

SERIE INGENIERÍA DE SISTEMAS

Cuadernos de Isdefe



2

INGENIERÍA DE SISTEMAS

aplicada a los Sistemas de Sistemas



La Ingeniería de Sistemas aplicada a los Sistemas de Sistemas

Volumen 2

Serie Cuadernos de Ingeniería de Sistemas

Cuadernos de Isdefe



AUTORES

Dra. Judith Dahmann / David Sánchez García

Dr. Michael Yokell / Dr. Alejandro Salado / Adolfo Sánchez Domínguez

Dr. James Moreland Jr / TCol. Víctor M. Sobrino García

Matthew Gagliardi / Matthew C. Hause / Dr. James N. Martin

Dr. Polinpapilinho F. Katina / Dr. Charles B. Keating / César Heras Menor de Gaspar / Víctor Ramos del Pozo

Dr. Michael Sievers / Pablo Marticorena San José

Mark Phillips / Dr. Keith Joiner / Aurelio Fernández Sáez / Manuel Fernández Astaburuaga

Tom McDermott / Miguel Ángel Coll

José Luis de Rosario / Dr. Paul Grogan / Dr. Alessandro Golkar /

Dr. Amro Farid / Dr. Young-Jun Son / Dr. Nil Egin / Dr. John Little



Título original: La Ingeniería de Sistemas aplicada a los Sistemas de Sistemas

© Isdefe

C/ Beatriz de Bobadilla, 3 -28040 Madrid

www.isdefe.es

Primera edición: Septiembre 2025

ISBN: 978-84-09-62257-3

No comercial

Depósito legal: M-14808-2024

Editor: Isdefe

Coordinador: Juan Manuel García Montaña

Coordinador técnico: Dr. Alejandro Salado

Equipo de edición y revisión de estilo: Juan Manuel García Montaña, Dr. Alejandro Salado

Diseño y maquetación: Iliana Aguilar Jiménez

Grabaciones y edición vídeos: Favorit Comunicación

Impreso por Byprint Madrid

Impreso en España – *Printed in Spain*

Las opiniones contenidas en este libro son de exclusiva responsabilidad de los autores firmantes. No pretenden reflejar las opiniones ni el punto de vista de Isdefe como empresa.

El equipo de edición ha hecho todos los esfuerzos posibles para obtener los permisos pertinentes de todo el material reproducido en este libro. Si se hubiera producido alguna omisión, pedimos que nos hagan llegar por escrito la solicitud correspondiente para subsanar el error.

Todos los derechos reservados. Queda prohibida la reproducción total o parcial sin autorización del editor ©.

Queremos expresar nuestro agradecimiento a las siguientes personas y organizaciones por haber hecho posible esta edición del segundo cuaderno de Ingeniería de Sistemas en el siglo XXI:

Al GD (R) D. Jesús Gómez Pardo, prologuista de la obra.

A nuestro coordinador técnico Dr. D. Alejandro Salado por su dirección y capacidad constantes para resolver las dificultades encontradas.

A Dña. Rocío Manjón Pérez, colaboradora e impulsora de estos cuadernos.

Al personal de la Dirección de Desarrollo Corporativo y Estrategia por su apoyo imprescindible y constante (en especial a D. Lluís Vinagre, D. Ildefonso Vera, Dña. Natalia Garcillán y Dña. Carmen Ekame).

"El mayor peligro en tiempos de turbulencia no es la turbulencia; es actuar con la lógica de ayer."

Peter Drucker

PRÓLOGO

Vivimos en un mundo convulso. La **inestabilidad e incertidumbre global** de nuestro entorno vital desde la caída del muro de Berlín se ven catalizadas por la **revolución tecnológica** en la que estamos inmersos. Tecnologías emergentes que introducen cambios profundos en nuestra forma de pensar y actuar, y que están acelerando la transición de nuestra sociedad hacia la era digital.

Nos enfrentamos a futuros desconocidos, a enormes desafíos para la seguridad y retos tecnológicos, ante los cuales, sólo podemos actuar creando **estructuras ágiles**, en permanente adaptación a la rápida evolución del entorno, y dotándonos de **organizaciones más flexibles y resilientes**, que nos permitan sobrevivir y operar en un entorno cada vez más volátil y complejo. Nuestros Ejércitos y Armada, parte integrante de la sociedad, están sumidos en un proceso de **transformación digital** que modifica el arte de la guerra. Un **campo de batalla digitalizado**, hiperconectado, en el que las operaciones se llevan a cabo en red por combatientes bien pertrechados que se constituyen en nodos de esa red de combate y en el que se ha generalizado el uso de vehículos no tripulados, sistemas autónomos, drones, robots armados, actuando de manera individual o colaborativa, son ya una realidad que está modificando sustancialmente los actuales conceptos operativos.



Un claro ejemplo es el conflicto en Ucrania¹. La invasión ilegal de Ucrania por la Federación Rusa nos devuelve, aparentemente, a un conflicto convencional de alta intensidad, pero con algunas salvedades: ambos bandos están haciendo uso extensivo de ciertas **tecnologías emergentes y disruptivas (EDT)**, características de un conflicto moderno; existe, además, un claro **enfrentamiento en la zona gris** entre Rusia y Occidente; y, todo esto, con el telón de fondo de la amenaza nuclear. Las operaciones se están llevando a cabo en el **multidominio** que integra los ámbitos de operación físicos, tierra, mar y aire, incluyendo el espacio ultraterrestre, y los dominios no físicos, el ciberespacio y el espacio cognitivo, que establece la doctrina española. Por otro lado, se han puesto de relieve algunos cambios fundamentales en la batalla asociados a la digitalización, destacando sobremanera el **uso masivo de drones**.

De este terrible conflicto, de elevada atrición, podemos extraer, centrándonos en **capacidades militares**, las siguientes lecciones aprendidas: que las capacidades de **inteligencia, vigilancia, reconocimiento y adquisición de objetivos** son claves para la superioridad en las operaciones; que en el campo de batalla resulta crítico disponer de **fuegos precisos y de largo alcance** para batir objetivos con precisión, minimizando daños colaterales; que es necesaria una gran **potencia de fuego** para saturar al adversario; que la digitalización del campo de batalla es ya una realidad, habiendo quedado patente que para que las operaciones en red sean posibles, debe garantizarse la **interconexión de los sensores y sistemas de combate con los centros de gestión y elementos productores de fuegos o efectos**; y, por último, que para garantizar dicha interconexión es imprescindible disponer de **sistemas de mando y control** robustos, redundantes y de alta movilidad, así como de **capacidades de guerra electrónica** para operar en el espectro electromagnético con la necesaria superioridad, de manera que pueda garantizarse la operatividad en entornos degradados y la libertad de acción en los cinco dominios de las operaciones.

