación se listan las entradas necesarias para el inicio de la fase de análisis:

1. Conector SRL basado en la interface ISource2SRL.

Procesos

1. Crear una instancia dentro del consumidor del conector SRL, con su respectiva opción que permita a los usuarios invocar el conector contra una o varias fuentes

3.4 Definición del proceso para la implementación de conectores SRL

de información.

Salidas

1. Consumidor con opción disponible para instanciar el conector SRL desarrollado.

3.4.5. Resumen de metodología de desarrollo de conectores SRL

A continuación se presenta una matriz resumen (ver 3-3) que contiene las fases de la metodología de desarrollo de conectores SRL expresadas en forma de columna, en las filas las correspondientes entradas, procesos, salidas y herramientas por fase.

Tabla 3-3: Resumen de metodología de desarrollo de conector SRL.

	Análisis	Preparación y diseño	Desarrollo de conector	Integración
Entradas	<ul style="list-style-type: none"> -SDK herramienta origen. -Documento de contenidos gestionados por las fuentes. -Licencia de software fuente. 	<ul style="list-style-type: none"> -Mecanismos de interoperabilidad seleccionados y sustentados. -Informe incluidos códigos de prueba con las funcionalidades necesarias para hacer la gestión del conocimiento en la fuente utilizando el SDK. -Informe con la tipología de información gestionada por la fuente. -Repositorio con Mapeable Elements existentes desarrollados para otros conectores. 	<ul style="list-style-type: none"> -Mapeable Elements para el soporte de la información gestionada por la fuente. -Componente parsing con sus respectivos test unitarios. 	<ul style="list-style-type: none"> -Conector SRL basado en la interface ISource2SRL.
Herramientas	<ul style="list-style-type: none"> -Lista de verificación con funciones principales a identificar en los SDK 	<ul style="list-style-type: none"> -SDK para el desarrollo de Mapeable Elements. 	<ul style="list-style-type: none"> -SDK de SRL con las clases, interfaces, extensiones y utilitarios para el desarrollo de conectores SRL. 	
Procesos	<ul style="list-style-type: none"> -Estudiar, analizar y entender las posibilidades que ofrece el SDK . -Desarrollar códigos de prueba para las operaciones del SDK. -Determinar los distintos artefactos de conocimiento gestionados por la fuente. 	<ul style="list-style-type: none"> -Analizar los Mapeable Elements existentes, con el propósito de validar que existen todos los Mapeable Elements necesarios para gestionar la tipología de información gestionada en la fuente. -Desarrollar o modificar Mapeable Elements para la gestión de información fuente. -Desarrollar componente parsing que realizará las invocaciones a la herramienta fuente desde las 	<ul style="list-style-type: none"> -Implementar la interface ISource2SRL, con las funciones definidas de acuerdo al alcance del conector SRL, con sus respectivos test unitarios. 	<ul style="list-style-type: none"> -Crear una instancia dentro del consumidor del conector SRL, con su respectiva opción que permita a los usuarios invocar el conector contra una o varias fuentes de información.
Salidas	<ul style="list-style-type: none"> -Mecanismos de interoperabilidad seleccionados y sustentados. -Informe incluidos códigos de prueba con las funcionalidades necesarias para hacer la gestión del conocimiento en la fuente utilizando el SDK. -Informe con la tipología de información gestionada por la fuente. 	<ul style="list-style-type: none"> -Mapeable Elements para el soporte de la información gestionada por la fuente. -Componente parsing con sus respectivos test unitarios. 	<ul style="list-style-type: none"> -Conector SRL basado en la interface ISource2SRL. 	<ul style="list-style-type: none"> -Consumidor con opción disponible para instanciar el conector SRL desarrollado.

3.5. Conclusiones sobre el marco de interoperabilidad para procesos de ingeniería

En la presente sección se detallan conclusiones identificadas referentes al marco metodológico propuesto:

- Considerando el paradigma en el que se basa el marco propuesto, en donde se presenta SRL como un entorno neutral que actúa como una solución intermedia, promete ser una alternativa potente no solo para gestionar distintos artefactos de conocimientos, sino también la facilidad de representación, junto con la reducción en el esfuerzo de agregar más fuentes de información a los entornos de interoperabilidad (ver sección 3.2).
- El poder contar con un componente de interpretación de modelos SRL (ver sección 3.3.4), permite estandarizar las diferentes formalizaciones que se obtienen de las fuentes de información, esto constituye una ventaja muy importante, ya que se garantiza el entendimiento entre las diversas fuentes de información que gestionen los mismos tipos de información(por ejemplo máquinas de estado, requisitos, etc.), y de esta forma se sobrepasan conocidos problemas de pérdida de fidelidad en el conocimiento en distintas herramientas, como ser el caso de UML [186].
- La simplicidad del modelo SRL constituye una ventaja al momento de realizar representaciones de información, la sintaxis compleja, implica inversiones de esfuerzo altas para poder dominar las formas adecuadas de representar la información, considerando que la interoperabilidad semántica implica un entendimiento del significado del conocimiento, entender las sintaxis utilizada por las distintas fuentes de información es una labor compleja, el contar un un lenguaje de representación sencillo como SRL a nivel de entorno neutral facilita la formalización del conocimiento.

4 Implementación de métodos para la interoperabilidad en procesos de ingeniería

“La habilidad de exponer una idea es tan importante como la idea en sí misma”

Aristóteles

En este capítulo, se presenta una arquitectura tecnológica basada en el marco de interoperabilidad propuesto en el Capítulo 3, se desglosan cada uno de los componentes que la conforman. También, se clasifican los conectores de acuerdo a sus posibilidades tecnológicas y a las capacidades de interoperabilidad que ofrecen las fuentes de información. Asimismo, se describen casos de uso industriales realizados a la fecha, junto con un listado de herramientas de uso comercial en las que se han implementado el marco propuesto en esta tesis. Finalmente se presenta un listado de fabricantes que han solicitado la integración de sus herramientas a las tecnologías de TRC utilizando el marco propuesto.

4.1. Arquitectura tecnológica del marco propuesto

La Figura 4-1 muestra la arquitectura tecnológica bajo la cual se implementó el marco de interoperabilidad propuesto, las tecnologías seleccionadas pueden cambiar y adaptarse a la arquitectura conceptual descrita en la sección 3.3. Algunos componentes de esta

4.1 Arquitectura tecnológica del marco propuesto

arquitectura han sido desarrollados como parte del trabajo presentado en esta tesis, otras eran tecnologías existentes y han sido ajustadas, otros componentes han sido reutilizados. Esta solución se llevó a cabo con el propósito de validar la solución conceptual propuesta en esta tesis, las tecnologías seleccionadas para proveer los servicios de reutilización, trazabilidad, transformación, calidad y verificación y validación son tecnologías robustas que actualmente son la base de herramientas que ofrecen a sus clientes apoyo en los procesos de ingeniería de sistemas [187].

Figura 4-1: Arquitectura tecnológica del marco propuesto.

En la Figura 4-1 se define una colorimetría indicando que componentes corresponden a nuevos desarrollos, a modificaciones o al consumo de componentes ya desarrollados.

4.1.1. API de fuentes de información

Este componente representa las tecnologías de interoperabilidad que los fabricantes ponen a disposición de los usuarios para interoperar con sus herramientas. Los fabricantes suelen proporcionar tecnologías que facilitan el acceso y la interacción con las mismas,

esto da valor agregado ya que facilita la automatización de procesos, así como la extensión de funcionalidades, en muchos casos se limitan a la gestión del contenido, en otros casos, la tecnología de interoperabilidad permite la ejecución de operaciones propias de cada herramienta.

4.1.2. Conector SRL

Consiste en un componente de software instanciable, desarrollado en C#, que ofrece interfaces, clases, extensiones y otros utilitarios que sirven como base para el desarrollo de conectores de forma estandarizada. Al implementar una interface común, los conectores son compatibles entre sí y se puede desarrollar un gestor de interoperabilidad que pueda integrar los conectores de forma automática, lo que significaría que minimizar los esfuerzos necesarios al momento de integrar nuevos conectores a una plataforma de gestión de interoperabilidad. La Figura 4-2 muestra un ejemplo de los componentes que conforman un conector SRL a Rational Rhapsody, se puede ver como este conector implementa la interface “ISource” que estandariza las operaciones de gestión del contenido, invocación de operaciones y una GUI de configuración.

Figura 4-2: Conector SRL a Rhapsody.

4.1 Arquitectura tecnológica del marco propuesto

En el ejemplo anterior esta gestión de contenido se produce de la implementación de funciones de la interface de *ISource*, la implementación de estas funciones que son comunes en todos los conectores desarrollados bajo *ISource*, permitirán las altas, bajas, modificaciones y consultas, estas implementaciones se llevan a cabo sobre las tecnologías que proveen los fabricantes para interoperar con las distintas herramientas, en el caso de la Figura 4-2, la implementación de estas funciones se hace con llamados a el Api de Rational Rhapsody, es decir que para invocar un alta de un bloque en Rhapsody, se invocará la funcionalidad dentro del Api que permite la ejecución de dicha operación, la información necesaria para realizar este proceso será proporcionada mediante elementos SRL. Lo anterior implica que si *ISource* ofrece una función llamada *InsertWorkproduct()* para crear nuevos contenidos, esta estará presente en todos los conectores que implementen el *ISource*.

4.1.3. SRL2Cake

Se desarrolló utilizando C# y permite la transformación de los grafos semánticos en SRL a elementos *Computer Aided Knowledge Environment (CAKE)*, este componente se desarrolló principalmente con el propósito de integrar el conocimiento representado en SRL, con la tecnología que ofrece el API de CAKE, la Tabla 4-1 detalla la equivalencia entre los componentes de SRL y CAKE.

Tabla 4-1: Equivalencia entre elementos SRL y elementos CAKE.

SRL	CAKE
<i>Artifact</i>	<i>Artifact</i>
<i>Artifact Occurrence</i>	KE
<i>Property</i>	<i>Meta</i>
<i>MetaData</i>	<i>Meta</i>
<i>Term</i>	Term
<i>Relationship</i>	RSHP
<i>Type</i>	<i>Artifact Type</i>
<i>Type</i>	SemanticCluster
<i>Type</i>	TermTag

Este componente ademas de facilitar la conversión de conocimiento representado en

SRL a CAKE, permite realizar la operación inversa, lo cual constituye el pilar de envío de información proveniente de las herramientas construidas alrededor de CAKE [187] a los conectores desarrollados en SRL.

4.1.4. SRL2RDF

Permite hacer serializaciones de la información representada en SRL a tripletas RDF, adicionalmente es posible generar archivos OWL a partir de SRL. Para el desarrollo de SRL2RDF se utilizó OWL API de Java [188], la cual fue transformada a un archivo .dll utilizando IKVM [189] como se muestra en la Figura 4-3. Se desarrolló una biblioteca de clases en Visual Studio .Net 2019 framework 4.8, que hace uso del Owl API en formato .dll, y que hace referencia al API SRL en su versión .Net, dentro de esta solución se establecieron reglas de transformación con el propósito de serializar información de SRL a RDF/OWL y viceversa.

Figura 4-3: Arquitectura de SRL2RDF.

4.1 Arquitectura tecnológica del marco propuesto

En la Tabla 4-2 se presentan las equivalencias entre los objetos de OWL con respecto a los objetos SRL. También OWL permite obtener información como ser entre otras cosas, sinónimos, relaciones inversas, en estos casos al momento de realizar la transformación se crearon relaciones en SRL de tipo Sinónimo e Inverso respectivamente entre estos elementos.

Tabla 4-2: Equivalencia entre componentes OWL y SRL.

SRL	OWL
<i>Property</i>	<i>Data property</i>
<i>Artifact Type</i>	<i>Type</i>
<i>Relationships</i>	<i>ObjectProperty</i>
<i>Artifact Type</i>	<i>Class</i>
<i>Artifact</i>	<i>Instance</i>

4.1.5. CAKE API

Como se muestra en la Figura 4-4, el ecosistema de herramientas TRC descansa sobre CAKE API. que es un componente basado en el lenguaje de representación RSHP propuesto por [11]. el Api está desarrollada en .Net y es de uso comercial, dentro del Api es posible gestionar un repositorio de conocimiento bajo las plataformas Oracle, SQL Server, MySQL y Access. El Api cuenta con un motor de retrieval que es configurable y que facilita la recuperación de información basados en la semántica y la topología de los elementos representados en la estructura de clases del CAKE Api. Asimismo, los elementos gestionados por CAKE Api, ofrecen posibilidades de traza, gestión de la configuración así como transformación de conocimiento, estas funcionalidades son potenciadas y ofrecidas a los usuarios por las diferentes herramientas de TRC [187]. Para el caso de los conectores SRL, al contar con una herramienta de transformación de los objetos de SRL a los objetos de CAKE (ver la sección 4.1.3) y viceversa, es posible utilizar todo el conocimiento representado en SRL con las funcionalidades que ofrece el Api de CAKE. Considerando el contexto en el que se realizó esta tesis, al contar con un acuerdo entre UC3M y The Reuse Company, se contó con el licenciamiento necesario para poder hacer uso de todos los servicios ofrecidos por el CAKE Api.

Figura 4-4: Ecosistema de herramientas de TRC.

4.2. Tipos de conectores SRL en función de sus posibilidades tecnológicas

Los conectores SRL, conceptualmente son componentes desarrollados para interoperar con una fuente de información, la cual, más concretamente puede ser una herramienta, por ejemplo Rational Rhapsody, esta fuente puede agrupar muchas tipologías de información, como ser modelos SysML, modelos UML, etc. Los conectores están concebidos para ser componentes instanciables, es decir si yo cuento con un conector SRL a Rational Rhapsody yo puedo instanciarlo tantas veces como sea necesario a distintos archivos contra los cuales necesito interoperar. Estos conectores se han dividido en dos posibilidades de acuerdo a la madurez de la tecnología ofrecida por los fabricantes para interoperar:

1. Conectores basados en APIs: Permiten la gestión total de la información fuente, es

4.2 Tipos de conectores SRL en función de sus posibilidades tecnológicas

decir altas, bajas, modificaciones y consultas, incluso en algunos casos, la invocación de funcionalidades especializadas, provistas por los fabricantes, ej: Rhapsody Model Check, Simulink Simulations.

2. Conectores basados en lectura de archivos: Son conectores de solo lectura, y consiste en la interpretación del contenido en archivos.

4.2.1. Conectores basados en APIs

El desarrollo de estos conectores se lleva a cabo sobre las APIs proporcionadas por los fabricantes para interactuar con las herramientas, dependiendo de las posibilidades que ofrezcan las APIs, estos conectores estandarizan la forma en que las operaciones se ejecutan en las distintas herramientas. Los grandes bloques de funcionalidad que se intentan desarrollar basándose en las APIs son los siguientes:

- Gestión de la información: La posibilidad de realizar operaciones de creación, modificación, eliminación y recuperación de elementos sobre la herramienta que se está integrando.
- Gestión de la configuración: Si la herramienta sobre la que se está realizando la integración permite la gestión de la configuración y las APIs permiten invocar esta funcionalidad, se integran dentro de los conectores SRL la posibilidad de realizar peticiones a los procesos de gestión de la configuración.
- Trazabilidad: Al igual que la gestión de la configuración, en caso de que la herramienta que se esté integrando cuente con la posibilidad de gestionar trazabilidad, los conectores SRL tendrán la capacidad de invocar las operaciones.
- Operaciones: Este gran bloque de funcionalidades está pensado para aprovechar funcionalidades propias de las herramientas integradas mediante un conector SRL. Por ejemplo la posibilidad de invocar la función de Check Model de Rhapsody, de forma que pueda ser invocada desde el conector SRL con información proporcionada por otro conector SRL, la cual puede provenir de otras herramientas, por ejemplo: Se quiere aplicar la funcionalidad de Check Model de Rhapsody a un modelo SysML que existe en un modelo de MagicDraw si se cuenta con un conector

SRL de Rhapsody y otro para MagicDraw.

- **Búsqueda:** Utilizar los motores de búsqueda y recuperación de información que estén disponible en las herramientas y que puedan ser invocados mediante las APIs. La Figura 4-5 muestra la arquitectura de los conectores basados en las APIs nativas, donde se tiene un componente interno que centraliza las peticiones de los grandes bloques de funcionalidad definidos en los conectores SRL, este bloque interno es el componente que ejecuta las peticiones mediante las APIs y es el que recibe la información proporcionada por las herramientas, internamente el conector recibe la información y la transforma a modelos SRL.

Figura 4-5: Arquitectura de conector SRL nativo.

4.2.2. Conectores basados en lectura de archivos

Este tipo de implementación se da cuando una herramienta no proporciona la tecnología necesaria para interoperar, en estos casos se desarrolla un componente capaz de interpretar los archivos fuentes que generan las herramientas. Estos conectores son solamente de lectura, ya que no se permite la creación de nuevos contenidos dentro de los

4.2 Tipos de conectores SRL en función de sus posibilidades tecnológicas

archivos. Como se puede ver en la Figura 4-6 la arquitectura de estos conectores consiste en un módulo que gestiona las peticiones de consulta de información, este módulo es desarrollado Ad-OC para entender la semántica de archivos generados por herramientas en concreto, independientemente del formato, que en la mayoría de los casos suele ser XML.