Ucrania también ha puesto de relieve el cómo una situación de combate espolea el ingenio y la capacidad de innovar y cómo la amenaza cambia y evoluciona de manera vertiginosa. Por eso es importante que, para dotar a nuestras FAS de estas nuevas capacidades militares, la **industria de defensa** sea capaz de fabricar, en tiempo breve, los sistemas de armas adecuados para enfrentar las cambiantes amenazas. Para ello deberá implementar **metodologías ágiles de desarrollo**, utilizando las posibilidades de la digitalización, en el desarrollo y maduración de las **EDT** que están irrumpiendo en el campo de batalla.

En este sentido, nuestras FAS deberán disponer de potentes **sistemas de guerra electrónica** preparados para la guerra de navegación (**NAVWAR**) de manera que pueda asegurarse la superioridad de la información PNT (posición, navegación y tiempo), protegiendo los sistemas de navegación propios y degradando la información PNT del adversario. El desarrollo de **tecnologías antidron** pasa a ser absolutamente prioritario. Para garantizar la seguridad de cualquier operación debe disponerse de un catálogo amplio de tecnologías

1. "Conclusiones iniciales de la guerra en Ucrania". Centro Conjunto de Desarrollo de Conceptos. Estado Mayor Conjunto, marzo 2023.

para la localización, identificación y seguimiento de la amenaza, así como para su neutralización, bien con medios *soft kill* (*jamming* o *spoofing*) o bien con medios *hardkill* – destrucción efectores cinéticos (proyectiles, cohetes, o sistemas de protección activa), redes, pulso electromagnético, arma láser, etc. Asimismo, resulta fundamental el desarrollo tecnológico en los campos de la **robótica y los vehículos autónomos** para automatizar los sistemas heredados (*drive-by-wire*), desarrollar interfaces avanzados conductor-vehículo, tecnologías para la interacción UGV-UA, etc. Otras tecnologías fundamentales a desarrollar y disponer serán las asociadas a la **industria 4.0** (IA, sistemas ciber físicos, *machine learning*, gemelos digitales, sistemas *cloud*, *Big Data*, ...) que permitirán completar la transformación de las estructuras logísticas de los ejércitos, evolucionando de una logística reactiva-preventiva, a una logística predictiva. Finalmente, el empleo de la IA y el Big Data, será prioritario para el mantenimiento predictivo de plataformas, el análisis automático e inteligente de grandes volúmenes de datos procedentes de sensores de sistemas de armas y el análisis inteligente de fuentes de información en apoyo a la decisión. Así, la IA resultará imprescindible, entre otros, para la simulación completa del entorno de batalla, los sistemas de gestión integral del combate C-UAS, la fusión de sensores, el posicionamiento preciso en entornos complejos, la navegación autónoma en entornos no estructurados, el desarrollo de algoritmos de automatización, la planificación de itinerarios, etc.

En cuanto a la **financiación** de estas EDT podemos decir que la incertidumbre e inestabilidad geopolítica, junto al debilitamiento del vínculo trasatlántico, están forzando el crecimiento de las inversiones en Defensa con el fin de alcanzar la autonomía estratégica europea y cumplir con los compromisos OTAN. En este sentido, el gobierno de España ha publicado recientemente el **Plan Industrial y Tecnológico para la Seguridad y la Defensa**² con el objetivo de garantizar la seguridad de España y consolidar a España como un miembro central y fiable de la Unión Europea y de la OTAN. El plan también pretende impulsar una nueva ola de innovación y de reindustrialización de empresas en torno a las tecnologías de doble uso. Las inversiones industriales se van a centrar en elaborar, fabricar y adquirir nuevas capacidades de telecomunicaciones y de ciberseguridad, así como en la fabricación y compra de nuevos equipos de defensa.

Es evidente que somos conscientes de los desafíos de seguridad y de las capacidades militares necesarias para enfrentarlos, que conocemos los retos tecnológicos y las capacidades industriales esenciales para el desarrollo de las EDT claves, y que disponemos de las herramientas económico financieras para obtener los sistemas de armas técnicamente más avanzados que proporcionen una ventaja operativa clara a nuestros combatientes. Y es en este proceso de obtención en el que la ingeniería de sistemas juega un papel fundamental.

La ingeniería de sistemas (IS) tradicional se ha aplicado en Defensa desde principios de los años 90. En el año 1992 se creó la Subdirección de Sistemas en el Mando de Apoyo Logístico del Ejército. El entonces coronel Torrón introdujo en el ámbito militar el

². <https://www.lamoncloa.gob.es/consejodeministros/resumenes/Documents/2025/230425-plan-industrial-y-tecnologico-para-la-seguridad-y-la-defensa.pdf>

vocabulario de la IS evidenciando la escasa cultura que teníamos, en aquel momento, en los ejércitos sobre el importante papel que la IS juega en el ciclo de vida de los sistemas de armas. Isdefe respondió al reto de cubrir esta laguna elaborando la primera serie de **publicaciones de ingeniería de sistemas de Isdefe**, iniciada en 1995, los famosos libros azules escritos por grandes especialistas en la materia (Blanchard, Sarabia, Aracil, ...) En 2002, ya con cierto nivel de concienciación, el MALE comenzó a trabajar en el **programa PRISMA** (Programa de Reingeniería de Sistemas del MALE), siendo el jefe de programa el entonces coronel Orts. En 2003 se procedió a su lanzamiento oficial con la contratación de una asistencia técnica a Isdefe para el periodo 2003-2005. Durante ese periodo se realizó un esfuerzo increíble, tanto por la empresa como por el MALE, para lograr su implantación en 2005. PRISMA nació con un doble objetivo:

- a) Adoptar la metodología OTAN del ciclo de vida de los sistemas.
- b) Proporcionar una metodología de trabajo común, en todas las secciones del MALE en el proceso de adquisición y mantenimiento de los sistemas de armas.

Con un vocabulario único, bien definido, PRISMA servía de guía a los jefes de programa y directores técnicos en sus relaciones con las empresas adjudicatarias y con los servicios de calidad oficiales. A su vez, sirvió para “educar” a las empresas en la IS. En definitiva, un conjunto de guías para la adecuada gestión del ciclo de vida de los sistemas de armas.

Sin embargo, cuando analizamos la realidad actual del campo de batalla y su evolución -operaciones en red, nube de combate, ciberdefensa, sistemas antidron, sistemas autónomos colaborativos, IA, logística predictiva, comunicaciones cuánticas y fotónicas, etc.- se observa que el concepto tradicional de sistema de armas está cambiando. Los sistemas modernos son muy complejos. Están constituidos por sistemas independientes interconectados entre sí (radares, satélites, vehículos no tripulados, sistemas de comunicaciones, etc.) e integrados en una arquitectura coordinada que actúa de forma más eficiente en el cumplimiento de la misión. A estos sistemas complejos, caracterizados por una gobernanza distribuida, se les da el nombre de **sistemas de sistemas (SdS)**, también conocidos, dependiendo del dominio de aplicación, por supersistemas.

En la primera monografía de este grupo de cuadernos “Introducción a la ingeniería de sistemas en el Siglo XXI” ya se describieron en detalle los nuevos tipos de sistemas -sistemas ciberfísicos, sistemas basados en el aprendizaje o sistemas con gobernanza distribuida (sistema de sistemas)-, para los que los métodos tradicionales de la ingeniería de sistemas pueden resultar ineficaces, razón por la que también se analizó la necesidad de evolucionar y adaptar la metodologías tradicionales de la ingeniería de sistemas a los SdS. De esta manera, transitamos de la ingeniería de sistemas a la ingeniería de sistemas de sistemas, caso que nos ocupa en el presente cuaderno, y cuyo magnífico contenido se presentará en el prefacio.

De lo dicho hasta ahora, vemos que las operaciones actuales en el multidominio no se conciben sin los SdS. Desde un punto de vista militar, los SdS son relevantes para:

- a) Garantizar la **interoperabilidad**, permitiendo que diferentes sistemas, operando en distintos dominios, trabajen en conjunto compartiendo información en tiempo real y coordinando acciones.
- b) Mejorar la **toma de decisiones** estratégicas, integrando múltiples fuentes de información que proporcionan una visión global del campo de operaciones y una mayor conciencia situacional.
- c) Asegurar la **resiliencia y adaptabilidad** puesto que, si un sistema falla, otro puede compensar su función, manteniendo la operación en marcha;
- d) Favorecer la **escalabilidad y evolución tecnológica**, mediante la incorporación de nuevas tecnologías, sin tener que rediseñar todo el sistema desde cero, lo cual es esencial en un entorno tecnológico y de seguridad (amenazas) en constante cambio.
- e) Facilitar la **coordinación multinacional**, permitiendo integrar sistemas de diferentes países, con protocolos y tecnologías diversas, en operaciones conjuntas con otros aliados.

Dada la complejidad de los SdS y que las metodologías de la Ingeniería de los SdS están aún en fase de desarrollo, resulta imprescindible disponer del **talento adecuado**. Así, el personal de los Cuerpos de Ingenieros de los Ejércitos, en los ámbitos de la Secretaría de Estado de Defensa y de los Mandos de apoyo Logísticos, deben formarse, de manera continua, en esta disciplina. La monografía que el lector tiene en sus manos, contribuye a paliar esta necesidad. Su excelente contenido, elaborado por verdaderos expertos internacionales y de Isdefe en la materia, constituye un magnífico referente para todos aquellos con responsabilidades en el ciclo de vida de los SdS y para la comunidad de los ingenieros de sistemas en general.

Para terminar, quiero agradecer a Isdefe, al coordinador de la serie, TG García Montaña, y a la jefa del Área de Innovación, Dña. Belina Misiego, la oportunidad que me han brindado de prologar la presente monografía. Un honor y un privilegio haber participado en la redacción de esta segunda entrega de la serie “Cuadernos de la Ingeniería de Sistemas. Cuadernos de Isdefe.”

Jesús Carlos Gómez Pardo.
General de División (R).
Dr. Ingeniero de Armamento.



PREFACIO

Esta monografía presenta una visión general estructurada y exhaustiva de la ingeniería de Sistemas de Sistemas (SdS), ofreciendo una instantánea de las prácticas actuales y de disciplinas emergentes en este dominio complejo y cada vez más crítico. Es la segunda monografía de la serie “azul” sobre ingeniería de sistemas publicada por Isdefe y continúa nuestro esfuerzo por proporcionar recursos técnicamente rigurosos y accesibles para los profesionales del ámbito gubernamental e industrial.

La monografía ha sido concebida con un público específico en mente: profesionales de los sectores de defensa, seguridad, espacio, energía y transporte de España y Europa, involucrados en la adquisición, desarrollo, integración u operación de SdS. Ya sea en calidad de proveedores o clientes, estos profesionales enfrentan los retos multifacéticos de trabajar con sistemas que son gestionados de forma independiente, distribuidos, en evolución y, con frecuencia, coordinados de manera débil. El contenido de este libro busca apoyar su comprensión y toma de decisiones, con un firme anclaje en la práctica actual.

Cada capítulo ha sido elaborado mediante una colaboración única entre expertos internacionales y profesionales de Isdefe, asegurando tanto rigor técnico como relevancia práctica. Aunque los capítulos pueden leerse de manera independiente, el libro ha sido diseñado como un volumen cohesivo, en el que cada contribución forma parte de una narrativa más amplia sobre la ingeniería de SdS. Cada capítulo actúa como una introducción a un concepto o actividad clave dentro de la ingeniería de SdS, sirviendo como punto de partida para una exploración más profunda en la literatura técnica.



El libro comienza con una introducción a los SdS (Capítulo 1), seguida de una discusión sobre cómo la ingeniería de sistemas tradicional debe evolucionar para afrontar los desafíos de los SdS (Capítulo 2), utilizando la norma ISO 21840 como guía. El Capítulo 3 introduce la ingeniería de misión como una disciplina que prioriza los resultados de misión por encima de las capacidades de plataforma. El Capítulo 4 presenta un ejemplo concreto de un hilo de misión, ofreciendo al lector una aplicación práctica de los conceptos tratados previamente.

La gobernanza, la planificación, la integración, así como la verificación y validación, procesos clave que requieren un enfoque distinto cuando se aplican a SdS, son abordados en los Capítulos 5 al 7. El Capítulo 8 cuestiona la noción tradicional del ciclo de vida del sistema, proponiendo en su lugar una perspectiva de evolución continua de los SdS. El capítulo final (Capítulo 9) presenta métodos avanzados de modelado, simulación y análisis, incluyendo modelado basado en agentes, enfoques federados y teoría de grafos heterofuncionales, entre otros, para abordar las complejas interdependencias de los SdS.

A lo largo de la obra, hemos procurado evitar visiones especulativas del futuro, opiniones subjetivas sobre el estado del campo o contenido promocional sin respaldo en la práctica. El resultado es una monografía deliberadamente sobria, basada en las capacidades actuales y en la experiencia real. No obstante, no ignora el progreso: allí donde surgen desarrollos prometedores, estos se presentan de manera mesurada y objetiva.

Es justo reconocer que el grado de madurez de las organizaciones en ingeniería de SdS varía ampliamente. Algunas pueden estar aún desarrollando capacidades en áreas ya consolidadas dentro de la ingeniería de sistemas tradicional. Esta monografía no debe leerse como un ideal inalcanzable, sino como evidencia de que el progreso estructurado y metódico es posible y que ya se está produciendo en ciertos entornos.

Confiamos en que esta monografía sirva tanto de referencia como de fuente de inspiración para quienes trabajan en el avance de las capacidades en SdS en sus propios contextos.

En nombre de los autores, del equipo de gestión del proyecto y de Isdefe, espero que la lectura les resulte educativa, amena y útil.