Figura 4-6: Arquitectura de conector SRL basado en la lectura de archivos.

4.2.3. Conexiones SRL

Las conexiones SRL son instancias de un conector SRL bajo configuraciones específicas, en ellas se define la fuente contra la que se realizarán los procesos de interoperabilidad, así como otras configuraciones que se requieran para definir comportamientos puntuales. Estas conexiones pueden crearse bajo demanda, como se muestra en la Figura 4-7 el conector se instancia de acuerdo a los archivos contra los que se quiere realizar operaciones de interoperabilidad.

Las conexiones permiten definir configuraciones de acuerdo a como se desarrolle cada conector, por ejemplo en el caso de un archivo de Simulink, una configuración podría

Figura 4-7: Ejemplo de conexiones a Simulink.

ser interoperar únicamente con los componentes de tipo subsistema, esta configuración implicaría que el conector únicamente trabajaría con bloques de tipo subsistema.

4.2.4. Ejemplo de integración de metodología de interoperabilidad con herramientas TRC

En la Figura 4-8, se presenta la arquitectura en la que se integran herramientas de la empresa The Reuse Company [187], con un módulo de interoperabilidad que está conformado por múltiples conectores SRL. El módulo de interoperabilidad permite la gestión del conocimiento desde las herramientas de TRC hacia los conectores y viceversa, también la realización de operaciones transversales es decir extraer información de una fuente y enviarla a otra, es decir es posible leer una máquina de estados en Simulink y escribirla en Rhapsody mediante la invocación de los conectores de SRL de Rhapsody y Simulink desde las herramientas de TRC.

Adicionalmente es posible utilizar la información proporcionada por los conectores SRL para crear una ontología usando KM, en donde se gestiona la estructura de componentes, vocabulario controlado y terminología entre otros. Asimismo, toda la información

4.2 Tipos de conectores SRL en función de sus posibilidades tecnológicas

proporcionada por los conectores SRL podrá ser utilizada en los diferentes procesos de gestión del conocimiento proporcionados por las herramientas de TRC, como ser trazabilidad, verificación, validación, transformaciones, etc.

Figura 4-8: Integración SRL con herramientas TRC.

4.3. Descripción de casos de uso de implementación del marco de interoperabilidad para procesos de ingeniería en herramientas

En esta sección se presenta un detalle de fabricantes que han solicitado la integración de sus herramientas bajo el marco propuesto. También los casos de uso de las distintas implementaciones del marco de interoperabilidad propuesto en herramientas de TRC.

4.3.1. Detalle de fabricantes que demandan integración de sus herramientas bajo el marco propuesto

Producto de los conectores desarrollados durante la actividad doctoral, los cuales como se ha especificado en las secciones anteriores están disponibles ya en herramientas de TRC, fabricantes de herramientas han solicitado la integración de sus herramientas en los entornos de TRC, estas integraciones bajo el marco propuesto descrita en este capítulo. En la Tabla 4-3 se presenta un detalle con los fabricantes que han realizado dicha solicitud.

Tabla 4-3: Listado de fabricantes que demandan integración de sus herramientas al marco propuesto en el ecosistema de TRC.

No	Herramienta	Tipo de información
1	Siemens Teamcenter	Requisitos, BOM, diseño
2	Jama	Requisitos, casos de prueba, riesgos
3	Vitech Genesys	Requisitos y modelos lógicos
4	Polarion	Requisitos, casos de prueba y elementos personalizados
5	Altium	Modelos físicos

Del listado presentado en la Tabla 4-3, actualmente se están en proceso de desarrollo los siguientes conectores:

- Siemens Teamcenter: El desarrollo de este conector se está llevando a cabo en conjunto con el equipo de desarrollo de Siemens, abarca la gestión de requisitos, BOM y otros elementos de diseño.

4.3 Descripción de casos de uso de implementación del marco de interoperabilidad para procesos de ingeniería en herramientas

- Polarion: Se están finalizando las actividades de desarrollo, producto de la solicitud realizada por personal de Siemens a TRC. Implica la gestión de requisitos, casos de prueba y elementos personalizados.

El resto de conectores, a la fecha de escritura de esta tesis están siendo planificados, para ser integrados a las herramientas de TRC bajo el marco propuesto en esta tesis.

Detalle de conectores SRL integrados en herramientas TRC

La Tabla 4-4 presenta el detalle de conectores SRL integrados en las herramientas KM, RQA y Traceability Studio.

Tabla 4-4: Detalle de conectores SRL actualmente implementados KM, RQA y Traceability Studio.

No	Conector SRL	Descripción	Herramientas
1	Conector SRL a MagicDraw	Permite obtener modelos lógicos SysML en MagicDraw	KM, RQA, Traceability Studio
2	Conector SRL a Rhapsody	Permite obtener modelos lógicos SysML en Rational Rhapsody	KM, RQA, Traceability Studio
3	Conector SRL a Papyrus	Permite obtener modelos lógicos SysML en Papyrus	KM, RQA, Traceability Studio
4	Conector SRL a Simulink	Conector SRL que permite obtener modelos físicos en Simulink	KM, RQA, Traceability Studio
5	Conector SRL a Capella	Permite la obtención de modelos Capella	KM, RQA, Traceability Studio
6	Conector SRL a ASCE	Obtención de diagramas de Safety Cases	KM, RQA, Traceability Studio
7	Conector SRL a Excel	Gestión de información tabular	KM, RQA, Traceability Studio
8	Conector SRL a Word	Obtención de párrafos y/o frases.	RQA, Traceability Studio
9	Conector SRL a Pure Variants	Obtención de líneas de producto	KM, RQA, Traceability Studio

4.3 Descripción de casos de uso de implementación del marco de interoperabilidad para procesos de ingeniería en herramientas

4.3.2. Caso de uso industrial del marco de interoperabilidad para procesos de ingeniería en la herramienta Knowledge Manager

Knowledge Manager (KM) es una herramienta de uso comercial permite la gestión de una ontología. como se muestra en la Figura 4-9, dentro del ámbito de KM, una ontología consiste en la definición de vocabulario de dominio incluidas agrupaciones semánticas, etiquetas sintácticas y otras propiedades. Modelos Conceptuales de Sistemas (SCM), esto permite la definición de arquitectura de sistemas mediante vistas y modelos, permitiendo la creación de relaciones entre sistemas, elementos y otras entidades, de esta forma clasificar información por significado y naturaleza entre otros [190]. Adicionalmente las ontologías en KM permiten la definición de patrones textuales para diversos propósitos. KM permite almacenar modelos SRL obtenidos ya sea por la aplicación de patrones textuales o como resultado de interoperar con otras diferentes fuentes de información, considerando que KM permite la búsqueda semántica de artefactos. Finalmente, KM permite la definición de reglas y acciones utilizando el conocimiento almacenado de de su repositorio de conocimientos [190].

Figura 4-9: Estructura de una ontología en el ámbito de KM [190].

Necesidad de interoperabilidad

KM ofrece la posibilidad de representar arquitectura de sistemas, mediante vistas y modelos, estos pueden ser definidos dentro de la herramienta de forma manual, sin embargo muchas organizaciones utilizan lenguajes de modelado lógico como SysML para representar arquitecturas, y este conocimiento ya existe en formato de alguna herramienta basada en SysML como ser como ser Papyrus, Magic Draw o Rhational Rhapsody. En vista de lo anterior, como alternativa para hacer uso este conocimiento que en muchos casos ya existe y evita los procesos de creación del mismo en KM la necesidad de interoperabilidad consistía en una conexión de lectura en tiempo real que extrajera de diferentes herramientas de modelado lógico los conceptos, arquitectura y modelos.

Como se muestra en la Figura 4-10 la integración de los conectores SRL dentro de KM, se realizó mediante un componente denominado “Knowledge Interface”, el cual permite realizar a demanda la invocación de operaciones que ofrecen los conectores SRL. esto permite la utilización del conocimiento de fuentes externas en los procesos de gestión de conocimientos de KM.

por otro lado, la obtención de modelos mediante los conectores SRL, permitió indexar modelos lógicos, físicos y otros artefactos de ingeniería gestionados en diversas fuentes de información para las cuales se cuenta con conectores SRL desarrollados, esta integración posibilitó la ejecución del experimento detallado en la sección 5.4, ya que a través de dichos conectores es posible realizar operaciones de reúso de artefactos, es decir indexar modelos y utilizar el motor de recuperación de KM.

4.3.3. Caso de uso industrial de implementación del marco de interoperabilidad para procesos de ingeniería en la herramienta Traceability Studio

Es una herramienta de software que permite la definición de enlaces de rastreo entre dos fuentes de información. Traceability Studio permite la posibilidad de rastrear enlaces entre procesos clave, como V&V, definición de requisitos, definición de arquitectura, de-

4.3 Descripción de casos de uso de implementación del marco de interoperabilidad para procesos de ingeniería en herramientas

Figura 4-10: Arquitectura de integración de conectores SRL en KM.

finición de diseño o gestión de riesgos, entre otros, como se define en la norma ISO IEEE 15288/12207 [191,192]. Traceability Studio proporciona un marco para la gestión trazas entre todo tipo de información, que cubre pares de artefactos de ingeniería en formato electrónico (accesibles). [192].

Necesidad de interoperabilidad

Considerando que la función principal de Traceability Studio es la de establecer enlaces de traza en artefactos de ingeniería y en vista de que la práctica actual de Ingeniería de Sistemas implica casi siempre la gestión de ecosistemas de varias herramientas de software, el proceso de definición de trazas estará condicionado por la conectividad que se tenga a distintas fuentes de conocimiento que generan los artefactos sobre los cuales se requiere establecer los enlaces. En ese sentido la necesidad de interoperabilidad del Traceability Studio consiste conectarse a múltiples fuentes de conocimientos para obtener los artefactos con el propósito de ofrecer a sus clientes la posibilidad de establecer enlaces entre elementos obtenidos.

La Figura 4-11 muestra como se integraron múltiples conectores dentro del módulo de conexión de Traceability Studio, de forma que los conectores permitan la definición de conexiones (ver sección 4.2.3) a fuentes de información en distintos formatos, de esta forma obtener los artefactos sobre los cuales se pueden realizar las operaciones de gestión de trazas que ofrece el Traceability Studio. Esta integración permitió la ejecución del experimento detallado en la sección 5.5, en donde se compararon dos algoritmos de sugerencia de trazas sobre distintas fuentes de información.

4.3.4. Caso de uso industrial del marco de interoperabilidad para procesos de ingeniería en la herramienta RQA - QUALITY Studio

Es una herramienta que permite automatizar la inspección y el análisis de calidad de diferentes artefactos de ingeniería. Considerando que los defectos pueden ser causados por decisiones de ingeniería inadecuadas o por la representación incorrecta de información de ingeniería en requisitos, modelos, etc. La automatización de las actividades de inspección de calidad permite a los ingenieros más tiempo para tomar mejores decisiones al tiempo que proporcionan los medios para que ningún defecto quede oculto. RQA posibilita el análisis de los elementos de ingeniería utilizando reglas, listas de verificación,

4.3 Descripción de casos de uso de implementación del marco de interoperabilidad para procesos de ingeniería en herramientas

Figura 4-11: Arquitectura de integración de conectores SRL en Traceability Studio.

políticas, etc., identificando defectos, inconsistencias e información incompleta [193].

Necesidad de interoperabilidad

Como se ha mencionado anteriormente, RQA permite la inspección de la calidad en diferentes artefactos de ingeniería, la necesidad de interoperabilidad radica puntualmente en la obtención de los artefactos de ingeniería de distintas fuentes de información, con el propósito de tener la posibilidad inspeccionar los distintos tipos de artefactos que interactúan en las diferentes fases de los procesos de ingeniería.

De forma similar a la implementación que se realizó en el Traceability Studio. La Figura 4-12 muestra como se integraron múltiples conectores dentro del módulo de conexión RQA, de forma que los conectores permitan la definición de conexiones (ver sección 4.2.3)

a fuentes de información en distintos formatos, de esta forma obtener los artefactos sobre los cuales se pueden realizar los procesos de inspección que permite RQA.

Figura 4-12: Arquitectura de integración de conectores SRL en RQA.

4.4. Conclusiones sobre implementación de métodos para la interoperabilidad en procesos de ingeniería

Producto de la experiencia adquirida durante la implementación de diversos conectores, se presentan las siguientes conclusiones:

- Durante el desarrollo de los conectores presentados en la sección 4.3.1, en donde se tuvo la oportunidad de interactuar con distintas herramientas que son utilizadas para la generación de diversos artefactos de conocimiento, se pudo experimentar

4.4 Conclusiones sobre implementación de métodos para la interoperabilidad en procesos de ingeniería

que el coste más alto de lograr interoperabilidad lo constituye el estudio de los diferentes modelos de objetos gestionados por las herramientas, esta actividad es fundamental al momento de lograr una interoperabilidad semántica. Las herramientas por lo general utilizan sintaxis que no son complejas de procesar, por ejemplo archivos XML, JSON, APIs REST, etc. Pero entender el modelo de objetos representado en estos formatos puede resultar sumamente complejo, incluso en herramientas que gestionan el mismo tipo de conocimiento, SysML, UML, etc.

- La arquitectura tecnológica del marco propuesto, proporciona una alternativa prometedora como alternativa tecnológica para resolver problemas de interoperabilidad y gestión del conocimiento. Permite explotar e integrar las funcionalidades que ofrecen los fabricantes de las herramientas. producto de lo anterior, en la sección 4.3.1 se presenta el detalle de fabricantes que han solicitado la integración de sus herramientas al ecosistema de TRC.

5 Experimentación: evaluación y análisis de resultados

“Habrá hoy, habrá mañana, siempre
habrá, y hubo ayer, y hay un día
antes...”

León Tolstói

En este capítulo se presenta la evaluación de la metodología propuesta, se explica detalladamente, la forma en que se lleva a cabo la validación de las hipótesis definidas en el Capítulo 1. La validación se lleva a cabo en dos partes, 1) Validación científica, la cual consta de tres (3) experimentos que corresponden a cada una de las tres (3) hipótesis definidas para esta tesis doctoral. También, se presentan consideraciones generales que facilitarán la comprensión de la evaluación, en donde se narran la estructura que sigue cada experimento y algunos criterios de evaluación. 2.) Validación industrial, se lleva a cabo mediante el desarrollo e integración de pilotos en casos de uso reales.

Los experimentos de la validación científica presentados en este capítulo son los siguientes:

- Experimento sobre evaluación de nivel de madurez (ver sección 5.3): Se pretende comprobar si existe mejora en los niveles de interoperabilidad comparando un escenario de interconexión entre diferentes herramientas sin utilizar la solución propuesta y replicando el mismo escenario utilizando la solución propuesta. Para determinar el nivel de interoperabilidad en los dos casos, se construyó una matriz basada en LISI [12].

5.1 Descripción de la validación

- Experimento para la reutilización de modelos lógicos y físicos (ver sección 5.4): Se compararon las capacidades de recuperación ofrecidas por las herramientas Rhapsody, Papyrus y Simulink, contra la solución propuesta, utilizando las métricas de Precisión, Recall y F1.
- Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas (ver sección 5.5): Se realiza una comparación entre dos algoritmos de descubrimiento de trazas, uno textual y el otro utilizando la solución propuesta. Dicha comparación se realiza utilizando las métricas de Precisión, Recall y F-Score (F_1).

5.1. Descripción de la validación

En la Tabla 5-1, se presentan las hipótesis de esta tesis, con una descripción que detalla el procedimiento utilizado para la realización de la validación de cada una de ellas.

5.2. Consideraciones generales

La validación se realizó utilizando la metodología propuesta por [194], en la que se definen las siguientes cuatro fases para la ejecución de los experimentos de validación:

1. Definición de objetivos: En esta fase se definirán las variables que serán examinadas.
2. Diseño: Se definirá un plan sobre el cual el experimento será llevado a cabo, esto implica determinar que variables pueden afectar el experimento.
3. Ejecución: Los experimentos más elementales son llevados a cabo en función a lo que se definió en la fase de diseño.
4. Análisis de resultados: En esta etapa se analiza la información recopilada de la ejecución del experimento.

Para efectos de validar niveles de Precisión y Recall, se utilizaron las medidas propuestas por [4]:

Tabla 5-1: Detalle de validación por hipótesis.