Dr. Alejandro Salado
The University of Arizona



LA INGENIERÍA DE SISTEMAS APLICADA A LOS SISTEMAS DE SISTEMAS

PRÓLOGO	9
PREFACIO	15
1. SISTEMAS DE SISTEMAS: UNA INTRODUCCIÓN	21
1.1. <i>Una historia de los Sistemas de Sistemas hasta nuestros días</i>	22
1.2. <i>Características de los Sistemas de Sistemas</i>	25
1.3. <i>Impacto de las Tecnologías Avanzadas en los Sistemas de Sistemas</i>	28
1.4. <i>Conclusión</i>	30
2. DE LA INGENIERÍA DE SISTEMAS A LA INGENIERÍA DE SISTEMAS DE SISTEMAS: UNA VISIÓN GENERAL	37
2.1. <i>Introducción</i>	38
2.2. <i>Consideraciones clave de la SoSE</i>	39
2.3. <i>Procesos de acuerdo</i>	40
2.4. <i>Procesos facilitadores de proyectos organizacionales</i>	41
2.5. <i>Procesos de gestión técnica</i>	42
2.6. <i>Procesos técnicos</i>	44
2.7. <i>Conclusiones</i>	48
3. INGENIERÍA DE MISIÓN	57
3.1. <i>Introducción</i>	58
3.2. <i>Razones que impulsan el nuevo enfoque y la disciplina</i>	60
3.3. <i>Propuesta de valor</i>	63
3.4. <i>Aspectos característicos de la ingeniería de misión</i>	63
3.5. <i>Conclusiones</i>	69
4. MODELADO DE LA ARQUITECTURA DE LA MISIÓN	75
4.1. <i>Introducción</i>	76
4.2. <i>Ejemplo de Misión</i>	77
4.3. <i>Desarrollo de una arquitectura operativa</i>	84
4.4. <i>Desarrollo de la arquitectura de recursos</i>	91
4.5. <i>Otras consideraciones</i>	93
4.6. <i>Conclusiones</i>	94
5. DISEÑO E IMPLEMENTACIÓN DE LA GOBERNANZA DE SISTEMAS DE SISTEMAS	101
5.1. <i>Introducción</i>	102
5.2. <i>Gobernanza en el contexto de un Sistema de Sistemas</i>	103
5.3. <i>Gobernanza de Sistemas de Sistemas</i>	104
5.4. <i>Aplicación de la gobernanza para el diseño e implementación de Sistemas de Sistemas</i>	109
5.5. <i>Prácticas de referencia</i>	113
5.6. <i>Conclusiones</i>	115

6. PLANIFICACIÓN E INTEGRACIÓN DE SISTEMAS DE SISTEMAS	123
6.1. <i>Introducción</i>	124
6.2. <i>Consideraciones de planificación de Sistemas de Sistemas</i>	124
6.3. <i>Planificación de la Interoperabilidad</i>	128
6.4. <i>Arquitectura y planificación de la integración</i>	129
6.5. <i>Conclusión</i>	132
7. PRUEBAS Y EVALUACIÓN DE SISTEMAS DE SISTEMAS	137
7.1. <i>Introducción</i>	138
7.2. <i>Conceptos que afectan a las pruebas de un SdS</i>	138
7.3. <i>Gestión de riesgos en pruebas de SdS</i>	139
7.4. <i>Consideraciones para el apoyo a las pruebas y evaluación de SdS</i>	140
7.5. <i>Metodologías de pruebas</i>	142
7.6. <i>Principios rectores de T&E en SdS</i>	146
7.7. <i>Conclusiones</i>	149
8. DEL CICLO DE VIDA A LA EVOLUCIÓN DE LOS SISTEMAS DE SISTEMAS	159
8.1. <i>Introducción: Acerca de la evolución de los Sistemas de Sistemas</i>	160
8.2. <i>Características de la evolución en Sistemas de Sistemas</i>	160
8.3. <i>Desafíos para liderar la evolución de los Sistemas de Sistemas</i>	168
8.4. <i>Métodos para guiar la evolución de los Sistemas de Sistemas</i>	170
8.5. <i>Competencias de liderazgo para la gestión de los Sistemas de Sistemas</i>	178
8.6. <i>Conclusión</i>	181
9. NUEVOS MÉTODOS DE ANÁLISIS, MODELADO Y SIMULACIÓN PARA SDS	187
9.1. <i>Introducción</i>	188
9.2. <i>Coordinación de Sistemas</i>	188
9.3. <i>Federación de sistemas</i>	191
9.4. <i>Introducción conceptual a la teoría de grafos hetero funcionales</i>	195
9.5. <i>Modelado y simulación federados</i>	200
9.6. <i>Modelado basado en agentes para Sistemas de Sistemas</i>	202
9.7. <i>Una aproximación de Sistemas de Sistemas al modelado</i>	207
9.8. <i>Conclusiones</i>	215



Sistemas de Sistemas: Una introducción

Dra. Judith Dahmann, *The MITRE Corporation* (jdahmann@mitre.org)
David Sánchez García, *Isdefe* (dsanchez@isdefe.es)

Resumen

Los sistemas de sistemas (SdS) son cada vez más frecuentes y sus características fundamentales plantean retos para la aplicación de la ingeniería de sistemas. Este capítulo presenta una introducción a los sistemas de sistemas (SdS) – su historia, sus características distintivas, ejemplos, el impacto de la tecnología avanzada en los SdS y el futuro de los SdS – como contexto para los capítulos posteriores que tratan de la ingeniería de SdS (ISdS).

Palabras clave

Sistemas de Sistemas, Ingeniería de Sistemas



1. UNA HISTORIA DE LOS SISTEMAS DE SISTEMAS HASTA NUESTROS DÍAS

El concepto de sistemas de sistemas (SdS) existe desde mucho antes de que se aceptara la terminología de sistemas de sistemas. Gorod et al [1] proporcionan un recorrido detallado de la bibliografía sobre los SdS, como se muestra en la figura 1, que incluye anotaciones para reflejar las características clave en la historia de los SdS.

Las primeras referencias a los SdS que se suelen citar [1, 2] incluyen el artículo de Boulding sobre los constructos teóricos generales [3], en el que describía el concepto de SdS como "la disposición de sistemas y constructos teóricos en una jerarquía de complejidad", considerando el SdS como un "sistema abierto" que puede verse afectado por sucesos externos, señalando, sin embargo, que esta definición no es diferente de las que han definido a los sistemas tradicionales [4]. Al principio, otros caracterizaron la planificación urbana, las estructuras de los sistemas científicos y los sistemas biológicos como SdS [5, 6,7].

A pesar de la creciente atención prestada a los SdS, ya en 2015 existía controversia sobre la definición de los SdS [2]:

Como era de esperar en un campo emergente, aún no existe una definición precisa y ampliamente aceptada de los SdS a la que se ajuste la mayor parte de la bibliografía, lo que dificulta delimitar el campo con precisión. La bibliografía es variada y presenta muchos intentos de definir y caracterizar los SdS. Varios trabajos han tratado de lograr cierta convergencia [8, 9, 10, 11, 12, 13].

A finales de los noventa, Mark Maier aportó una perspectiva fundamental que ha establecido las bases del pensamiento sobre los SdS hasta nuestros días [14]. Esto se analiza en la siguiente sección sobre las características de los SdS.

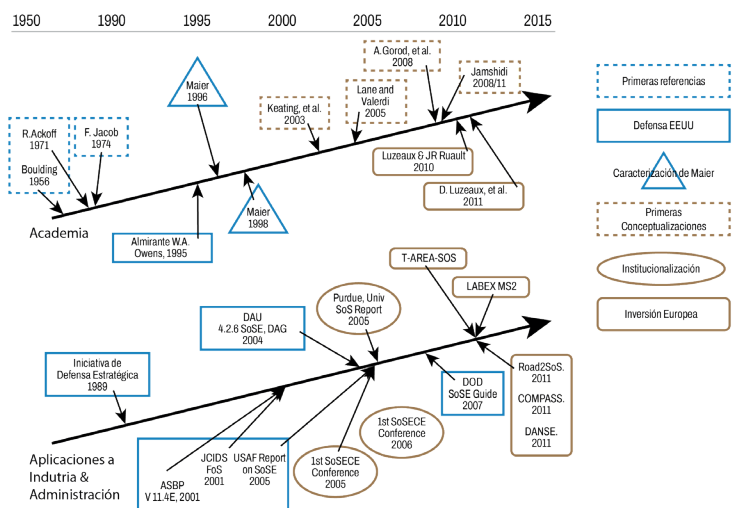


Figura 1: Historia Moderna de los Sistemas de Sistemas y la Ingeniería de Sistemas de Sistemas [1] con superposición.

Los primeros ejemplos de SdS desde el punto de vista de la ingeniería proceden del sector de la defensa de EEUU, con la Iniciativa de Defensa Estratégica, que inició el proceso de considerar las capacidades de defensa como SdS en EEUU. El Almirante W.A. Owens, de la US Navy, introdujo los SdS en el ámbito militar [15] en "El Emergente Sistema de Sistemas de EEUU".

Los elementos que confieren a las fuerzas militares su capacidad de combate están cambiando y estos cambios apuntan hacia un salto cualitativo en nuestra capacidad para utilizar la fuerza militar con eficacia.

Probablemente, en relación con la forma en la que planificamos, programamos y presupuestamos estas cosas, somos más competentes en visualizar los árboles individuales que el vasto bosque de capacidades militares (el sistema de sistemas) que los sistemas por separado están constituyendo para nuestras fuerzas de combate.

El sistema de sistemas depende en última instancia de las contribuciones bien orquestadas de todos los ejércitos. Esto asume una apreciación común y una adhesión a lo que estamos construyendo. Y lo que es más importante, requiere una doctrina estratégica y operacional conjunta que permita organizar, planificar y llevar a cabo las operaciones militares.

Las Fuerzas Aéreas [16] y el Ejército [17] de EE.UU. empezaron a tener en cuenta estas perspectivas y la publicación de la Guía de Ingeniería de SdS del Departamento de Defensa de EEUU [18] adoptó una perspectiva de ingeniería de SdS (ISdS) para toda la defensa.

A principios de la década de 2000, las instituciones empezaron a reconocer los sistemas de sistemas. El primer congreso anual sobre Ingeniería de SdS se celebró en Los Ángeles en 2006 y estos congresos continúan hasta hoy. El Consejo Internacional de Ingeniería de Sistemas (INCOSE) estableció un grupo de trabajo de Sistemas de Sistemas en 2011 que proporcionó liderazgo a través de seminarios web y publicaciones, incluyendo los Puntos Débiles (Pain Points) de los SdS (2013) [19], la Introducción (Primer) a ISdS (2018) [20] y la Guía de los Estándares de SdS (2020) [21]. Los SdS se incluyeron como un área de conocimiento en el SE Body of Knowledge en la primera versión en 2012.

Tradicionalmente, se ha considerado que los SdS se aplican principalmente a la defensa, principalmente centrado en Estados Unidos. Sin embargo, un meta-

análisis de 168 artículos del IEEE publicados entre 2020 y 2023 sobre SdS indica que esto ya no es así [22]. Como se muestra en la Tabla 1, las aplicaciones de los SdS en estos artículos abordan una amplia gama de dominios y menos de una quinta parte de los artículos versan sobre aplicaciones de defensa. Además, el meta-análisis reveló que los autores de los artículos procedían de 29 países, lo que muestra un interés geográficamente diversificado por los SdS.

Área de Aplicación	#	%
Defensa	31	18.5%
Transporte	22	13.1%
Salud	10	6.0%
IOT/CPS	8	4.8%
Energía	7	4.2%
Emerg./Gestión Crisis	6	3.6%
Espacio	4	2.4%
Búsqueda y Rescate	4	2.4%
Educación	3	1.8%
Medioambiente	3	1.8%
Resto (< 3)	70	41.7%

Tabla 1: Ámbitos de aplicación de los artículos.

En Europa, la Comisión Europea inició una iniciativa de investigación sobre los SdS en 2011:

(ICT-2011.3.3) con el objetivo de aumentar la competitividad de la industria europea y permitir a Europa dominar y dar forma a los futuros desarrollos de las TIC (Tecnologías de la Información y la Comunicación) de modo que se satisfagan las futuras demandas de su sociedad y economía. Competitividad significa, en este contexto, que Europa será líder mundial en ISdS, lo que se traducirá en un mayor retorno de la inversión (ROI) para la industria europea, una mayor innovación dentro de la comunidad de sistemas técnicos en el gobierno, la industria y el mundo académico, y la sostenibilidad económica a largo plazo de, y a través de, la ingeniería de grandes sistemas complejos [23].

Esta iniciativa contaba con cuatro grandes proyectos: T-AREA-SoS (Trans-Atlantic Research and Education Agenda in Systems of Systems) con el objetivo de desarrollar y entregar a la Comisión Europea una Agenda Estratégica de Investigación en Ingeniería de Sistemas de Sistemas (ISdS), Designing for Adaptability and evolution in System of systems Engineering (DANSE), centrado en el desarrollo de nuevos

enfoques para el diseño y la gestión del funcionamiento de los SdS, Comprehensive Modelling for Advanced Systems of Systems (COMPASS), que desarrolló una nueva tecnología de modelado para software avanzado, y ROAD2SoS, que se centró en el desarrollo de hojas de ruta estratégicas de investigación e ingeniería en Ingeniería de Sistemas de Sistemas y estudios de casos relacionados. Estos proyectos sembraron el interés por los sistemas de sistemas en las universidades y la industria europeas.

Estas acciones han conformado la dirección y el crecimiento de los SdS. En uno de los primeros documentos sobre ingeniería aplicada a los sistemas de sistemas, se define un «SdS como un conjunto o agrupación de sistemas que resulta cuando sistemas útiles e independientes se integran en un sistema mayor que ofrece capacidades únicas» [24]. Este documento señala que los sistemas individuales y los SdS se ajustan a la definición aceptada de sistema en el sentido de que cada uno consta de partes, relaciones y un todo que es mayor que la suma de las partes; sin embargo, aunque un SdS es un sistema, no todos los sistemas son SdS. También aclara que el simple hecho de tener múltiples sistemas no hace que un sistema sea un sistema de sistemas, un sistema de sistemas incluye el hecho de que estos sistemas múltiples den lugar a una nueva capacidad cuyo propósito no estaba anticipado por los sistemas por sí solos.