No	Hipótesis	Descripción
1	Hipótesis 1 (H1)	<i>Mediante una representación unificada de conocimiento es posible mejorar la interoperabilidad.</i> Con el propósito de corroborar esta hipótesis, se realiza un cálculo de nivel de interoperabilidad denominado IL. Se define un caso de estudio en donde se compara el nivel de interoperabilidad sin utilizar la solución propuesta (IL_x), contra un nivel de interoperabilidad utilizando la solución propuesta (IL_y). Una vez obtenidos los niveles de interoperabilidad se corrobora la Hipótesis 1. Para más detalle ver sección 5.3.
2	Hipótesis 2 (H2)	<i>Mediante la posibilidad de tener un mecanismo de interoperabilidad se puede mejorar la selección de artefactos.</i> Para corroborar esta hipótesis se construye un caso de estudio en el que se utiliza la solución propuesta para indexar y recuperar modelos lógicos y físicos. Se construyen un listado de consultas para modelos lógicos ($LM=Q_1, Q_2..Q_n$) y otro grupo de consultas para modelos físicos ($PM=Q_{H1}, Q_2..Q_n$), estas consultas se ejecutan utilizando la solución propuesta y también utilizando las capacidades de búsqueda ofrecidas por las herramientas Simulink, Papyrus y Rhapsody. Se calculan las métricas de Precisión, Recall y F_1 para cada herramienta y se realiza la comparación. Para más detalle ver sección 5.4.
3	Hipótesis 3 (H3)	<i>Mediante la posibilidad de tener un mecanismo de interoperabilidad se puede mejorar el descubrimiento de trazas.</i> Se corrobora la hipótesis, mediante un caso de estudio en donde se comparan dos enfoques de descubrimiento de trazas, uno basado en comparaciones textuales y otro basado en comparaciones semánticas utilizando la solución propuesta, se comparan utilizando las métricas de Precisión, Recall y F_1 . Para más detalle ver sección 5.5.

Tabla 5-2: Tabla de niveles de evaluación de métricas de precisión y Recall [4].

Niveles			
Medida	Aceptable	Bueno	Excelente
Recall	>60 %	>70 %	>80 %
Precision	>20 %	>30 %	>50 %

5.3 Experimento sobre evaluación del grado de interoperabilidad en un entorno de ingeniería

A continuación se detalla la información necesaria para realizar el cálculo de Precisión, Recall y F_1 .

- TP: Elementos recuperados y relevantes.
- FP: Elementos recuperados pero irrelevantes.
- TN: Elementos no recuperados e irrelevantes.
- FN: Elementos no recuperados pero relevantes.

Tabla 5-3: Matriz utilizada definir los elementos necesarios para el cálculo de Precisión, Recall y F_1 .

	Recuperado	No recuperado
Irrelevante	FP	TN
Relevante	TP	FN

La fórmula para calcular la Precisión es la siguiente:

$$Precision = \frac{TP}{TP+FP}$$

La fórmula para calcular el Recall es la siguiente:

$$Recall = \frac{TP}{TP+FN}$$

La fórmula para calcular F_1 , que es una combinación de Precisión y Recall, es la siguiente:

$$F_1 = \frac{2 * P * R}{P + R}$$

5.3. Experimento sobre evaluación del grado de interoperabilidad en un entorno de ingeniería

Este experimento consiste en la definición de un escenario de interconexión entre las siguientes herramientas:

1. IBM Rhapsody.
2. IBM Doors.
3. KnowledgeManager.

4. Altium Designer.
5. Traceability Studio.
6. Excel.
7. Word.

En base al listado anterior, se realizó el cálculo de un nivel de interoperabilidad denominado Interoperability Level (IL), éste, se calcula construyendo una matriz que especifica un valor numérico de acuerdo al nivel de interoperabilidad que tiene. En la Tabla 5-4 se presenta la matriz que debe ser construida para determina IL:

Tabla 5-4: Matriz de nivel de sofisticación de interoperabilidad (IM).

	Herramienta ₁	Herramienta ₂	Herramienta _j
Herramienta ₁	LV _{1,1}	LV _{1,2}	LV _{1,j}
Herramienta ₂	LV _{2,1}	LV _{2,2}	LV _{2,j}
Herramienta _i	LV _{i,1}	LV _{i,2}	LV _{i,j}

En donde IL es:

$$IL = \sum_{i=1}^n \sum_{j=1}^m LV_{i,j}$$

y LV se define en base a los criterios presentados en la Tabla 5-5:

5.3.1. Objetivo

El objetivo definido para la validación de este experimento es comparar el nivel de interoperabilidad en el mismo escenario, sin utilizar la solución propuesta y utilizando la solución propuesta.

5.3.2. Diseño

Como se ha especificado en el Capítulo 1, la actividad empresarial implica la interacción entre múltiples fuentes de información [5,42]. Esto se traduce en necesidades de interconexión entre diversas fuentes de conocimiento, en algunos casos los fabricantes de las herramientas facilitan el traslado de información de una fuente a otra, mediante

5.3 Experimento sobre evaluación del grado de interoperabilidad en un entorno de ingeniería

Tabla 5-5: Criterios para definición del valor numérico de LV.

Nivel	Descripción	Valor
Nivel 0	Sin intercambio. No existe intercambio entre fuentes.	0
Nivel 1	Comunicación a nivel de protocolo. Se utiliza el mismo protocolo de comunicación, p. Ej. HTTP.	1
Nivel 2	Comunicación a nivel de sintaxis. Se utiliza el mismo formato de datos, p. Ej. JSON.	2
Nivel 3	Comunicación a nivel semántico. Es posible un entendimiento a nivel semántico de la información.	3
Nivel 4	Conceptual. Es posible gestionar la misma terminología tanto en el origen como el destino.	4

funcionalidades de importación. Lo anterior facilita en cierto grado el traslado del conocimiento entre herramientas, por lo tanto, aun y cuando no se cuenten con sistemas de interconexión personalizados, en algunos casos existe cierto grado de interoperabilidad, por lo que este experimento se basa en la comparación entre el cálculo de IL sin utilizar la solución propuesta (IL_x) y el cálculo utilizando la solución propuesta (IL_y). El experimento se ha diseñado de la siguiente forma:

1. Se define un listado de herramientas sobre las que se realizará el experimento.
 - a) IBM Rhapsody.
 - b) IBM Doors.
 - c) KnowledgeManager.
 - d) Altium Designer.
 - e) Traceability Studio.
 - f) Excel.
 - g) Word.
2. Se construye la matriz de niveles de sofisticación de interoperabilidad basada en 5-4, sin considerar la solución propuesta, IM_x .

3. Se calcula el IL para la matriz construida en el paso anterior, IL_x .
4. Se construye la matriz de niveles de sofisticación de interoperabilidad basada en 5-4, considerando la solución propuesta, IM_y .
5. Se calcula el IL para la matriz construida en el paso anterior, IL_y .
6. Se comparan los resultados de IL_x e IL_y .

5.3.3. Ejecución del experimento

En la Tabla 5-6 (IM_x) se presenta el nivel de sofisticación alcanzado, considerando las capacidades de importación que por defecto ofrecen las herramientas, así como la compatibilidad existente entre las herramientas, por ejemplo, Excel y Word son herramientas muy compatibles, de Excel a Word y viceversa es posible incrustar información, adicionalmente Doors ofrece capacidades de importación de información desde Word y Excel, Knowledge Manager tiene la posibilidad de extraer información a nivel semántico. Una vez construida la matriz, se procedió al cálculo de IL_x , que haciendo la sumatoria total da como resultado 56. Por lo tanto podemos decir que $IL_x=56$. Con respecto IM_y , en la Tabla 5-7 se presenta la matriz de sofisticación considerando que todas las conexiones se llevan a cabo utilizando la solución propuesta, utilizando IM_y se realiza el cálculo de IL_y , en donde se determina que $IL_y=87$.

Nota: Para la construcción de IM_y solamente se consideraron los conectores SRL actualmente desarrollados (ver sección 4.3.1), a la fecha no se ha desarrollado conector a Altium Designer, por ende en el experimento se consideró que no existe conectividad.

5.3 Experimento sobre evaluación del grado de interoperabilidad en un entorno de ingeniería

Tabla 5-6: Matriz de sofisticación de interoperabilidad sin utilizar la solución propuesta IM_x .

	IBM Rhapsody	IBM Doors	Knowledge Manager	Altium Designer	Traceability Studio	Excel	Word
IBM Rhapsody	2	1	0	0	0	1	1
IBM Doors	1	2	0	0	0	1	1
Knowledge Manager	3	3	3	0	3	3	3
Altium Designer	0	0	0	0	0	0	0
Traceability Studio	3	3	3	0	3	3	3
Excel	0	1	0	0	0	2	2
Word	0	1	0	0	0	2	2

Tabla 5-7: Matriz de sofisticación de interoperabilidad utilizando la solución propuesta IM_y .

	IBM Rhapsody	IBM Doors	Knowledge Manager	Altium Designer	Traceability Studio	Excel	Word
IBM Rhapsody	3	3	0	0	0	3	3
IBM Doors	3	3	0	0	0	3	3
Knowledge Manager	3	3	3	0	3	3	3
Altium Designer	0	0	0	0	0	0	0
Traceability Studio	3	3	3	0	0	3	3
Excel	3	3	3	0	0	3	3
Word	3	3	3	0	0	3	3

5.3.4. Análisis de resultados del experimento

De acuerdo a los resultados de IL_x e IL_y , se puede concluir que **la solución propuesta ofrece mayores niveles de interoperabilidad**. Esto se debe principalmente a la formalización que se construye en la solución propuesta, dicha formalización ofrece una representación estándar gracias al Mappable Element (ver sección 3.3.4)así como también, provee semántica que puede ser utilizada por los consumidores de la información, para realizar diversas tareas como ser, reuso, trazabilidad, transformación, etc. A pesar de que existen formas de obtener información en las herramientas, como la importación de información, para poder lograr una integración que provea mayores niveles de sofisticación es necesario desarrollar conectores personalizados para cada conexión.

Adicionalmente, como se puede apreciar en la Tabla 5-6 solamente Knowledge Manager cuenta con posibilidades de interconexión a Nivel 3, de acuerdo a los criterios definidos en la Tabla 5-5. Esto debido a que las herramientas no ofrecen mecanismos de interoperabilidad a nivel semántico, es decir no existe un nivel de comprensión profundo entre fuentes de información, salvo el caso de KnowledgeManager, en donde todas las conexiones permiten este nivel de sofisticación, debido a que el mismo hace uso de grafos semánticos para gestionar el conocimiento.

5.3.5. Validación Hipótesis H1

Considerando los resultados obtenidos en el experimento, en donde IL_y es mayor que IL_x , lo que significa que la sumatoria total en los niveles de sofisticación de interoperabilidad entre herramientas es mayor utilizando la solución propuesta. Se concluye que se valida la **Hipótesis H1: Mediante una representación unificada de conocimiento es posible mejorar la interoperabilidad**. Se experimentan mejoras en el nivel de interoperabilidad general utilizando la solución propuesta.

5.4. Experimento sobre reutilización de modelos lógicos y físicos

Con el propósito de validar si la solución propuesta por esta tesis mejora las posibilidades de reutilización, se realizó el experimento descrito a continuación. Para llevar a cabo el experimento se desarrollaron tres conectores bajo la arquitectura propuesta en esta tesis. Un conector a Simulink que permite la lectura de archivos mdl y slx y su posterior representación en SRL [1]. Un segundo conector que permite la lectura y representación en SRL de los artefactos de modelos lógicos creados en la herramienta Papyrus y un tercer conector que realizara el mismo procedimiento para archivos de Rhapsody [1]. Estos tres conectores permiten la interpretación semántica de los artefactos de ingeniería de tres fuentes distintas, para los procesos de reuso se realizó la transformación de SRL a CAKE, con las tecnologías desarrolladas en esta tesis, una vez se contó con el conocimiento representado en RSHP, se utilizaron las Apis de CAKE que proveen servicios de reuso de artefactos.

Este experimento se publicó en [1].

5.4.1. Objetivo

Determinar la efectividad de la solución propuesta para encontrar modelos SysML y Simulink en un repositorio de modelos [1].

5.4.2. Diseño

En este caso, se han realizado dos experimentos diferentes para probar las capacidades de recuperación de:

1. Intercambio y selección de modelos lógicos en SysML. Papyrus e IBM Rhapsody se han seleccionado como herramientas de destino que pueden administrar modelos lógicos y se espera que proporcionen capacidades de reutilización.

2. Intercambio y selección de modelos físicos. En este caso, Simulink ha sido seleccionado como herramienta para crear y gestionar modelos de sistemas físicos.

En ambos casos, las capacidades de reuso de estas herramientas se comparan con el enfoque propuesto [1].

Para el experimento basado en modelos lógicos se realizaron los siguientes pasos [1]:

1. Definir un conjunto de modelos de prueba

$$D = \{M_1, M_2, \dots, M_k, \dots, M_m\}$$

, El conjunto para el experimento de modelos lógicos consta de 44 modelos SysML basados en diagramas de requisitos existentes [195, 196]. Los cuales se han desarrollado en Papyrus e IBM Rhapsody. En el caso de los modelos de simulink el conjunto consta de 26 modelos de Simulink obtenidos de los ejemplos disponibles en la instalación de Simulink. Para ambos casos los modelos han sido indexados en la herramienta knowledMANAGER, la cual se basa en las Apis de CAKE [1].

2. Definir un conjunto de consultas y resultados esperados, Q donde cada consulta q_1 devolverá un conjunto de modelos lógicos D_k , en el caso del primer experimento el conjunto de consultas consiste en 25 elementos (ver Tabla 5-10), para el segundo experimento el conjunto está conformado por 20 (ver Tabla 5-10) consultas [1].

$$Q = \{(q_1, D_1), (q_2, D_2), \dots, (q_k, D_k), \dots, (q_n, D_n)\}$$

3. Ejecutar los procesos de indexación y recuperación sobre conectores desarrollados para la lectura de archivos Payrus, IBM Rational Rhapsody [1].
4. Calcular métricas de Precisión, Recall y F1 (la media armónica de precisión y recall) haciendo una comparación de los resultados esperados y generados [1].

5.4.3. Ejecución del experimento

La Tabla 5-8, muestra las métricas de Precisión, Recall y F1 obtenidas de la ejecución del experimento, la primera columna corresponde al indicador de la consulta indexada,

5.4 Experimento sobre reutilización de modelos lógicos y físicos

las siguientes tres columnas proporcionan las métricas de Precisión, Recall y F1 obtenidas con la recuperación, utilizando la metodología propuesta en esta tesis, en el caso de las columnas 5 a la 7 corresponden a las métricas de Precisión, Recall y F1 obtenidas del uso de las plataformas de búsqueda que ofrece Papyrus, en el caso de las columnas de la 8 a la 10, corresponden a las métricas de Precisión, Rrecall y F1 utilizando el motor de búsquedas que ofrece IBM Rhapsody [1].

Tabla 5-8: Resultados de la recuperación de modelos lógicos con SRL, Papyrus e IBM Rhapsody [1].

Query	SRL			Papyrus			IBM Rhapsody		
	P	R	F1	P	R	F1	P	R	F1
Q₁	0.45	0.94	0.61	1.00	0.44	0.61	1.00	0.44	0.61
Q₂	1.00	0.83	0.91	1.00	0.67	0.80	1.00	0.67	0.80
Q₃	0.67	1.00	0.80	0.75	0.38	0.50	0.80	0.50	0.62
Q₄	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Q₅	1.00	0.75	0.86	0.00	0.00	0.00	0.00	0.00	0.00
Q₆	0.36	1.00	0.53	1.00	1.00	1.00	1.00	1.00	1.00
Q₇	1.00	1.00	1.00	1.00	0.56	0.71	1.00	0.56	0.71
Q₈	0.33	1.00	0.50	0.50	1.00	0.67	0.00	0.00	0.00
Q₉	0.38	1.00	0.55	0.75	1.00	0.86	0.75	1.00	0.86
Q₁₀	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Q₁₁	0.50	1.00	0.67	0.33	1.00	0.50	0.33	1.00	0.50
Q₁₂	0.50	1.00	0.67	0.12	1.00	0.22	0.12	1.00	0.22
Q₁₃	0.36	1.00	0.53	0.12	1.00	0.22	0.14	1.00	0.24
Q₁₄	1.00	0.73	0.84	0.73	1.00	0.85	0.71	0.91	0.80
Q₁₅	0.75	1.00	0.86	1.00	0.67	0.80	1.00	0.67	0.80
Q₁₆	1.00	1.00	1.00	1.00	0.75	0.86	1.00	1.00	1.00
Q₁₇	1.00	1.00	1.00	1.00	0.75	0.86	1.00	0.75	0.86

continúa en la siguiente página

Tabla 5-8 – Continuación de la página anterior.

Query	SRL			Papyrus			IBM Rhapsody		
	P	R	F1	P	R	F1	P	R	F1
Q₁₈	1.00	1.00	1.00	0.50	1.00	0.67	0.50	1.00	0.67
Q₁₉	0.70	1.00	0.82	0.88	1.00	0.93	0.88	1.00	0.93
Q₂₀	1.00	1.00	1.00	0.18	1.00	0.31	0.24	1.00	0.38
Q₂₁	1.00	1.00	1.00	0.12	1.00	0.22	1.00	1.00	1.00
Q₂₂	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Q₂₃	0.78	1.00	0.88	0.16	1.00	0.27	0.88	1.00	0.93
Q₂₄	1.00	1.00	1.00	0.50	1.00	0.67	0.50	1.00	0.67
Q₂₅	0.43	0.75	0.55	1.00	1.00	1.00	1.00	1.00	1.00
Avg	0.77	0.96	0.82	0.67	0.85	0.66	0.71	0.82	0.70

La Tabla 5-9, muestra las métricas de Precisión, Recall y F1 obtenidas de la ejecución del segundo experimento, la primera columna corresponde al indicador de la consulta indexada, las siguientes tres columnas proporcionan las métricas de Precisión, Recall y F1 obtenidas con la recuperación utilizando la metodología propuesta en esta tesis, en el caso de las columnas 5 a la 7 corresponden a las métricas de Precisión, Recall y F1 obtenidas del uso de las plataformas de búsqueda que ofrece Simulink.