En 2019, la Organización Internacional de Normalización (ISO) adoptó los primeros estándares para los SdS [25]. Un grupo de estudio de ISO para los estándares que aplican a los SdS reconoció la creciente atención prestada a los SdS y el valor de los estándares para la maduración de la ISdS [26]. Entre ellos se incluye una definición de SdS y de sistemas constituyentes [27]:

- **Sistema de Sistemas (SdS):** Conjunto de sistemas o elementos de sistemas que interactúan para proporcionar una capacidad única que ninguno de los sistemas constituyentes puede lograr por sí solo.
- **Sistemas constituyentes:** Los sistemas constituyentes pueden formar parte de uno o varios SdS. Cada constituyente es un sistema útil por sí mismo, con su propio desarrollo, recursos y objetivos de gestión, pero interactúa dentro del SdS para proporcionar la capacidad única del SdS.

Así se estableció la primera definición de SdS ampliamente aceptada y se sentaron las bases de la actual comunidad de SdS. Nótese que la definición de sistema en ISO es esencial para interpretar la definición de SdS. Esto se debe a que un sistema (que forma SdS) debe ser útil por sí mismo y decide interactuar dentro del SdS. Por ejemplo, una batería que alimenta el motor de un coche no se consideraría un sistema constituyente en este contexto, sino un elemento del coche.

Transporte	Múltiples activos y servicios de transporte trabajan juntos para proporcionar capacidades de transporte integradas	Gestión de Tráfico Aéreo: SES: Single European Sky, Actividades SESAR y el camino hacia el Cielo Digital Europeo [28], SkyNex iTEC [29], próxima generación de Sistema de Control de Tráfico Aéreo. Red ferroviaria: Single European Rail Area, European Rail Traffic Management System (ERTMS) y otros proyectos TEN-T [30].
Energía	La red inteligente, las casas inteligentes y la producción/consumo integrados proporcionan servicios de gestión de la energía	Red Europea de Operadores de Sistemas de Transmisión para Electricidad (ENTSO-E) vision sobre el Sistema de Energía Europeo.[31]
Sanidad	Gestión de infraestructuras regionales, servicios de emergencia y gestión de la salud personal	Iniciativa E-health: conectando sistemas de salud en Europa.[32]
Defensa	Misiones militares como defensa antimisiles, sensores en red, mando y control	Sistemas ISR Conjuntos o Cooperantes compuestos por sistemas ISR Navales, Espaciales, Terrestres, Aéreos, colaborando para ISR Conjunto a nivel nacional o internacional. [33] AFSC: Alliance Future Surveillance and Control.[34] NGWS/FCAS: New Generation Weapon System / Future Combat Air System [35]. FCAS europeo mostrará la próxima generación de cazas volando junto a aviones de transporte no tripulados de varias dimensiones. Estos elementos serán parte de un SdS totalmente en red. FMN [36]: Federated Mission Networking es una capacidad que tiene como objetivo apoyar el mando y el control y la toma de decisiones en operaciones futuras mediante un mejor intercambio de información.
Telecomunicaciones	Los sistemas de telecomunicaciones prestan servicios de telecomunicaciones a múltiples dominios.	GNSS/Galileo [37]: Galileo es el Sistema de Navegación por Satélite Global europeo (GNSS) proporciona información temporal y de posición utilizado en teléfonos y en aplicaciones como ferrocarril, aviación, agricultura, marítimo y más. Ciudades Inteligentes [38] con iniciativas como ICC [39]: The Intelligent Cities Challenge (ICC) es una de las mayores iniciativas de la Comisión Europea que apoya a las ciudades en su transición verde y digital. La Constelación de Satélites IRIS2 [40] es el tercer buque insignia de la Unión Europea, que aborda desafíos a largo plazo en seguridad y resiliencia, ofreciendo servicios mejorados de conectividad a usuarios gubernamentales.
Gestión de recursos naturales	Medio ambiente global, recursos hídricos regionales, silvicultura y recursos recreativos	Copernicus [41] es el componente de observación de la Tierra del programa especial de la Unión Europea, observando nuestro planeta y su medioambiente para el beneficio de los ciudadanos europeos. Ofrece servicios de información que se basan en datos de observación de la Tierra por satélite y datos in situ (no espaciales).
Seguridad y respuesta a desastres	Respuestas a catástrofes como incendios forestales, inundaciones, atentados terroristas y control de fronteras	Mecanismo de Protección Civil Europeo y el Centro de Coordinación de Respuestas a Emergencias (ERCC) [42] Mecanismo Europeo de Gestión de Crisis [43]. La UE podría estar expuesta a una variedad de crisis y desastres (tales como los causados por el cambio climático, amenazas a la salud, ciberataques y ataques terroristas, inestabilidad política, conflictos violentos, fallos en infraestructuras críticas) y debería ser capaz de responder rápido y apropiadamente. EuroSur/FRONTEX [44] el Sistema de Vigilancia de Fronteras Europeo (EUROSUR) es un marco para el intercambio de información y cooperación entre los Estados Miembros y Frontex para mejorar el conocimiento de la situación y aumentar la capacidad de reacción en las fronteras externas.
Ciencia	Astronomía, centros de cálculo, centros de investigación	Astronomía: Telescopios distribuidos como LOFAR [45] (Low Frequency Array), la red europea VLBI [46], BOOTES [47] la primera red mundial de telescopios robóticos. CERN. El complejo de aceleradores en el CERN [48] es una sucesión de máquinas con energías cada vez más altas. EuroHPC JU [49], es una iniciativa conjunta entre la UE, países europeos y socios privados para desarrollar un Ecosistema de Supercomputación en Europa de Clase Mundial.

Tabla 2: Ejemplos de sistemas que podrían considerarse potencialmente Sistemas de Sistemas.

2. CARACTERÍSTICAS DE LOS SISTEMAS DE SISTEMAS

2.1. Distinguiendo Sistemas de Sistemas

Como se ha señalado anteriormente, la caracterización de Maier de los SdS es fundamental para nuestra comprensión de los SdS. Maier [14] presentó cinco características de los SdS que pueden observarse en la mayoría de los SdS, aunque no todas ellas son condiciones necesarias para que algo sea considerado un SdS:

- Independencia de operación de los sistemas constituyentes.
- Independencia de gestión de los sistemas constituyentes.
- Distribución geográfica.
- Comportamiento emergente.
- Procesos de desarrollo evolutivo.

De éstas, Maier identificó la independencia de operación y la independencia de gestión como las dos principales características distintivas para aplicar el término «sistema de sistemas». Argumenta que **un sistema que carece de al menos una de estas dos características no se considera un SdS**, independientemente de la complejidad o la distribución geográfica de sus componentes.