Tabla 5-9: Resultados de la recuperación de modelos físicos con SRL y Simulink [1].

Query	SRL			Simulink		
	P	R	F1	P	R	F1
Q1	0.44	0.41	0.42	0.64	0.94	0.76
Q2	0.58	0.65	0.61	0.64	0.94	0.76
Q3	0.53	1.00	0.69	1.00	0.90	0.95
Q4	1.00	1.00	1.00	0.00	0.00	0.00
Q5	1.00	1.00	0.00	0.00	0.00	0.00
Q6	1.00	1.00	1.00	0.00	0.00	0.00
Q7	0.10	1.00	0.18	0.00	0.00	0.00
Q8	1.00	0.44	0.61	1.00	0.69	0.81
Q9	0.13	1.00	0.22	0.00	0.00	0.00
Q10	0.67	1.00	0.80	0.24	1.00	0.39
Q11	0.40	1.00	0.57	0.16	1.00	0.28
Q12	0.33	0.75	0.46	0.16	1.00	0.28
Q13	1.00	0.26	0.42	0.00	0.00	0.00
Q14	0.90	0.56	0.69	0.80	0.25	0.38
Q15	0.90	0.56	0.69	0.67	1.00	0.80
Q16	0.17	0.25	0.20	0.32	1.00	0.48
Q17	0.58	0.85	0.69	0.31	0.31	0.31
Q18	0.90	1.00	0.95	0.25	0.56	0.34
Q19	0.90	1.00	0.95	0.28	0.56	0.37
Q20	1.00	1.00	1.00	0.00	0.00	0.00
Avg	0.68	0.79	0.61	0.32	0.51	0.35

Tabla 5-10: Consultas para recuperación de modelos lógicos [1].

Id	Query string
Q1	System availability
Q2	Maximum rate of failure
Q3	Manage Traffic flow
Q4	System for purify water
Q5	System using remote control component
Q6	System use cameras
Q7	System with an statistical data component
Q8	System Performance Requirements
Q9	Requirements of System Usability
Q10	System with Simulation Component
Q11	Group Creation
Q12	System Restrictions Requirements
Q13	System that use Sensors
Q14	Gather and Interpret Information Module
Q15	Adaptive Control
Q16	Consistency in transaction
Q17	Manual Control
Q18	intruders detection
Q19	Time Validation
Q20	computer response time
Q21	System validation cards
Q22	tasks and scenarios
Q23	traffic management based in the region
Q24	semaphores automatic operation
Q25	Control standard

Tabla 5-11: Consultas para recuperación de modelos físicos [1].

Id	Query string
Q1	A flow between a constant, product, block sum and a output block.
Q2	A flow between an inport, product, an a block sum.
Q3	A flow between an inport, block sum and integrator.
Q4	A flow between a subsystem and output block.
Q5	A flow between a subsystem and to Workspace block.
Q6	A flow between a Transport Delay and Subsystem block.
Q7	A flow between a Integrator block, Transport Delay and Subsystem block.
Q8	A flow between a Inport and constant blocks with a product block.
Q9	A flow between a Inport and constant blocks with a product block and the product block with output block
Q10	A flow between a Integrator and Subsystem, Add block and subsystem and Subsystema with Subsystem
Q11	A flow between a Integrator and Subsystem, Add block and subsystem and Subsystema with Subsystem1 and subsystem2
Q12	A flow between a Integrator and Subsystem, Add block and subsystem and Subsystema with Subsystem1 and subsystem2 with to Workspace block
Q13	Model with no flows only inport block, output block and product block
Q14	Two submodels of A flow between an inport, product, an a block sum and output.
Q15	Two submodels of A flow between an inport, product, an a block sum and output with two constants
Q16	A flow between inport and add block, and two inports nodes without flow
Q17	A flow between add bloc and constant with divide block.
Q18	A flow between divide block tro integrator nodes and tree outputs block
Q19	A flow between integrator block and aoutport block and two outputs block and one add block with no flows
Q20	A flow between 4 transfer delay with two subsystems.

5.4.4. Análisis de resultados del experimento

La Tabla 5-8 presenta los resultados del primer experimento en el que es posible encontrar las siguientes métricas (solo se comentan los resultados buenos, excelentes y perfectos): La precisión y recall son al menos buenas para todas las consultas y excelentes en promedio. La Precisión es buena para 8 consultas (32 %) y excelente para 17 (68 %) siendo perfecta (1,00) para 13 consultas (52 %). El Recall es bueno para 3 consultas (12 %) y excelente para 22 (88 %) siendo perfecto (1,00) para 20 consultas (80 %). F1 es perfecto (1,00) para 10 consultas (40 %).

Por otro lado, la Tabla 5-9 presenta los resultados del segundo experimento que compara las capacidades de Simulink para recuperar modelos y el enfoque presentado. Un análisis de los resultados muestra que: La Precisión y el Recall son al menos buenos para todas las consultas y excelentes en promedio. La Precisión es buena para 3 consultas (12 %) y excelente para 14 (56 %) siendo perfecta (1,00) para 6 consultas (25 %). El Recall es bueno para 18 consultas (72 %) y excelente para 16 (64 %) siendo perfecta (1,00) para 11 consultas (44 %). F1 es perfecto (1,00) para 3 consultas (40 %).

En ambos experimentos, el enfoque presentado de transformar los artefactos del sistema existentes en SRL para ser indexados en el repositorio de artefactos del sistema, es una solución valiosa para hacer que la reutilización de modelos lógicos y físicos sea efectiva. El nivel de Precisión y Recall ha sido excelente para la mayoría de las consultas. Los resultados también sugieren que un usuario podría encontrar casi todos los modelos que podrían reutilizarse para un propósito determinado. Aquí, es necesario señalar, que es más importante recuperar todos los posibles modelos reutilizables que recuperar modelos que no son reutilizables. La puntuación F1 también es prometedora, lo que sugiere que el enfoque presentado se puede utilizar para implementar eficazmente una estrategia de reutilización.

5.4.5. Validación Hipótesis H2

Considerando los resultados obtenidos en el experimento, se concluye que se valida la Hipótesis H2: **Mediante la posibilidad de tener un mecanismo de interoperabilidad**

se puede mejorar la selección de artefactos. La solución propuesta mejora los niveles de selección de artefactos.

5.5. Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

Con el propósito de comparar los niveles de trazabilidad alcanzados utilizando la solución propuesta, se desarrolló un caso de estudio utilizando información sobre dominio ferroviario, más específicamente de EIRENE [197].

Se utilizó la metodología de interoperabilidad propuesta en esta tesis para desarrollar dos conectores, el primero que permitiera la realización de los procesos de interpretación semántica sobre el archivo de Word que contiene la especificación de requisitos textuales. El segundo conector para obtener la estructura de desglose de producto de un modelo SysML en Rational Rhapsody [2].

Una vez obtenida la información, siguiendo la metodología, la misma sería representada en SRL, para su posterior transformación a RSHP, con el propósito de hacer uso de las tecnologías de reutilización que ofrece el RSHP mediante la herramienta Traceability Studio [2].

Se desarrolló un algoritmo para descubrir trazas basada en la información semántica de los elementos de un diagrama de bloques en Rational Rhapsody (S1) y una especificación de requisitos (S2) en Word, luego se calcularon métricas de Precisión y Recall, el resultado de esto se comparó con un algoritmo de búsqueda de trazas textual entre las mismas fuentes de información [2].

Este experimento se publicó en [2].

5.5.1. Objetivo

El principal objetivo definido para la validación de este experimento es comparar dos mecanismos de descubrimiento de trazas: (1) Descubrimiento de trazas basados en comparaciones textuales, (2) Descubrimiento de trazas basados en comparaciones semánticas [2].

5.5.2. Diseño

Un proceso de descubrimiento de enlaces de trazabilidad puede verse como un sistema de búsqueda en el que, dada una consulta (un artefacto del sistema de origen, por ejemplo, requisitos) y un conjunto de recursos (un conjunto objetivo de artefactos del sistema, por ejemplo, elementos en un modelo SysML), es necesario para establecer cuál es el mejor conjunto de asignaciones para el artefacto del sistema de origen en el conjunto de recursos de destino [2]. En este experimento, se han seleccionado algunas fuentes públicas de requisitos y modelos SysML. Más específicamente, la especificación de requisitos del sistema de la red europea mejorada de radio ferroviaria integrada (EIRENE) (versión 15.4.0) ha sido seleccionada como la fuente principal de requisitos [2]. Este documento incluye un conjunto de requisitos para los aspectos de interoperabilidad del sistema ferroviario dentro de la Comunidad Europea. También contiene requisitos funcionales y no funcionales para sistemas de radio ferroviarios [2].

Por otro lado, como recursos de destino, se han creado algunos modelos SysML creados con la herramienta IBM Rhapsody que representan la estructura de desglose proporcionada por los alcances del producto de nivel IRIS 5. Estos modelos son una representación lógica de la documentación desarrollada por International Railway Industry Standard [197] y proporcionan un desglose que se utiliza como guía para el proceso de auditoría de productos de los vehículos que se mueven en un ferrocarril, por ejemplo, locomotoras, vagones de ferrocarril y autocares. En este contexto, el experimento para comparar los algoritmos de descubrimiento de trazas se ha diseñado de la siguiente forma [2]:

5.5 Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

1. Diseñar un vocabulario basado en el dominio, O , para representar los conceptos y relaciones que se utilizaron para representar los artefactos del sistema en el dominio ferroviario [2].
2. Crear un conjunto de patrones P_R y P_{SR^R} / $P_R \cap P_{SR^R} = \emptyset$, para representar tanto la especificación de requisitos del sistema donde R representa el conjunto de requisitos de la especificación proporcionada por EIRENE [197] y SR^R es el conjunto de elementos del modelo para la especificación R [2].
3. Seleccionar un conjunto de requisitos basados en los patrones anteriores:

$$R = \{R^1, R^2, \dots, R^k, \dots, R^m\}$$

Donde $\#R^k$ representa el numero de requisitos de sistema en la especificación R^k , para este caso la especificación de requisitos presentados en la Tabla 5-12 [2].

4. Para cada especificación de requisitos, R^k , se definió un conjunto de elementos del modelo que tienen relación con los requisitos seleccionados en el numeral anterior, SR^{R^k} , donde $\#SR^{R^k}$ representa el numero de elementos del modelo en la especificación $\#SR^{R^k}$, estos componentes se presentan en la Tabla 5-13.
5. Ejecutar la función de trazabilidad implementada en la parte superior de la API de CAKE para descubrir trazas entre los elementos del modelo, $\#SR^{R^k}$, y los requisitos, R^k . Para cada requisito en R^k , buscar cuál es el mejor conjunto de asignaciones en $\#SR^{R^k}$. Para ello, se utilizarán dos métodos de coincidencia: 1) basado en texto y 2) basado en conceptos (patrones semánticos).
6. Extraer métricas de Precisión (P), Recall (R) y la puntuación F1 (la media armónica de precisión y recall) haciendo una comparación de los resultados esperados y generados. Siendo $P = tp / tp + fp$, $R = tp / tp + fn$ y $F1 = 2 P * R / P + R$, donde se da una especificación de requisitos R^k , utilizada como una consulta, y los elementos del modelo en $\#SR^{R^k}$, utilizados como recursos objetivo: Tp (verdadero positivo) es “el número de elementos del modelo en $\#SR^{R^k}$ que se han recuperado y representan asignaciones correctas”, fp (falso positivo) es “la cantidad de elementos del modelo en $\#SR^{R^k}$ que se han recuperado y representan asignaciones incorrectas”,

tn (verdadero negativo) es el número de elementos del modelo en $\#SR^{R^k}$ que no se han recuperado y representan asignaciones incorrectas y fn (falso negativo) es “el número de elementos del modelo en $\#SR^{R^k}$ que no se han recuperado y representan asignaciones correctas”.

7. Verificar la solidez de la comparación realizando pruebas de hipótesis estadísticas.

5.5.3. Ejecución del experimento

La Tabla 5-18, muestra las métricas de precisión, recall y la medida F1 de los dos enfoques. Las dos primeras columnas corresponden al identificador de prueba; las siguientes tres columnas contienen los valores de las métricas cuando se ejecuta el enfoque basado en conceptos para descubrir enlaces de trazabilidad, en este enfoque se utilizó una ontología como conocimiento subyacente para representar todos los artefactos del sistema. Después de eso, el segundo conjunto de columnas muestra los valores de las métricas cuando se ejecuta un enfoque basado en texto [2].

Adicionalmente a lo definido en el diseño del experimento se calcularon las métricas de precisión, recall y F1 de los dos enfoques utilizando como fuentes de información, R^k , contra R^k , con el propósito de identificar posibles requisitos relacionados entre sí, se utilizaron los requisitos de la Tabla 5-12, esta información puede ser visualizada en 5-14.

Tabla 5-12: Listado de requisitos de EIRENE utilizados para realizar la experimentación [2].

No	Requirement
R ₁	A railway GSM network is also likely to have external interfaces to: (I) - private railway fixed networks; - public operator networks; - controller equipment; - specialised railway systems (eg train control systems).
continúa en la siguiente página	

Tabla 5-12 – Continuación de la página anterior.

No	Requirement
R ₂	Enhanced multi-level precedence and pre-emption: This GSM specification is to be implemented in order to achieve the high performance requirements necessary for emergency group calls. It is also necessary to meet different grades of service requirements for different types of communications traffic on the system (eg safety (eg train control system), operational and administrative communications). (I)
R ₃	Location dependent addressing: Train drivers need to be able to contact controllers and other staff at the push of a single button. As the train moves through different areas, controllers are liable to change. As a consequence it is necessary to provide a means of addressing calls from a train to certain functions based on the location of the train. (I)
R ₄	Many trains employ multiple active traction vehicles. Where these vehicles are not connected by on-train wiring, it shall be possible for a permanent radio connection to be established between each of the active cabs. (I)
R ₅	Many trains employ multiple active traction vehicles. Where these vehicles are not connected by on-train wiring, it shall be possible for a permanent radio connection to be established between each of the active cabs. (I)

continúa en la siguiente página

Tabla 5-12 – Continuación de la página anterior.

No	Requirement
R ₆	The call will be established from the active cab of the lead traction vehicle. Each of the other cabs on the train will be contacted using its functional number (registered by the other drivers prior to the establishment of the call). The procedure for setting up a multi-party call is outlined in figure 5-2. The multi-party call shall have 'Railway operation' priority (see section 10.2) and whilst on-going a 'multi-drivers' indication shall be displayed permanently at all Cab radios.
R ₇	On activation of the "call other drivers on the same train" function, the MMI shall provide additional guidance to the user in the establishment and management of a Multi-Party call.
R ₈	An emergency power supply should be provided for Cab radios which will enable the driver's radio to continue to operate for a period of 6 hours in the event of failure of the train's main power supply, based on the following cycle (see section 4.5.21): - point-to-point calls 20 %; - group calls 5 %; - standby 75 %.
R ₉	The driver and other in-cab equipment shall be protected against all electrical hazards arising from EIRENE mobile equipment as defined in [EN 50153].
R ₁₀	The following list catalogues the interfaces that should be provided by the Cab Radio to the on-train systems: - Train borne recorder; - ERTMS/ETCS interface; - Public Address; - UIC Intercom; - Driver's Safety Device; - Other interfaces.
R ₁₁	The Operational radio user shall be protected against all electrical hazards arising from the mobile equipment as defined in [EN 50153].

continúa en la siguiente página

5.5 Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

Tabla 5-12 – Continuación de la página anterior.

No	Requirement
R ₁₂	The User Identifier Number (UIN) shall be one of the following numbers (as identified by the CT): - Train Number (TN): a number given to a train by operational staff for a particular journey. This number shall be unique for the duration of the journey. - Note. For certain Train Numbers (e.g. 1234 and 123), a risk exists when dialling a number by keying in individual digits e.g. by the dispatcher.
R ₁₃	If a more accurate way of location determination is used, then position information shall be provided to the radio system which shall be used to associate the short code with the correct called party subscriber number.
R ₁₄	A Railway emergency call is a high priority call for informing drivers, controllers and other concerned personnel of a level of danger requiring all Railway movements in a pre-defined area to stop. Two types of Railway emergency calls are defined: (I) - Train emergency calls (for Railway emergencies whilst not involved in Shunting operations); - Shunting emergency calls (for Railway emergencies whilst involved in Shunting operations).
R ₁₅	The maximum power consumption of the Rolling Stock Components shall be 2500 kw
R ₁₆	The maximum power consumption of the Auxiliary systems shall be 40 kw
R ₁₇	The maximum power consumption of the Braking systems shall be 30 kw
R ₁₈	The maximum power consumption of the Heating, ventilating and air conditioning shall be 300 kw

continúa en la siguiente página

Tabla 5-12 – Continuación de la página anterior.

No	Requirement
R ₁₉	The maximum power consumption of the interiors shall be 23 kw
R ₂₀	The maximum power consumption of the passenguer information system shall be 500 w
R ₂₁	The Rolling stock component shall have 1 auxiliary system
R ₂₂	The Rolling stock component shall have 1 braking system
R ₂₃	The Rolling stock component shall have 1 cabinet
R ₂₄	The Rolling stock component shall have 1 cabling
R ₂₅	The Rolling stock component shall have 1 car body
R ₂₆	The Rolling stock component shall have 1 car body fitting
R ₂₇	The Rolling stock component shall have 1 communication system
R ₂₈	The Rolling stock component shall have 1 coupler
R ₂₉	The Rolling stock component shall have 20 doors
R ₃₀	The Rolling stock component shall have 1 Heating, ventilating and air conditioning system
R ₃₁	The Rolling stock component shall have 1 interior
R ₃₂	The Rolling stock component shall have 1 lighting system
R ₃₃	The Rolling stock component shall have 1 on board vehicle
R ₃₄	The Rolling stock component shall have 1 passenger information system
R ₃₅	The Rolling stock component shall have 1 power system
R ₃₆	The Rolling stock component shall have 1 propulsion system
R ₃₇	The Rolling stock component shall have 1 tilt system
R ₃₈	The propulsion shall have 1 gear box
R ₃₉	The propulsion shall have 1 mechanical transmission
R ₄₀	The propulsion shall have 1 power converter
R ₄₁	The propulsion shall have 1 traction control unit

continúa en la siguiente página

5.5 Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

Tabla 5-12 – Continuación de la página anterior.

No	Requirement

5.5.4. Análisis de resultados del experimento

Según los resultados, el enfoque basado en conceptos es en general mejor que el basado en texto tanto en precisión como en Recall, como se muestra en la Figura 5-1. La razón principal de este comportamiento es que los enfoques basados en conceptos pueden aprovechar la explotación de conceptos y relaciones semánticas, mientras que el enfoque basado en texto solo puede realizar comparaciones de cadenas. Sin embargo, los valores de precisión siguen siendo bajos y se esperarían valores más altos. Esto se debe a que el contexto del experimento (palabras vacías, acrónimos, etc.) es bastante genérico y una versión más personalizada de la ontología para el dominio ferroviario podría implicar mejores resultados en términos de precisión [2].

Por otro lado, se ha realizado una prueba de hipótesis estadística para demostrar si los resultados variarían según el tipo de método utilizado para descubrir vínculos de trazabilidad. Para ello, se ha formulado una comparación de los valores de precisión de ambos métodos (basados en texto y basados en conceptos) a través de las siguientes hipótesis:

Hipótesis 1 (H Exp₀): *No hay cambios en el cálculo de precisión cuando se aplica un enfoque basado en texto o basado en conceptos*

Hipótesis 2 (H Exp₁): *Hay cambios en el cálculo de precisión cuando se aplica un enfoque basado en texto o basado en conceptos*

Para ejecutar la prueba de hipótesis estadísticas, se ha utilizado la Prueba-F con alfa 0.05 para asegurar que las varianzas sean desiguales (hay significación estadística). Después de eso, la prueba-t de dos muestras asumiendo varianzas desiguales se ha realizado

Tabla 5-13: Elementos de la estructura de desglose de productos en Rational Rhapsody [2].

No	Rhapsody element
E ₁	Air_supply_system
E ₂	Auxiliary_electric_system
E ₃	Hydraulic_system
E ₄	Auxiliary_systems
E ₅	Magnetic_track_brake_equipment
E ₆	Eddy_current_brake_equipment
E ₇	Brake_control_system
E ₈	Emergency_brake_equipment
E ₉	Friction_brake_equipment
E ₁₀	Braking_System
E ₁₁	Interior_equipment
E ₁₂	Interior_architecture
E ₁₃	Toilet_system
E ₁₄	Interiors
E ₁₅	Electronic_rear_mirror
E ₁₆	Automatic_Train_Operation_unit
E ₁₇	Electronic_Train_Control_System
E ₁₈	Fault_data_logger
E ₁₉	Heritage_Automatic_Train_Protection_unit
E ₂₀	System_capture_unit
E ₂₁	Train_Control_Management_System
E ₂₂	Voice_recorder
E ₂₃	On_board_vehicle_control
E ₂₄	Billing_System
E ₂₅	Central_Passenger_information_System_unit
E ₂₆	Driver_Machine_Interface_for_train
E ₂₇	Public_Address_System
E ₂₈	Safety_Alarm_Systems
E ₂₉	Seat_reservation
E ₃₀	Passenger_information_System
E ₃₁	Traction_motor
E ₃₂	Gear_box
E ₃₃	Mechanical_transmission
E ₃₄	Traction_Control_Unit
E ₃₅	Propulsion
E ₃₆	Propulsion
E ₃₇	Braking_System
E ₃₈	Interiors
E ₃₉	On_board_vehicle_control
E ₄₀	Passenger_information_System
E ₄₁	Rolling_Stock_Components

5.5 Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

Tabla 5-14: Métricas de Precision, Recall y F1 resultado de la ejecución de algoritmos de descubrimiento de trazas, usando búsquedas basadas en texto y búsquedas basadas en semántica, sobre elementos de la especificación de requisitos contra otros elementos de la misma especificación de requisitos [2].

No	Semantic			Textual		
	Precision	Recall	F1	Precision	Recall	F1
R1	0.00	0.00	0.00	0.33	1.00	0.50
R2	0.00	0.00	0.00	0.20	1.00	0.33
R3	0.00	0.00	0.00	0.00	0.00	0.00
R4	0.00	0.00	0.00	0.00	0.00	0.00
R5	0.00	0.00	0.00	0.11	1.00	0.20
R6	0.04	0.40	0.07	0.25	0.20	0.22
R7	0.10	0.25	0.14	0.00	0.00	0.00
R8	0.03	0.20	0.05	0.00	0.00	0.00
R9	0.00	0.00	0.00	0.00	0.00	0.00
R10	0.00	0.00	0.00	0.25	1.00	0.40
R11	0.08	0.50	0.13	0.00	0.00	0.00
R12	0.00	0.00	0.00	1.00	0.25	0.40
R13	0.00	0.00	0.00	0.13	1.00	0.22
R14	0.06	0.17	0.08	0.00	0.00	0.00
R15	0.00	0.00	0.00	0.00	0.00	0.00
R16	0.00	0.00	0.00	0.00	0.00	0.00
R17	0.05	0.50	0.09	0.04	0.50	0.08
R18	0.00	0.00	0.00	0.00	0.00	0.00
R19	0.00	0.00	0.00	0.04	1.00	0.07
R20	0.00	0.00	0.00	0.06	0.50	0.10
R21	0.00	0.00	0.00	0.00	0.00	0.00
R22	0.05	0.17	0.07	0.05	0.17	0.08
R23	0.12	0.50	0.19	0.00	0.00	0.00
R24	0.18	0.60	0.27	0.00	0.00	0.00
R25	0.24	0.67	0.35	0.00	0.00	0.00
R26	0.29	0.71	0.42	0.00	0.00	0.00
R27	0.29	0.75	0.41	0.00	0.00	0.00
R28	0.41	0.78	0.54	0.00	0.00	0.00
R29	0.53	0.80	0.64	0.00	0.00	0.00
R30	0.43	0.69	0.53	0.00	0.00	0.00
R31	0.59	0.71	0.65	0.04	0.07	0.05
R32	0.52	0.85	0.65	0.00	0.00	0.00
R33	0.71	0.80	0.75	0.00	0.00	0.00
R34	0.62	0.72	0.67	0.06	0.06	0.06
R35	0.68	0.94	0.79	0.00	0.00	0.00
R36	0.60	0.83	0.70	0.00	0.00	0.00
R37	0.76	0.80	0.78	0.00	0.00	0.00
R38	0.00	0.00	0.00	0.00	0.00	0.00
R39	0.13	0.33	0.18	0.00	0.00	0.00
R40	0.50	0.50	0.50	0.00	0.00	0.00
R41	0.75	0.38	0.50	0.08	0.25	0.12
Avg	0.21	0.35	0.25	0.06	0.19	0.07

Figura 5-1: Precisión para cada caso de prueba y enfoque.

con un alfa 0.05 para afirmar si se rechaza o no. De acuerdo con la Tabla 5-15, se puede rechazar, ya que t Stat es menor que “- t Crítico (dos colas)”. En conclusión, el método basado en patrones que explota las relaciones semánticas puede mejorar en términos de precisión el problema de descubrir vínculos entre dos tipos de artefactos del sistema. Adicionalmente se realizaron Prueba-F con alfa 0.05 para las variables Recall(ver Tabla 5-16) y F1(ver Tabla 5-17).

También es posible observar en la Tabla 5-18, que el promedio de la métrica de Recall y F1 es mayor en el enfoque basado en conceptos. De acuerdo a los niveles definidos en 5-2 por [4], el enfoque basado en semántica tiene un nivel “bueno” de precisión al igual que el enfoque basado en texto, aunque el nivel de precisión del primero es mayor con un promedio de 0.49 respecto a un 0.45 respectivamente, en el caso de Recall el nivel del enfoque basado en semántica es “Excelente”, con un valor de 0.84 frente a un nivel “Aceptable” con un valor de 0.60. queda evidenciado que el enfoque basado en semántica es superior al basado en texto tanto a nivel de precisión como recall.

5.5 Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

Tabla 5-15: Prueba de hipótesis estadística, la prueba t de dos muestras que asume varianzas desiguales para comparar los procesos de descubrimiento de enlaces de trazabilidad basados en texto y basados en conceptos para métricas de precisión [2].

	<i>Proceso de recuperación de trazabilidad basado en conceptos</i>	<i>Proceso de recuperación de trazabilidad basado en texto</i>
Media	0.4945	0.4509
Varianza	0.1255	0.1742
Muestra de observaciones	41	41
Estimadas	0	
Df	78	
Estadístico t	0.5094	
$P(T_j=t)$ una-cola	0.306	
Valor crítico de t (una-cola)	1.6646	
$P(T_j=t)$ dos colas	0.6119	
Valor crítico de t (dos colas)	1.9908	

Tabla 5-16: Prueba de hipótesis estadística para la métrica Recall, la prueba t de dos muestras que asume varianzas desiguales para comparar los procesos de descubrimiento de enlaces de trazabilidad basados en texto y basados en conceptos [2].

	<i>Proceso de descubrimiento de enlaces basado en conceptos</i>	<i>Proceso de descubrimiento de enlaces basado en comparaciones textuales</i>
Media	0.839	0.5955
Varianza	0.1102	0.1825
Muestra de observaciones	41	41
Estimadas	0	
Df	75	
Estadístico t	2.8818	
$P(T_j=t)$ una-cola	0.0026	
Valor crítico de t (una-cola)	1.6654	
$P(T_j=t)$ dos colas	0.0052	
Valor crítico de t (dos colas)	1.9921	

Tabla 5-17: Prueba de hipótesis estadística para la métrica F1, la prueba t de dos muestras que asume varianzas desiguales para comparar los procesos de descubrimiento de enlaces de trazabilidad basados en texto y basados en conceptos [2].

	<i>Proceso de descubrimiento de enlaces basado en conceptos</i>	<i>Proceso de descubrimiento de enlaces basado en comparaciones textuales</i>
Media	0.5563	0.4546
Varianza	0.1169	0.1433
Muestra de observaciones	41	41
Estimadas	0	
Df	79	
Estadístico t	1.2759	
P($T_j=t$) una-cola	0.1029	
Valor crítico de t (una cola)	1.6644	
P($T_j=t$) dos colas	0.2057	
Valor crítico de t (dos colas)	1.9905	

Tabla 5-18: Precision, Recall an F1 for Scenario 1 [2].

No	Semantic			Textual		
	Precision	Recall	F1	Precision	Recall	F1
t_1	0.20	0.33	0.25	0.00	0.00	0.00
t_2	0.50	0.40	0.44	0.00	0.00	0.00
t_3	0.03	1.00	0.06	0.00	0.00	0.00
t_4	0.03	1.00	0.06	1.00	0.50	0.67
t_5	1.00	1.00	1.00	0.00	0.00	0.00
t_6	1.00	1.00	1.00	0.00	0.00	0.00
t_7	0.25	1.00	0.40	0.29	1.00	0.44
t_8	0.00	0.00	0.00	0.00	0.00	0.00
t_9	1.00	0.33	0.50	0.00	0.00	0.00
t_{10}	0.07	1.00	0.14	0.08	1.00	0.15
t_{11}	0.00	0.00	0.00	0.00	0.00	0.00

continúa en la siguiente página

5.5 Experimento sobre algoritmos de descubrimiento de traza entre dos fuentes de información distintas

Tabla 5-18 – Continuación de la página anterior.

No	Semantic			Textual		
	Precision	Recall	F1	Precision	Recall	F1
t₁₂	0.00	0.00	0.00	0.00	0.00	0.00
t₁₃	0.00	0.00	0.00	0.00	0.00	0.00
t₁₄	0.67	0.67	0.67	1.00	0.67	0.80
t₁₅	0.33	1.00	0.50	0.50	1.00	0.67
t₁₆	1.00	1.00	1.00	1.00	0.50	0.67
t₁₇	0.60	1.00	0.75	0.50	0.67	0.57
t₁₈	0.08	1.00	0.14	0.09	1.00	0.17
t₁₉	1.00	1.00	1.00	1.00	0.50	0.67
t₂₀	0.67	1.00	0.80	1.00	1.00	1.00
t₂₁	0.50	1.00	0.67	0.33	0.50	0.40
t₂₂	1.00	1.00	1.00	0.00	0.00	0.00
t₂₃	0.50	1.00	0.67	0.67	1.00	0.80
t₂₄	1.00	1.00	1.00	1.00	0.50	0.67
t₂₅	0.67	1.00	0.80	1.00	1.00	1.00
t₂₆	0.50	1.00	0.67	0.54	1.00	0.70
t₂₇	0.21	1.00	0.35	0.15	0.67	0.25
t₂₈	0.14	1.00	0.25	0.08	0.50	0.13
t₂₉	0.50	1.00	0.67	0.00	0.00	0.00
t₃₀	0.36	1.00	0.53	0.40	1.00	0.57
t₃₁	0.80	1.00	0.89	0.75	0.75	0.75
t₃₂	1.00	1.00	1.00	1.00	1.00	1.00
t₃₃	0.50	1.00	0.67	1.00	1.00	1.00
t₃₄	0.33	1.00	0.50	0.50	1.00	0.67
t₃₅	0.75	1.00	0.86	0.86	1.00	0.92
t₃₆	0.75	1.00	0.86	0.86	1.00	0.92

continúa en la siguiente página

Tabla 5-18 – Continuación de la página anterior.

No	Semantic			Textual		
	Precision	Recall	F1	Precision	Recall	F1
t ₃₇	0.07	1.00	0.14	0.08	1.00	0.15
t ₃₈	0.67	0.67	0.67	1.00	0.67	0.80
t ₃₉	0.50	1.00	0.67	0.67	1.00	0.80
t ₄₀	0.18	1.00	0.31	0.20	1.00	0.33
t ₄₁	0.90	1.00	0.95	0.95	1.00	0.97
Avg	0.49	0.84	0.56	0.45	0.60	0.45

5.5.5. Validación Hipótesis H3

Considerando los resultados obtenidos en el experimento, se concluye que se valida la Hipótesis H3: **Mediante la posibilidad de tener un mecanismo de interoperabilidad se puede mejorar el descubrimiento de trazas.** Un método de descubrimiento automático de trazas utilizando la semántica obtenida por el marco de interoperabilidad propuesto mejora los niveles de descubrimiento de trazas.

5.6. Validación industrial del marco de interoperabilidad para procesos de ingeniería

A mediados de los años 70, NASA introduce el término madurez tecnológica, en inglés “technology readiness”, esto con el propósito de evaluar y transmitir la madurez de una nueva tecnología [198,199], este concepto se consolida en un modelo de madurez de tecnologías de 9 niveles que inicialmente fue definido para tecnologías aeroespaciales, pero pronto fue aplicado tanto a Hardware como a Software. Según lo descrito

5.6 Validación industrial del marco de interoperabilidad para procesos de ingeniería

en [198, 200, 201], en donde se detallan estos niveles de madurez, las tecnologías desarrolladas para la solución propuesta tienen un nivel de madurez TRL-9, considerando que las mismas están implementadas en aplicaciones de uso comercial, que actualmente apoyan los procesos de ingeniería en empresas como:

- AIRBUS.
- THALES.
- SAFRAN.
- FORD.

La definición de un TRL-9 para el software es: “El sistema ha sido operado con éxito en el entorno operativo” [200]. Lo que implica un grado de madurez alto. En el caso de los componentes desarrollados bajo la solución propuesta que han sido implementados en las herramientas de TRC, durante su desarrollo han pasado distintas fases de pruebas, considerando que TRC desarrolla su tecnología bajo un entorno de Integración Continua, la cual es una práctica de desarrollo de software que implica la ejecución de múltiples pruebas automáticas en entornos de desarrollo colaborativo, en donde cada miembro realiza una integración que dispara la ejecución de múltiples pruebas automáticas [202].