En términos de comportamiento emergente [50], «el concepto de emergencia se refiere a fenómenos que ocurren a nivel de sistema sin estar presentes a nivel de elementos del sistema». Para sistemas de sistemas, esto significa que puede haber un comportamiento de resultados no previstos por los sistemas constituyentes individuales. Como se analiza en el Cuerpo de Conocimientos de Ingeniería de Sistemas [51]:

En la caracterización de Maier, la emergencia se señala como una característica común de los SdS, particularmente en SdS compuestos por múltiples sistemas grandes existentes, basándose en el desafío (en términos de tiempo y recursos) de someter todos los caminos lógicos posibles a la multitud de funciones, capacidades y datos de los sistemas en un SdS... [Por lo tanto] existen riesgos asociados con comportamientos inesperados o no intencionados resultantes de la combinación de sistemas con un comportamiento complejo individual. Estos riesgos se agravan en casos en los que la seguridad, por ejemplo, se ve amenazada por interacciones no intencionadas entre las funciones proporcionadas por los múltiples sistemas constituyentes de un SdS.

Finalmente, en términos de distribución geográfica, esto es algo que comparten la mayoría de los SdS existentes, pero es más una consecuencia de cómo se despliegan la mayoría de los sistemas de sistemas que un requisito para ser un SdS.

2.2. Desafíos (Pain Points) de los Sistemas de Sistemas

Las características identificadas en la sección anterior son importantes porque proporcionan las claves para los desafíos particulares a los que se enfrentan los ingenieros de sistemas al aplicar la ingeniería de sistemas a los SdS. Estos desafíos se reflejan en los Puntos Débiles de los SdS [19] (Figura 2), identificados por el grupo de trabajo de SdS de INCOSE y que han sido particularmente útiles para comprender los SdS.



Figura 2: Desafíos (Pain Points) de los Sistemas de Sistemas [40].

Estos desafíos se describen brevemente en la tabla 3. Se presentan en el Manual de Ingeniería de Sistemas de INCOSE [52] con especial atención al impacto que tienen en la aplicación de la ingeniería de sistemas a los SdS.

Autoridades de los SdS	En un SdS, cada sistema constituyente tiene un “propietario”, partes interesadas, usuarios, procesos de negocio y un enfoque de desarrollo, alejándose de la autoridad tradicional jerárquica sobre el desarrollo y la operación del SdS.
Liderazgo	La falta de autoridad común en un SdS significa que las decisiones relativas al SdS dependen menos del mando y control tradicionales y más de la influencia y la persuasión, que se pueden lograr de varias maneras, siendo una de ellas a través del liderazgo.
Perspectivas de los Sistemas Constituyentes	La mayoría de los SdS están compuestos por sistemas constituyentes preexistentes, cada uno de los cuales aporta su propia perspectiva, que puede estar o no alineada con las perspectivas de los otros sistemas constituyentes del SdS.
Capacidades y requisitos	Tradicionalmente, un sistema cuenta con un conjunto coherente de capacidades de usuario. Los SdS se componen de múltiples sistemas independientes, cada uno con sus propias capacidades, que al combinarse pueden o no proporcionar capacidades de SdS coherentes.
Autonomía, interdependencia y emergencia	La independencia de los sistemas constituyentes significa que un sistema constituyente puede cambiar independientemente del SdS, e incluso abandonar el SdS, e impactar a otros sistemas constituyentes y al SdS de maneras inesperadas o impredecibles.
Pruebas, validación y aprendizaje	Dado que los SdS se componen de sistemas constituyentes independientes, esto plantea desafíos a la hora de realizar pruebas de los SdS de principio a fin como se hace habitualmente con los sistemas.
Principios de los SdS	Es necesario trabajar para identificar y articular los principios transversales que se aplican a los SdS en general, y para desarrollar ejemplos prácticos de la aplicación de estos principios.

Tabla 3. Desafíos (Pain Points) de los SdS.

2.3. Taxonomía de los Sistemas de Sistemas

La caracterización definitoria del SdS como carente de una autoridad de alto nivel es fundamental para los desafíos que plantean los SdS. Esta característica es la que impulsa la taxonomía más aceptada para los SdS. Esta taxonomía ha sido adoptada en la norma ISO/IEC/IEEE 21841. Como se describe en el área de conocimiento de SdS del Cuerpo de Conocimientos de Ingeniería de Sistemas [51]:

En aquellas situaciones en las que el SdS se reconoce y se trata como un sistema en sí mismo, un SdS puede describirse como uno de estos cuatro tipos [14, 53, 54]:

- **Dirigido:** El SdS se crea y gestiona para cumplir propósitos específicos y los sistemas constituyentes están subordinados a él. Los sistemas constituyentes mantienen la capacidad de operar de forma independiente; sin embargo, su modo operativo normal está subordinado al propósito de gestión centralizado.
- **Reconocido:** El SdS cuenta con objetivos reconocidos, un gestor designado y recursos para el SdS; sin embargo, los sistemas constituyentes conservan su propiedad, objetivos, financiación y enfoques de desarrollo y mantenimiento de manera independiente. Los cambios en los sistemas se basan en acuerdos de cooperación entre el SdS y los sistemas.
- **Colaborativo:** Los sistemas que lo componen interactúan de forma más o menos voluntaria para cumplir con los objetivos centrales acordados. Los actores centrales deciden colectivamente cómo prestar o denegar el servicio, lo que proporciona medios para aplicar y mantener los estándares.
- **Virtual:** El SdS carece de una autoridad central de gestión y de un propósito acordado centralmente. Surge un comportamiento a gran escala —y podría ser deseable—, pero este tipo de SdS debe depender de mecanismos relativamente invisibles para mantenerlo.

La Figura 3 ilustra estos 4 tipos.

En realidad, la mayoría de los SdS reales son una combinación de estos tipos.

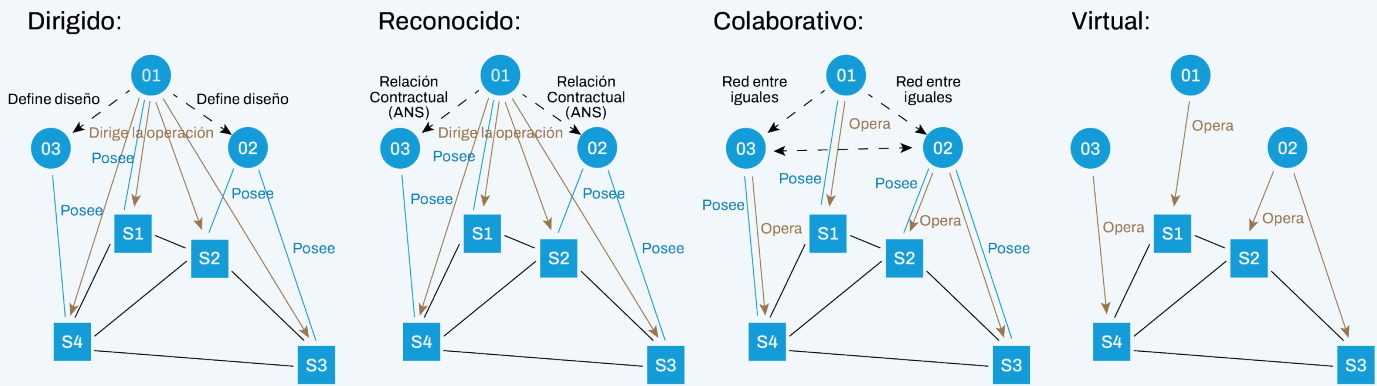


Figura 3: Tipos de SdS [55].