Previo a la integración de nuevos componentes a la solución que será desplegada en los entornos de los clientes. TRC somete los componentes a la aplicación de diversas pruebas que garantizan la calidad del software, esto en vista de que los clientes de TRC cuentan con entornos de ingeniería sumamente complejos, por lo que se deben cuidar aspectos como tiempos de respuesta, disponibilidad, eficiencia entre otros. Una vez habiendo concluido satisfactoriamente los estándares de calidad definidos por TRC, los componentes son desplegados en los entornos de los clientes en donde apoyan los distintos procesos.

Los componentes que se describen en la sección 4.3.1 han pasado por los procesos descritos anteriormente, y actualmente están siendo utilizados en entornos de diversos tamaños y complejidades, de acuerdo a los clientes que hacen uso de las versiones de las herramientas de TRC.

6 Conclusiones y futuras líneas de investigación

“Dudo, luego pienso”

Descartes

La representación unificada del conocimiento desde una perspectiva de interoperabilidad aporta muchos beneficios. En el presente capítulo se detallan las conclusiones obtenidas producto de la etapa de evaluación, desde una perspectiva de cumplimiento de los objetivos establecidos al inicio de la investigación. La presentación de las conclusiones se divide en conclusiones científicas, las cuales se basan en los resultados de los experimentos del Capítulo 5, y conclusiones tecnológicas, que son conclusiones obtenidas del desarrollo de la actividad doctoral. En la parte final de este capítulo se proponen futuras líneas de investigación que podrían fortalecer el marco propuesto y que están muy relacionadas con los hallazgos encontrados durante la realización de esta tesis.

6.1. Conclusiones científicas

En esta sección se detallan las conclusiones de cara a los objetivos establecidos en el Capítulo 1. Se presentan los hallazgos que permiten dar respuesta a las hipótesis y objetivos trazados al inicio de la tesis.

- Se comprobó que el marco de interoperabilidad propuesto en esta tesis, ofrece la posibilidad de contar con un mecanismo de interoperabilidad entre distintas fuentes de información, dicho mecanismo mejora las capacidades de interoperabilidad,

ofreciendo la posibilidad no solo de realizar intercambios de conocimiento sino también proporcionando la posibilidad de invocar operaciones disponibles en las fuentes de conocimiento. Considerando la evaluación presentada en el Capítulo 5, la cual demostró que la representación unificada del conocimiento proporciona fortalezas especialmente a nivel semántico, es posible concluir que el marco propuesto se constituye como una alternativa prometedora para mejorar los niveles de interoperabilidad en las organizaciones.

- Fue posible comprobar el marco propuesto mejora las capacidades de reutilización del conocimiento para distintos artefactos de ingeniería, considerando casos en los que los procesos de una organización requieren la interacción entre distintos artefactos del sistema, la reutilización de conocimientos se vuelve un factor clave para optimizar los procesos, esto, aunado a la posibilidad de contar con un repositorio de conocimientos y con servicios nativos de indexación y recuperación de artefactos, permiten la consolidación de un entorno colaborativo.
- Se pudo corroborar que contar con una representación unificada del conocimiento, mejora las capacidades de desarrollo de algoritmos de descubrimiento de traza, entre otras cosas debido a que para determinar la similitud entre elementos se considera la semántica de los mismos y no comparaciones textuales. Esto facilita la comparación entre elementos de distintas fuentes y formatos, pero con semánticas similares. En vista de que actualmente la ingeniería de sistemas, considera la trazabilidad un proceso clave para impulsar la ingeniería colaborativa facilitando la tarea de descubrir y mapear artefactos similares, la mejora de los algoritmos de comparación podrían ser una contribución importante a nivel industrial. Los experimentos reflejaron que mientras el artefacto a comparar sea más rico semánticamente el algoritmo puede ser más exacto, es por eso que su aplicación es más exacta utilizando artefactos como “componentes de modelos”, en comparación con texto en lenguaje natural, aun y cuando el lenguaje natural puede ser representado como grafos semánticos mediante la aplicación de patrones textuales.
- Fue posible analizar, las distintas alternativas existentes para el desarrollo de inter-

operabilidad, se encontraron propuestas para diversos dominios, y con diferentes enfoques, se hizo una valoración en el Capítulo 2 de las diferentes propuestas disponibles, y se concluyó que la alternativa propuesta podría ser complementaria a muchos de los marcos de interoperabilidad, sobre todo en lo referente a la interoperabilidad semántica.

- Durante el desarrollo de esta tesis, se realizaron publicaciones en congresos, revistas indexadas, contribuciones en proyectos nacionales y regionales. Adicionalmente el marco propuesto en esta tesis se ha integrado en forma de múltiples conectores a distintas fuentes de información, en las herramientas de TRC. El detalle del impacto y la difusión de la tesis se presenta en el apartado VI.

6.2. Conclusiones tecnológicas

- Fue posible comprobar que la presente tesis, propone una arquitectura conceptual, así como también una arquitectura tecnológica sobre herramientas que actualmente están siendo comercializadas en la industria, y que apoyan procesos de ingeniería de sistemas en varios dominios, como ser aeroespacial, automotriz, entre otros. Adicionalmente, durante el desarrollo de esta tesis se desarrollaron nuevas tecnologías, así como también se realizaron mejoras a tecnologías existentes que están disponibles de forma gratuita. También, en la sección 3.4 se presenta una metodología iterativa para el desarrollo de conectores SRL, en donde se detallan los procesos, entradas, salidas y herramientas, esta metodología ha sido aplicada para el desarrollo de los conectores integrados en las herramientas mencionadas en el Capítulo 4, en la cual se ha podido comprobar que la ejecución de la misma permite llevar a cabo el desarrollo de los conectores de forma eficiente y eficaz.
- Como parte de esta tesis, se presenta SRL, el cual es un metamodelo de representación de conocimientos, basado en trabajos anteriores. Este lenguaje se propuso debido a dos factores: (1) la simplicidad de su metamodelo, (2) El potencial que tiene el uso de grafos semánticos para representar conocimiento. Adicionalmente

6.3 Futuras líneas de investigación

en esta tesis se desarrollaron componentes de conversión del SRL a otros lenguajes de representación de conocimiento como ser RDF y RSHP.

- Se llevó a cabo la integración con las herramientas de The Reuse Company, las cuales funcionan con las APIs de CAKE. Para la arquitectura tecnológica descrita en el Capítulo 3, se reutilizó el API de CAKE, y se desarrollaron diferentes componentes de serialización.

6.3. Futuras líneas de investigación

En esta sección se presentan posibles líneas futuras de investigación tanto a nivel tecnológico como científico.

6.3.1. Futuras líneas de investigación científicas

Las posibles líneas de acción a nivel científico identificadas tras finalizar la actividad doctoral son las siguientes:

- Evolucionar aspectos relacionados con la transformación del conocimiento mediante la construcción de un marco de transformación que permita la realización de procesos como: conversión de modelos lógicos a físicos. Este conocimiento está en la mente de los ingenieros y si fuese posible definir marco conceptual y tecnológico capaz de gestionar estas reglas y criterios de forma eficiente, sería factible hacer estas transformaciones de forma automática.
- Desarrollar mas experimentación, incluir casos de estudio sobre calidad, verificación y validación utilizando la metodología propuesta.
- Desarrollar conectores SRL sobre fuentes de información que no son del dominio de la ingeniería de sistemas.
- Proponer instrumentos de validación de modelos SRL que permitan garantizar la consistencia y el grado de fidelidad de los modelos respecto a fuentes de información.

- desarrollar instrumentos de certificación de la correcta implementación del marco de interoperabilidad para procesos de ingeniería propuesto en esta tesis.

6.3.2. Futuras líneas de investigación tecnológicas

Luego de la implementación tecnológica del marco de interoperabilidad para procesos de ingeniería, se han identificado algunas líneas futuras de investigación que se presentan a continuación:

- Desarrollar una plataforma que permita gestionar de forma dinámica los conectores SRL, es decir tener la posibilidad de integrar nuevos conectores de forma dinámica y sin esfuerzo, para poder ser instanciados a demanda e integrados a los procesos de las organizaciones que hagan uso de los conectores SRL.
- Desarrollo de un sistema centralizado de gestión de funcionalidades ofrecidas por los conectores SRL, esto debido a que los conectores se basan en la utilización de las tecnologías de interoperabilidad que los fabricantes de herramientas ponen a disposición de los consumidores, los conectores SRL permiten invocar funcionalidades que son características de las herramientas que generan conocimiento, por ejemplo Simulink (simulación, verificación y validación), Rhapsody (generación de código fuente, módulos de reportería).
- Incluir aspectos tecnológicos relacionados con la seguridad y no repudio de la información obtenida de las fuentes.

Bibliografía

- [1] J. M. Alvarez-Rodríguez, R. Mendieta, J. L. De la Vara, A. Fraga, and J. Llorens, “Enabling system artefact exchange and selection through a linked data layer,” *Journal of Universal Computer Science*, vol. 24, no. 11, pp. 1536–1560, 2018.
- [2] J. M. Alvarez-Rodríguez, R. Mendieta, V. Moreno, M. Sánchez-Puebla, and J. Llorens, “Semantic recovery of traceability links between system artifacts,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 30, no. 10, pp. 1415–1442, 2020.
- [3] T. Clark, R. Jones, L. Jones, and C. Pty, “Organisational interoperability maturity model for c2,” *Proceedings of the 1999 Command and Control Research and Technology Symposium*, vol. 29, 1999.
- [4] J. H. Hayes, A. Dekhtyar, and S. K. Sundaram, “Improving after-the-fact tracing and mapping: supporting software quality predictions,” *IEEE Software*, vol. 22, no. 6, pp. 30–37, 2005.
- [5] Christopher Delp, “Open model-based engineering environments,” NIST. https://www.nist.gov/system/files/documents/2019/04/05/14_delp.pdf (acceso: 03 de septiembre de 2020).
- [6] T. McDermott, D. DeLaurentis, P. Beling, M. Blackburn, and M. Bone, “Ai4se and se4ai: A research roadmap,” *INSIGHT*, vol. 23, pp. 8–14, 03 2020.
- [7] H. Van Der Veer and A. Wiles, “Achieving Technical Interoperability: the ETSI Approach,” *European Telecommunications Standards Institute*, no. 3, p. 29, 2008, [En

- línea]. Disponible en: <https://portal.etsi.org/CTI/Downloads/ETSIApproach/IOPwhitepaperEdition3final.pdf>. Acceso: 05 de enero de 2021.
- [8] R. F. Paige, D. S. Kolovos, and F. A. C. Polack, "An action semantics for mof 2.0," in *Proceedings of the 2006 ACM Symposium on Applied Computing*, ser. SAC '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 1304–1305.
- [9] O. Gotel, J. Cleland-Huang, J. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic, and P. Mäder, "Traceability fundamentals," *Software and Systems Traceability*, pp. 3–22, 10 2012.
- [10] S. Cook, C. Bock, P. Rivett, T. Rutt, E. Seidewitz, B. Selic, and D. Tolbert, "Unified modeling language (UML) version 2.5.1," Object Management Group (OMG), Standard, 12 2017, [En línea]. Disponible en: <https://www.omg.org/spec/UML/2.5.1>.
- [11] J. Llorens, J. Morato, and G. Genova, "Rshp: an information representation model based on relationships," in *Soft computing in software engineering*. Berlin, Heidelberg: Springer, 2004, pp. 221–253.
- [12] C4ISR AWG, "Levels of information systems interoperability (LISI)," *C4ISR Architecture Working Group*, no. 30 March 1998, p. 139, 1998.
- [13] S. Chiu, "Can level of information systems interoperability (lisi) improve dod c4i systems' interoperability?" Master's thesis, Naval Postgraduate School Monterrey CA, 2001.
- [14] J. Siegel and J. Perdue, "Cloud services measures for global use: The service measurement index (smi)," in *2012 Annual SRII Global Conference*. IEEE, 2012, pp. 411–415.
- [15] European Commission, ISA Group, "About Common Assessment Method for Standards and Specifications (CAMSS)," JOINUP. <https://joinup.ec.europa.eu/collection/common-assessment-method-standards-and-specifications-camss/about> (acceso: 16 de agosto de 2020).

- [16] European Commission, *New European Interoperability Framework - Promoting seamless services and data flows for European public administrations*. Luxembourg, Luxembourg: Publication Office of the European Union, 2017, vol. 32, no. 9, [En línea]. Disponible en: <https://ec.europa.eu/isa2/sites/isa/files/eif{ }brochure{ }final.pdf>. Acceso: 05 de enero de 2021.
- [17] R. Jardim-Goncalves, A. Grilo, C. Agostinho, F. Lampathaki, and Y. Charalabidis, "Systematisation of Interoperability Body of Knowledge: The foundation for Enterprise Interoperability as a science," *Enterprise Information Systems*, vol. 7, no. 1, pp. 7–32, 2013.
- [18] D. Chen, "Enterprise interoperability framework," *CEUR Workshop Proceedings*, vol. 200, 2 2006.
- [19] Organization for the Advancement of Structured Information Standards Technical Committees (OASIS TC), "Open services for lifecycle collaboration core specification version 2.0," *OslcCoreSpecification*. <https://archive.open-services.net/bin/view/Main/OslcCoreSpecification> (acceso: 03 de septiembre de 2020).
- [20] J. Alvarez-Rodríguez, J. Llorens, M. Alejandres, and J. M. Fuentes, "Oslc-km: A knowledge management specification for oslc-based resources," *INCOSE International Symposium*, vol. 25, pp. 16–34, 10 2015.
- [21] MoSSEC Project, "Mossec standardization business case for white paper," MoSSEC. http://www.mossec.org/c/document_library/get_file?p_l_id=410483&folderId=349888&name=DLFE-11276.pdf (acceso: 03 de septiembre de 2020).
- [22] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 1, pp. 101–111, 2012.
- [23] LOTAR International, "Lotar standard," LOTAR International. <https://lotar-international.org/lotar-standard/> (acceso: 03 de septiembre de 2020).

-
- [24] Mathworks, "Aircraft longitudinal flight control," Mathworks. <https://es.mathworks.com/help/simulink/slref/aircraft-longitudinal-flight-control.html> (acceso: 03 de septiembre de 2020).
- [25] UC3M Knowledge Reuse Group, "Collaborations," Knowledge Reuse Group. <http://www.kr.inf.uc3m.es/colaborations/> (acceso: 03 de septiembre de 2020).
- [26] UC3M Knowledge Reuse Group, Knowledge Reuse Group. <http://www.kr.inf.uc3m.es/research/> (acceso: 03 de septiembre de 2020).
- [27] I. Cabral and A. Grilo, "Impact of business interoperability on the performance of complex cooperative supply chain networks: A case study," *Complexity (New York, N.Y.)*, vol. 2018, pp. 1–30, 2018.
- [28] P. Wegner, "Interoperability," *ACM Computing Surveys*, vol. 28, pp. 285–287, 1996.
- [29] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business and Information Systems Engineering*, vol. 6, no. 4, p. 239–242, 2014.
- [30] M. Platenius-Mohr, S. Malakuti, S. Grüner, J. Schmitt, and T. Goldschmidt, "File- and api-based interoperability of digital twins by model transformation: An iiot case study using asset administration shell," *Future Generation Computer Systems*, vol. 113, pp. 94 – 105, 2020, [En línea]. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0167739X20302600>. Acceso: 05 de enero de 2021.
- [31] P. A. Laplante and J. F. DeFranco, "Software engineering of safety-critical systems: Themes from practitioners," *IEEE Transactions on Reliability*, vol. 66, no. 3, pp. 825–836, 2017.
- [32] "Embedded software development for safety-critical systems," *ProtoView*, vol. 3, no. 6, 2016.
- [33] European Commission, "Interoperability is at the heart of the new eu digital strategy," European Commission ISA News <https://ec.europa.eu/isa2/news/>

- interoperability-heart-new-eu-digital-strategy_en (acceso: 03 de septiembre de 2020).
- [34] Mathworks Inc. Simulink, “Shaping europe’s digital future,” European Commission https://ec.europa.eu/info/sites/info/files/communication-shaping-europes-digital-future-feb2020_en_4.pdf (acceso: 03 de septiembre de 2020).
- [35] M. Gallaher, A. O’Connor, J. Dettbarn, and L. Gilday, “Cost analysis of inadequate interoperability in the us capital facilities industry,” *National Institute of Standards and Technology (NIST)*, pp. 223–253, 01 2004.
- [36] West Health Institute, “The value of medical device interoperability: Improving patient care with more than \$ 30 billion in annual health care savings,” pp. 1–46, 03 2013, [En línea]. Disponible en: <https://www.westhealth.org/wp-content/uploads/2015/02/The-Value-of-Medical-Device-Interoperability.pdf>. Acceso: 05 de enero de 2021.
- [37] H. Panetto, M. Zdravkovic, R. Jardim-Goncalves, D. Romero, J. Cecil, and I. Mezgár, “New perspectives for the future interoperable enterprise systems,” *Computers in industry*, vol. 79, pp. 47–63, 08 2016.
- [38] M. Fakhar Manesh, M. M. Pellegrini, G. Marzi, and M. Dabic, “Knowledge management in the fourth industrial revolution: Mapping the literature and scoping future avenues,” *IEEE Transactions on Engineering Management*, vol. 68, no. 1, pp. 289–300, 2021.
- [39] M. Liebenberg and M. Jarke, “Information systems engineering with digital shadows: Concept and case studies,” in *Advanced Information Systems Engineering*. Cham: Springer International Publishing, 2020, pp. 70–84.
- [40] F. Bonilla, T. Holzer, and S. Sarkani, “Complexity measure for engineering systems incorporating system states and behavior,” *IEEE Systems Journal*, pp. 1–12, 2020.

-
- [41] M. Herrera, M. Pérez-Hernández, A. K. Parlikad, and J. Izquierdo, "Multi-agent systems and complex networks: Review and applications in systems engineering," *Processes*, vol. 8, no. 3, pp. 1–28, 2020.
- [42] D. Nešić, J. El-khoury, J. Westman, and M. Nyberg, "Building a web-based federated toolchain: Lessons learned from a four-year industrial project," in *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services*, ser. iiWAS2019. New York, USA: Association for Computing Machinery, 2019, p. 512–521.
- [43] G. B. Ozturk, "Interoperability in building information modeling for aeco/fm industry," *Automation in construction*, vol. 113, pp. 103–122, 2020.
- [44] S. Balakrishna and M. Thirumaran, "Chapter 7 - semantic interoperability in iot and big data for health care: a collaborative approach," in *Handbook of Data Science Approaches for Biomedical Engineering*, V. E. Balas, V. K. Solanki, R. Kumar, and M. Khari, Eds. Academic Press, 2020, pp. 185 – 220. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/B9780128183182000076>
- [45] A. Koulou, M. Zemzami, N. El hami, A. Elmir, and N. Hmina, "Optimization in collaborative information systems for an enhanced interoperability network," *International Journal for Simulation and Multidisciplinary Design Optimization*, vol. 11, p. 2, 01 2020.
- [46] M. G. P. Velthuis, J. A. Cruz-Lemus, and M. F. Genero Bocco, *Métodos de investigación en ingeniería del software*. Grupo Editorial RA-MA, 2014.
- [47] W. Kerber and H. Schweitzer, "Interoperability in the digital economy," *J. Intell. Prop. Info. Tech. & Elec. Com. L.*, vol. 8, p. 39, 2017.
- [48] A. Geraci, F. Katki, L. McMonegal, B. Meyer, J. Lane, P. Wilson, J. Radatz, M. Yee, H. Porteous, and F. Springsteel, "Ieee standard computer dictionary: A compilation of ieee standard computer glossaries," *IEEE Std 610*, pp. 1–217, 1991.

- [49] S. R. Ray and A. Jones, "Manufacturing interoperability," *Journal of Intelligent Manufacturing*, vol. 17, no. 6, pp. 681–688, 2006.
- [50] R. C. Motta, K. M. De Oliveira, and G. H. Travassos, "Rethinking interoperability in contemporary software systems," in *2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS)*. IEEE, 2017, pp. 9–15.
- [51] C. H. Asuncion and M. J. van Sinderen, "Pragmatic interoperability: A systematic review of published definitions," in *Enterprise Architecture, Integration and Interoperability*, P. Bernus, G. Doumeingts, and M. Fox, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 164–175.
- [52] A. L. Fraga, M. Vegetti, and H. P. Leone, "Ontology-based solutions for interoperability among product lifecycle management systems: A systematic literature review," *Journal of Industrial Information Integration*, p. 100176, 2020.
- [53] S. Heiler, "Semantic interoperability," *ACM Computing Surveys (CSUR)*, vol. 27, no. 2, pp. 271–273, 1995.
- [54] B. Thalheim, "Syntax, semantics and pragmatics of conceptual modelling," in *Natural Language Processing and Information Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1–10.
- [55] M. Silverstein, "Linguistic theory: syntax, semantics, pragmatics," *Annual review of Anthropology*, vol. 1, no. 1, pp. 349–382, 1972.
- [56] H. Kubicek, R. Cimander, and H. J. Scholl, *Layers of Interoperability*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 85–96.
- [57] H. Kubicek and R. Cimander, "Three dimensions of organizational interoperability. Insights from recent studies for improving interoperability frameworks," *European Journal of ePractice*, no. January, pp. 1–12, 2009.

-
- [58] A. Tolk and J. A. Mugira, "The levels of conceptual interoperability model," in *Proceedings of the 2003 fall simulation interoperability workshop*, vol. 7. Citeseer, 2003, pp. 1–11.
- [59] F. B. Vernadat, "Technical, semantic and organizational issues of enterprise interoperability and networking," *Annual Reviews in Control*, vol. 34, no. 1, pp. 139–144, 2010.
- [60] A. Ouardi, A. Sekkaki, and D. Mammass, "Technical and Semantic Interoperability in the Cloud Broker," *International Journal of Computers*, vol. 01, pp. 47–55, 2016.
- [61] R. L. Craft, "Toward technical interoperability in telemedicine," *Telemedicine Journal & e-Health*, vol. 11, no. 3, pp. 384–404, 2005.
- [62] S. Garde, P. Knaup, E. J. Hovenga, and S. Heard, "Towards semantic interoperability for electronic health records: Domain knowledge governance for openEHR archetypes," *Methods of Information in Medicine*, vol. 46, no. 3, pp. 332–343, 2007.
- [63] M. Sudmanns, D. Tiede, S. Lang, and A. Baraldi, "Semantic and syntactic interoperability in online processing of big earth observation data," *International Journal of Digital Earth*, vol. 11, no. 1, pp. 95–112, 2018.
- [64] S. Jabbar, F. Ullah, S. Khalid, M. Khan, and K. Han, "Semantic interoperability in heterogeneous iot infrastructure for healthcare," *Wireless Communications and Mobile Computing*, vol. 2017, 2017.
- [65] M. T. Horsch, C. Niethammer, G. Boccardo, P. Carbone, S. Chiacchiera, M. Chiricotto, J. D. Elliott, V. Lobaskin, P. Neumann, P. Schiffels *et al.*, "Semantic interoperability and characterization of data provenance in computational molecular engineering," *Journal of Chemical & Engineering Data*, vol. 65, no. 3, pp. 1313–1329, 2019.
- [66] J. Nilsson and F. Sandin, "Semantic interoperability in industry 4.0: Survey of recent developments and outlook," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 127–132.

- [67] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Towards semantic interoperability between internet of things platforms," in *Integration, interconnection, and interoperability of iot systems*. Springer, 2018, pp. 103–127.
- [68] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska, "Semantic interoperability in the internet of things: An overview from the inter-iot perspective," *Journal of Network and Computer Applications*, vol. 81, pp. 111 – 124, 2017.
- [69] B. S. Adamczyk, A. L. Szejka, and O. Canciglieri, "Knowledge-based expert system to support the semantic interoperability in smart manufacturing," *Computers in Industry*, vol. 115, p. 103161, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0166361519303963>
- [70] O. Novo and M. D. Francesco, "Semantic interoperability in the iot: Extending the web of things architecture," *ACM Transactions on Internet of Things*, vol. 1, no. 1, pp. 1–25, 2020.
- [71] V. Margariti, D. Anagnostopoulos, A. Papastilianou, T. Stamati, and S. Angeli, "Assessment of organizational interoperability in e-government: a new model and tool for assessing organizational interoperability maturity of a public service in practice," in *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, 2020, pp. 298–308.
- [72] G. Hatzivasilis, I. Askoxylakis, G. Alexandris, D. Anicic, A. Bröring, V. Kulkarni, K. Fysarakis, and G. Spanoudakis, "The interoperability of things: Interoperable solutions as an enabler for iot and web 3.0," in *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018, pp. 1–7.
- [73] Y. Bouallouche, R. Chenouard, C. Da Cunha, and A. Bernard, "Organizational interoperability between public and private actors in an extended administration," *Enterprise Interoperability: INTEROP-PGSO Vision*, vol. 1, pp. 151–165, 2017.

- [74] M. Zoubeidi, O. Kazar, N. Mesbahi, and S. Benharzallah, "Toward an architecture for the semantic integration in enterprise resource planning," in *2014 International Workshop on Advanced Information Systems for Enterprises*. Tunis, 2014, pp. 45–50.
- [75] T. Benson, *Principles of health interoperability HL7 and SNOMED*. London: Springer Science & Business Media, 2012.
- [76] R. Venanzi, F. Montori, P. Bellavista, and L. Foschini, "Industry 4.0 solutions for interoperability: a use case about tools and tool chains in the arrowhead tools project," in *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2020, pp. 429–433.
- [77] D. Grdr, "Making interoperability visible: A novel approach to understand interoperability in cyber-physical systems toolchains," Ph.D. dissertation, KTH Royal Institute of Technology, Stockholm, Suecia, 01 2016.
- [78] Object Management Group, "OMG Meta Object Facility (MOF) Core Specification, Version 2.5," 2015, OMG. <https://www.omg.org/spec/MOF/2.5.1/PDF> (acceso: 03 de septiembre de 2020).
- [79] M. F. Caro, D. A. Diaz, D. P. Josyula, and J. A. Jimenez, "Metathink: A mof-based metacognitive modeling tool," in *2016 IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. Palo Alto, CA: IEEE, 2016, pp. 351–358.
- [80] S. Crawley, S. Davis, J. Indulska, S. McBride, and K. Raymond, "Meta-meta is better-better!" *Distributed Applications and Interoperable Systems*, pp. 223–234, 1997.
- [81] L. Bichler, "A flexible code generator for mof-based modeling languages," in *2nd OOPSLA Workshop on Generative Techniques in the context of Model Driven Architecture*, 2003.
- [82] D. Gasevic, V. Devedzic, and D. Duric, "A MDA-based approach to the ontology

- definition metamodel,” *Proceedings of the 4TH Workshop on Computational Intelligence and Information Technologies, Niš, Serbia*, pp. 51–54, 2003.
- [83] O. Betari, M. Erramdani, S. Roubi, K. Arrhioui, and S. Mbarki, “Model transformations in the mof meta-modeling architecture: from uml to codeigniter php framework,” in *Europe and MENA Cooperation Advances in Information and Communication Technologies*. Springer, 2017, pp. 227–234.
- [84] T. Vale, E. S. de Almeida, V. Alves, U. Kulesza, N. Niu, and R. de Lima, “Software product lines traceability: A systematic mapping study,” *Information and Software Technology*, vol. 84, pp. 1–18, 2017.
- [85] D. Meedeniya, I. Rubasinghe, and I. Perera, “Traceability establishment and visualization of software artefacts in devops practice: A survey,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 7, 2019.
- [86] D. Amalfitano, V. De Simone, A. R. Fasolino, and S. Scala, “Improving traceability management through tool integration: an experience in the automotive domain,” in *Proceedings of the 2017 International Conference on Software and System Process*, 2017, pp. 5–14.
- [87] N. Mustafa and Y. Labiche, “The need for traceability in heterogeneous systems: a systematic literature review,” in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2017, pp. 305–310.
- [88] I. Gunawan, I. Vanany, and E. Widodo, “Cost-benefit model in improving traceability system: case study in indonesian bulk-liquid industry,” in *Supply Chain Forum: An International Journal*, vol. 20, no. 2. Taylor & Francis, 2019, pp. 145–157.
- [89] T. K. Dasaklis, F. Casino, and C. Patsakis, “Defining granularity levels for supply chain traceability based on iot and blockchain,” in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, 2019, pp. 184–190.

-
- [90] S. J. Mellor, T. Clark, and T. Futagami, "Model-driven development: guest editors' introduction." *IEEE software*, vol. 20, no. 5, pp. 14–18, 2003.
- [91] B. Selic, "Models, software models and uml," in *UML for Real: Design of Embedded Real-Time Systems*. Boston, MA: Springer US, 2003, pp. 1–16.
- [92] J.-L. Voirin, *Model-based System and Architecture Engineering with the Arcadia Method*. Elsevier, 2017.
- [93] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Annals*, vol. 66, no. 1, pp. 141–144, 2017.
- [94] A. M. Madni and M. Sievers, "Model-based systems engineering: motivation, current status, and needed advances," in *Disciplinary convergence in systems engineering research*. Springer, 2018, pp. 311–325.
- [95] S. Abrahão, F. Bourdeleau, B. Cheng, S. Kokaly, R. Paige, H. Stöerrle, and J. Whittle, "User experience for model-driven engineering: Challenges and future directions," in *2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS)*. IEEE, 2017, pp. 229–236.
- [96] M. Ozkaya and F. Erata, "A survey on the practical use of uml for different software architecture viewpoints," *Information and Software Technology*, vol. 121, p. 106275, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950584920300252>
- [97] T. Ho-Quang, R. Hebig, G. Robles, M. R. V. Chaudron, and M. A. Fernandez, "Practices and perceptions of uml use in open source projects," in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, 2017, pp. 203–212.
- [98] R. Platt and N. Thompson, "The past, present, and future of uml," in *Advanced Methodologies and Technologies in Network Architecture, Mobile Computing, and Data Analytics*. IGI Global, 2019, pp. 1452–1460.

- [99] J. Osis and U. Donins, *Topological UML modeling: an improved approach for domain modeling and software development*. Elsevier, 2017.
- [100] T. Ahmad, J. Iqbal, A. Ashraf, D. Truscan, and I. Porres, “Model-based testing using uml activity diagrams: A systematic mapping study,” *Computer Science Review*, vol. 33, pp. 98–112, 2019.
- [101] O. Betari, S. Filali, A. Azzaoui, and M. A. Boubnad, “Applying a model driven architecture approach: Transforming cim to pim using uml,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 14, no. 09, pp. 170–181, 2018.
- [102] V. Vyas, R. G. Vishwakarma, and C. Jha, “Modeling aspects with aodml: Extended uml approach for aodml,” *IJ Engineering and Manufacturing (IJEM)*, vol. 2, pp. 11–22, 2017.
- [103] B. Bouihi and M. Bahaj, “An uml to owl based approach for extracting moodle’s ontology for social network analysis,” *Procedia computer science*, vol. 148, pp. 313–322, 2019.
- [104] B. Rumpe, *Agile Modeling with UML: Code Generation, Testing, Refactoring*. Springer, 2017.
- [105] S. Bran, “Uml 2: a model-driven development tool,” *IBM Systems Journal*, vol. 45, no. 3, pp. 607–620, 2006.
- [106] M. Hause, F. Thom, and A. Moore, “Inside sysml,” *Computing and Control Engineering*, vol. 16, no. 4, pp. 10–15, 2005.
- [107] D. E. O’Leary, “How knowledge reuse informs effective system design and implementation,” *Intelligent Systems, IEEE*, vol. 16, no. 1, pp. 44–49, 2001.
- [108] D. Hislop, R. Bosua, and R. Helms, *Knowledge management in organizations: A critical introduction*. Oxford university press, 2018.

- [109] L. M. Markus, "Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success," *Journal of management information systems*, vol. 18, no. 1, pp. 57–93, 2001.
- [110] V. R. Basili and H. D. Rombach, "Support for comprehensive reuse," *Softw. Eng. J.*, vol. 6, no. 5, p. 303–316, 9 1991.
- [111] R. Land, D. Sundmark, F. Lüders, I. Krasteva, and A. Causevic, "Reuse with software components-a survey of industrial state of practice," in *Formal Foundations of Reuse and Domain Engineering*. Springer, 2009, p. 150–159.
- [112] M. L. Griss, "Software reuse architecture, process, and organization for business success," in *Proceedings of the Eighth Israeli Conference on Computer Systems and Software Engineering*, 1997, pp. 86–89.
- [113] B. W. Boehm, *Software Engineering Economics*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1981.
- [114] J. Fortune and R. Valerdi, "Considerations for successful reuse in systems engineering," 2008.
- [115] U. Shani and H. Broodney, "Reuse in model-based systems engineering." IEEE, 4 2015, pp. 77–83, [Online; accessed 2017-09-01]. [Online]. Available: <http://ieeexplore.ieee.org/document/7116732/>
- [116] W. B. Smith, "3.3.2 re-use libraries leveraging model-based systems engineering to greatly increase engineering productivity," *INCOSE International Symposium*, vol. 24, no. 1, pp. 298–312, 7 2014.
- [117] O. F. Bolanle, "Software reuse facilitated by the underlying requirement specification document: A knowledge-based approach," *American Journal of Software Engineering and Applications*, vol. 3, no. 3, p. 21, 2014, [Online; accessed 2015-05-07].
- [118] H. Mili, *Reuse based software engineering: techniques, organization and measurement*. New York: Wiley, 2002.

- [119] W. Tracz, *Confessions of a Used Program Salesman: Institutionalizing Software Reuse*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.
- [120] D. E. O’Leary, “Knowledge management: an empirical analysis of reuse and productivity,” *Journal of Decision Systems*, vol. 23, no. 3, pp. 249–265, 2014.
- [121] M. Morisio, M. Ezran, and C. Tully, “Success and failure factors in software reuse,” *IEEE Transactions on Software Engineering*, vol. 28, no. 4, pp. 340–357, 4 2002, [Online; accessed 2015-05-15].
- [122] K. C. Desouza, Y. Awazu, and A. Tiwana, “Four dynamics for bringing use back into software reuse,” *Commun. ACM*, vol. 49, no. 1, p. 96–100, 1 2006.
- [123] J. A. Mora Soto, “Marco metodológico y tecnológico para la gestión del conocimiento organizativo que dé soporte al despliegue de buenas prácticas de ingeniería del software,” Ph.D. dissertation, Universidad Carlos III de Madrid, Madrid, España, 9 2011.
- [124] M.-I. Sanchez-Segura, A. de Amescua Seco, F. Medina-Dominguez, G.-L. Dugarte-Peña, and J.-A. Mora-Soto, “Altus: A process-oriented, knowledge governance maturity model,” in *Knowledge, People, and Digital Transformation*, ser. Contributions to Management Science. Cham: Springer International Publishing, 2020, pp. 133–162.
- [125] H. de Jesus Ginja Antunes and P. G. Pinheiro, “Linking knowledge management, organizational learning and memory,” *Journal of Innovation and Knowledge*, vol. 5, no. 2, pp. 140 – 149, 2020.
- [126] M. Shahzad, Y. Qu, A. U. Zafar, S. U. Rehman, and T. Islam, “Exploring the influence of knowledge management process on corporate sustainable performance through green innovation,” *Journal of Knowledge Management*, 2020.
- [127] M. M. Migdadi, “Knowledge management, customer relationship management and innovation capabilities,” *Journal of Business and Industrial Marketing*, 2020.

- [128] A. Di Vaio, R. Palladino, A. Pezzi, and D. E. Kalisz, "The role of digital innovation in knowledge management systems: A systematic literature review," *Journal of Business Research*, vol. 123, pp. 220–231, 2021.
- [129] R. Davis, H. Shrobe, and P. Szolovits, "What is a knowledge representation?" *AI magazine*, vol. 14, no. 1, pp. 17–17, 1993.
- [130] J. M. A. Rodríguez, V. Moreno, and J. Llorens, "Formal ontologies and data shapes within the Software Engineering development lifecycle (TSE)," *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, vol. 2019-July, pp. 64–70, 2019.
- [131] T. Groza, S. Handschuh, T. Clark, S. Buckingham Shum, and A. de Waard, "A short survey of discourse representation models," 2009.
- [132] D. Paulius and Y. Sun, "A survey of knowledge representation in service robotics," *Robotics and Autonomous Systems*, vol. 118, pp. 13 – 30, 2019.
- [133] D.-H. Pham and A.-C. Le, "Learning multiple layers of knowledge representation for aspect based sentiment analysis," *Data and Knowledge Engineering*, vol. 114, pp. 26 – 39, 2018, special Issue on Knowledge and Systems Engineering (KSE 2016).
- [134] T. Gruber, "Toward principles for the desing of ontologies used for knowledge sharing," *Academic Press Limited*, no. 43, pp. 907–928, 1995.
- [135] P. Zangeneh and B. McCabe, "Ontology-based knowledge representation for industrial megaprojects analytics using linked data and the semantic web," *Advanced Engineering Informatics*, vol. 46, p. 101164, 2020.
- [136] M. Samwald, J. A. M. Gimenez, R. D. Boyce, R. R. Freimuth, K.-P. Adlassnig, and M. Dumontier, "Pharmacogenomic knowledge representation, reasoning and genome-based clinical decision support based on owl 2 dl ontologies," *BMC medical informatics and decision making*, vol. 15, no. 1, pp. 12–12, 2015.

- [137] S. A. Elnagdy, M. Qiu, and K. Gai, "Cyber incident classifications using ontology-based knowledge representation for cybersecurity insurance in financial industry," in *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CS-Cloud)*, 2016, pp. 301–306.
- [138] S. R. Fiorini, J. Bermejo-Alonso, P. Gonçalves, E. Pignaton de Freitas, A. Olivares Alarcos, J. I. Olszewska, E. Prestes, C. Schlenoff, S. V. Ragavan, S. Redfield, B. Spencer, and H. Li, "A suite of ontologies for robotics and automation [industrial activities]," *IEEE Robotics Automation Magazine*, vol. 24, no. 1, pp. 8–11, 2017.
- [139] L. Zhao, R. Ichise, S. Mita, and Y. Sasaki, "Core ontologies for safe autonomous driving," 10 2015.
- [140] A. G. Ryman, A. Le Hors, and S. Speicher, "Oslc resource shape: A language for defining constraints on linked data." *LDOW*, vol. 996, 2013.
- [141] J. E. L. Gayo, E. Prud'Hommeaux, I. Boneva, and D. Kontokostas, "Validating rdf data," *Synthesis Lectures on Semantic Web: Theory and Technology*, vol. 7, no. 1, pp. 1–328, 2017.
- [142] A. R. da Silva, "Tools exhibits," in *UML Modeling Languages and Applications*, N. Jardim Nunes, B. Selic, A. Rodrigues da Silva, and A. Toval Alvarez, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 281–291.
- [143] R. A. Mendieta Zuniga, J. L. d. l. Vara González, J. Llorens Morillo, and J. M. Álvarez Rodríguez, "Towards effective sysml model reuse." in *MODELSWARD*. Scitepress, 2017, pp. 536–541.
- [144] K. Robles, A. Fraga, J. Morato, and J. Llorens, "Towards an ontology-based retrieval of uml class diagrams," *Information and Software Technology*, vol. 54, no. 1, pp. 72–86, 2012.
- [145] E. Gallego, J. M. A. Rodríguez, and J. Llorens, "Reuse of physical system models by means of semantic knowledge representation: a case study applied to modelica," in

Proceedings of the 11th International Modelica Conference, Versailles, France, September 21-23, 2015, no. 118. Linköping University Electronic Press, 2015, pp. 747–757.

- [146] W. Guédria, Y. Naudet, and D. Chen, “Interoperability maturity models – survey and comparison –,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5333, no. November, pp. 273–282, 2008.
- [147] T. Ford, J. Colombi, S. Graham, and D. Jacques, *The Interoperability Score*, 2007.
- [148] European Commission, ISA Group, “CAMSS, common assessment method for standards and specifications,” JOINUP. <https://op.europa.eu/es/publication-detail/-/publication/b188707b-fe8e-11e8-a96d-01aa75ed71a1> (acceso: 16 de agosto de 2020).
- [149] European Commission, “Semantic Interoperability Maturity Assessment of a Public Service (SIMAPS),” JOINUP. <https://joinup.ec.europa.eu/collection/imaps-interoperability-maturity-assessment-public-service/solution/simaps/about> (acceso: 16 de agosto de 2020).
- [150] European Commission, “Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions European Interoperability Framework – Implementation Strategy COM/2017/0134 final,” EUR-Lex. <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=COM:2017:134:FIN> (acceso: 03 de septiembre de 2020).
- [151] A. J. Berre, B. Elvesæter, N. Figay, C. Guglielmina, S. G. Johnsen, D. Karlsen, T. Knothe, and S. Lippe, “The athena interoperability framework,” in *Enterprise Interoperability II*. London: Springer London, 01 2007, pp. 569–580.
- [152] H. Panetto, “Towards a classification framework for interoperability of enterprise applications,” *International Journal of Computer Integrated Manufacturing*, vol. 20, no. 8, pp. 727–740, 2007.

- [153] F. Vernadat, "Ueml: Towards a unified enterprise modelling language," *International journal of production research*, vol. 40, no. 17, pp. 4309–4321, 2002.
- [154] CORDIS, "Unified enterprise modelling language," CORDIS. <https://cordis.europa.eu/project/id/IST-2001-34229/es> (acceso: 03 de noviembre de 2020).
- [155] "Iec 62264-1 enterprise-control system integration–part 1: Models and terminology," International Organization for Standardization, Geneva, CH, Standard, 2003.
- [156] D. Chen, G. Doumeingts, and F. Vernadat, "Architectures for enterprise integration and interoperability: Past, present and future," *Computers in Industry*, vol. 59, no. 7, pp. 647–659, 3 2008.
- [157] R. Rezaei, T. K. Chiew, and S. P. Lee, "An interoperability model for ultra large scale systems," *Advances in Engineering Software*, vol. 67, pp. 22–46, 2014.
- [158] R. Reza, C. Thiam Kian, and L. Sai Peck, "A review on e-business interoperability frameworks," *Journal of Systems and Software*, vol. 93, pp. 199 – 216, 2014.
- [159] L. Guijarro, "Semantic interoperability in eGovernment initiatives," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 174–180, 2009.
- [160] S. D. Nagowah, H. Ben Sta, and B. A. Gobin-Rahimbux, "An overview of semantic interoperability ontologies and frameworks for iot," in *2018 Sixth International Conference on Enterprise Systems (ES)*. IEEE, 2018, pp. 82–89.
- [161] A. Gyrard, P. Patel, S. Datta, and M. Ali, "Semantic web meets internet of things (iot) and web of things (wot)," in *The 15th International Conference on Semantic Web*. ISWC, 8 2016.
- [162] OASIS TC, "Open services for lifecycle collaboration," OSLC <https://open-services.net/> (acceso: 03 de septiembre de 2020).
- [163] M. Liu, P. Lovett, N. Godwin, and E. Fletcher, "Improving the Intelligibility of Step Models for the Construction Industry," *Construction Information Technology*, vol. 1, pp. 1–14, 2000.

- [164] M. J. Pratt, "Introduction to iso 10303—the step standard for product data exchange," *Journal of Computing and Information Science in Engineering*, vol. 1, no. 1, pp. 102–103, 2001.
- [165] Adrian Murton, "A proposed standard for sharing modelling and simulation information in a collaborative systems engineering context," Mossec. http://www.mossec.org/c/document_library/get_file?p_l_id=349707&folderId=349884&name=DLFE-11356.pdf (acceso: 03 de septiembre de 2020).
- [166] M. M. Schmidt and R. Stark, "Model-based systems engineering (mbse) as computer-supported approach for cooperative systems development," in *Proceedings of 18th European Conference on Computer-Supported Cooperative Work*. European Society for Socially Embedded Technologies (EUSSET), 2020.
- [167] INCOSE, OMG, "Mbse wiki," Omgwiki. <http://www.omgwiki.org/MBSE/doku.php> (acceso: 03 de septiembre de 2020).
- [168] W. Lahlayl, N. Berbiche, F. Guerouate, and M. Sbihi, "Solving the interoperability problem between UML modeling tools: Modelio and ArgoUML," *International Journal of Applied Engineering Research*, vol. 12, no. 19, pp. 8632–8641, 2017.
- [169] OASIS, "Oslc requirements management specification version 2.0," Open Services for Lifecycle Collaboration Lifecycle integration inspired by the web <https://archive.open-services.net/wiki/requirements-management/OSLC-Requirements-Management-Specification-version-2.0/index.html> (acceso: 03 de septiembre de 2020).
- [170] OASIS, "Oslc asset management 2.0 specification," Open Services for Lifecycle Collaboration Lifecycle integration inspired by the web <https://archive.open-services.net/wiki/asset-management/OSLC-Asset-Management-2.0-Specification/index.html> (acceso: 03 de septiembre de 2020).
- [171] OASIS, "Oslc quality management version 2.1. part 1: Specification," Open Services for Lifecycle Collaboration Lifecycle integration inspired by the web <https://docs>.

- oasis-open-projects.org/oslc-op/qm/v2.1/ps01/quality-management-spec.html (acceso: 03 de septiembre de 2020).
- [172] B. K. Aichernig, K. Hörmaier, F. Lorber, D. Nickovic, R. Schlick, D. Simoneau, and S. Tiran, "Integration of requirements engineering and test-case generation via oslc," in *2014 14th International Conference on Quality Software*, 2014, pp. 117–126.
- [173] OASIS, "Oslc change management version 3.0. part 1: Specification," Open Services for Lifecycle Collaboration Lifecycle integration inspired by the web <https://docs.oasis-open-projects.org/oslc-op/cm/v3.0/ps01/change-mgt-spec.html> (acceso: 03 de septiembre de 2020).
- [174] OASIS, "Oslc performance monitoring specification version 2.0," Open Services for Lifecycle Collaboration Lifecycle integration inspired by the web <https://archive.open-services.net/wiki/performance-monitoring/OSLC-Performance-Monitoring-Specification-Version-2.0/index.html> (acceso: 03 de septiembre de 2020).
- [175] A. Reichwein and R. Burkhart, "Integration of mbse artifacts using oslc," *INSIGHT*, vol. 18, pp. 42–44, 08 2015.
- [176] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.
- [177] S. K. Mohamed, A. Nounu, and V. Nováček, "Biological applications of knowledge graph embedding models," *Briefings in Bioinformatics*, 2020.
- [178] Y. Dai, S. Wang, N. N. Xiong, and W. Guo, "A survey on knowledge graph embedding: Approaches, applications and benchmarks," *Electronics*, vol. 9, no. 5, p. 750, 2020.
- [179] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Product knowledge graph

- embedding for e-commerce,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 672–680.
- [180] Y. Shen, N. Ding, H.-T. Zheng, Y. Li, and M. Yang, “Modeling relation paths for knowledge graph completion,” *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [181] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation learning of knowledge graphs with entity descriptions,” in *Thirtieth AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.
- [182] S. Zhang, Y. Tay, L. Yao, and Q. Liu, “Quaternion knowledge graph embeddings,” in *Advances in Neural Information Processing Systems*, 2019, pp. 2735–2745.
- [183] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, and H. Hajishirzi, “Text generation from knowledge graphs with graph transformers,” *arXiv preprint arXiv:1904.02342*, 2019.
- [184] SIEMENS, “Teamcenter,” SIEMENS. <https://www.plm.automation.siemens.com/global/es/products/teamcenter/> (acceso: 03 de noviembre de 2020).
- [185] TRC and KR Group, “System representation language (srl),” trc-research <https://github.com/trc-research/oslc-km/tree/master/trunk/apps/dotnet/OSLC-KM-SRL> (acceso: 03 de septiembre de 2020).
- [186] S. Huang, V. Gohel, and S. Hsu, “Towards interoperability of uml tools for exchanging high-fidelity diagrams,” in *Proceedings of the 25th Annual ACM International Conference on Design of Communication*, ser. SIGDOC '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 134–141.
- [187] The Reuse Company, “Systems engineering suite (se suite),” The Reuse Company. <https://www.reusecompany.com/systems-engineering-suite> (acceso: 03 de mayo de 2020).

- [188] University of Manchester, “The owl api,” The OWL API <http://owlapi.sourceforge.net/> (acceso: 03 de septiembre de 2020).
- [189] J. Frijters, “Ikvm.net home page,” IKVM. <https://www.ikvm.net/> (acceso: 03 de mayo de 2020).
- [190] The Reuse Company, “Km knowledge manager,” The Reuse Company. <https://www.reusecompany.com/km-knowledge-manager> (acceso: 03 de mayo de 2020).
- [191] ISO, “Iso/iec/ieee 15288: 2015, systems and software engineering–system life cycle processes,” International Organization for Standardization, Geneva, CH, Standard, 2015.
- [192] The Reuse Company, “Traceability studio,” The Reuse Company. <https://www.reusecompany.com/traceability-studio> (acceso: 03 de mayo de 2020).
- [193] The Reuse Company, “Rqa-quality studio,” The Reuse Company. <https://www.reusecompany.com/rqa-quality-studio> (acceso: 03 de mayo de 2020).
- [194] N. Juristo and A. M. Moreno, *Basics of software engineering experimentation*. Springer Science & Business Media, 2013.
- [195] S. Friedenthal, A. Moore, and R. Steiner, *A practical guide to SysML: the systems modeling language*. San Francisco: Morgan Kaufmann, 2014.
- [196] OMG, “The Official OMG SysML site,” 2016, OMG. <http://www.omgsysml.org/> (acceso: 03 de septiembre de 2020).
- [197] E. Integrated, R. Radio, and E. Network, “System Requirements Specification Version 15.4.0,” no. March, 2014, [En línea]. Disponible en: <https://uic.org/IMG/pdf/p0028d004.3r0.5-15.4.0.pdf>. Acceso: 05 de enero de 2021.
- [198] J. C. Mankins, “Technology readiness levels,” *White Paper, April*, vol. 6, p. 1995, 1995.
- [199] J. C. Mankins, “Technology readiness assessments: A retrospective,” *Acta Astronautica*, vol. 65, no. 9-10, pp. 1216–1223, 2009.

- [200] NASA, "Nasa systems engineering processes and requirements," 2013, [En línea]. Disponible en: https://nodis3.gsfc.nasa.gov/npg_img/N_PR_7123_001C_/N_PR_7123_001C_.pdf. Acceso: 07 de enero de 2021.
- [201] J. Straub, "In search of technology readiness level (trl) 10," *Aerospace Science and Technology*, vol. 46, pp. 312 – 320, 2015, [En línea]. Disponible en: <http://www.sciencedirect.com/science/article/pii/S127096381500214X>. Acceso: 07 de enero de 2021.
- [202] P. M. Duvall, S. Matyas, and A. Glover, *Continuous integration: improving software quality and reducing risk*. Pearson Education, 2007.