2.4. Escala, alcance y complejidad de los SdS

El SdS puede variar en escala y alcance, como se muestra en la Figura 4, que extrae ejemplos de la iniciativa de investigación en SdS de 2011 de la Comisión Europea, analizada anteriormente.

Por un lado, los SdS pueden estar compuestos por sistemas puramente tecnológicos con la integración de sistemas heterogéneos desarrollados independientemente en una capacidad compuesta. La figura ilustra la integración de capacidades audiovisuales desarrolladas independientemente en un sistema de sistemas de entretenimiento audiovisual doméstico.

Existen SdS sociotécnicos más amplios en áreas como la respuesta a desastres, que incluyen interacciones entre diversos elementos en la respuesta a un desastre, incluidos los incendios, la seguridad pública, las organizaciones de voluntarios y otros.

Por último, los SdS pueden abordar cuestiones que afectan a toda la organización, como el fraude en defensa en EE. UU. [56], que incorporan sistemas, organizaciones, políticas y esfuerzos competitivos.

En cada caso, hay elementos independientes –algunos tecnológicos, otros organizativos– que componen el SdS.

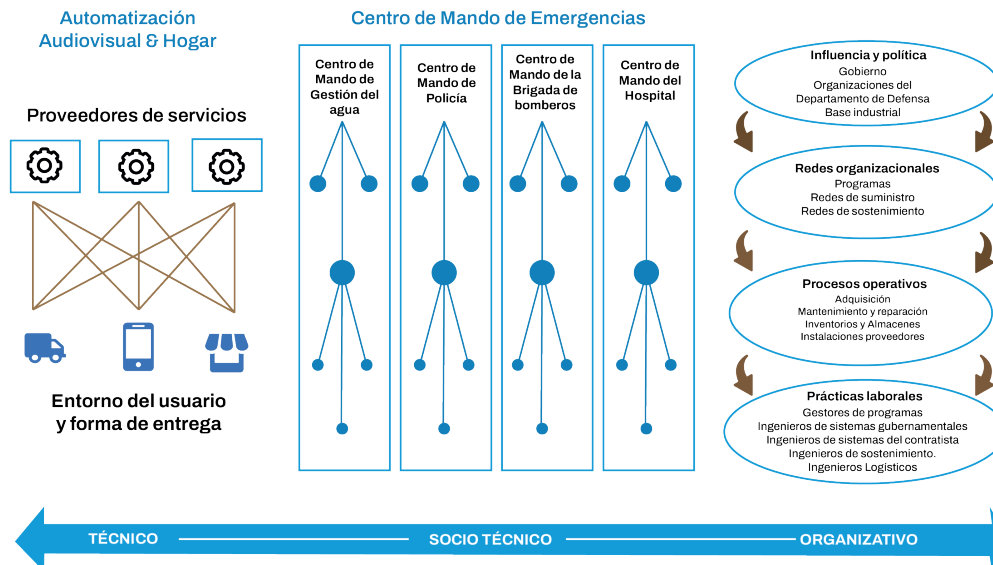


Figura 4: Escala y alcance de los Sistemas de Sistemas [57].

A medida que los SdS aumentan en alcance y escala e incorporan cada vez más elementos no tecnológicos, como se señala en la característica de Maier sobre emergencia en los SdS, los SdS son particularmente susceptibles a la complejidad [58], que se define según el SEBOK [59] como:

“Una medida de la dificultad para comprender el comportamiento de un sistema o predecir las consecuencias de modificarlo. Se produce cuando no existe una relación simple entre la función de un elemento individual y la función del sistema en su conjunto, y cuando el sistema incluye algún elemento de adaptación o resolución de problemas para alcanzar sus objetivos en diferentes situaciones. Puede verse afectado por atributos objetivos de un sistema, como la cantidad, los tipos y la diversidad de elementos y relaciones del sistema, o por las percepciones subjetivas de los observadores del sistema debido a su experiencia, conocimientos, formación u otras consideraciones sociopolíticas.”

Utilizando el trabajo sobre complejidad de INCOSE como marco de referencia [60], se puede demostrar que los SdS por su propia naturaleza exhiben muchas dimensiones de complejidad y que los principios rectores del pensamiento sobre complejidad [61] se pueden aplicar para abordar la complejidad en los SdS.

3. IMPACTO DE LAS TECNOLOGÍAS AVANZADAS EN LOS SISTEMAS DE SISTEMAS¹

Se reconoce que los avances tecnológicos actuales impactan los sistemas de diversas maneras. La pregunta que se aborda aquí es cómo los avances tecnológicos afectan a los SdS. La Figura 5 muestra el conjunto de tecnologías que tienen impacto en los sistemas de sistemas.

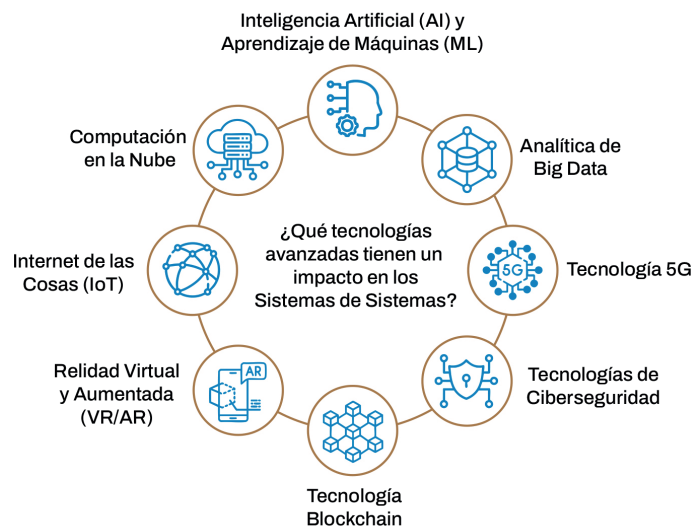


Figura 5. Tecnologías que afectan a los Sistemas de Sistemas.

Las oportunidades para aplicar **la inteligencia artificial y el aprendizaje automático** están creciendo rápidamente, especialmente con la llegada del acceso a grandes modelos de lenguaje. Para los SdS, estas tecnologías brindan oportunidades para automatizar tareas complejas, optimizar el rendimiento del sistema y realizar análisis predictivos. Con inteligencia artificial (IA) y machine learning (ML), los SdS pueden aprender y evolucionar con el tiempo, lo que le permite adquirir a los SdS características de desarrollo evolutivo e independencia operativa. Estas ventajas potenciales conllevan los desafíos de la necesidad de alta potencia computacional, conocimiento especializado y mayor imprevisibilidad. En cuanto a los desafíos de los SdS, la IA/ML aborda los desafíos de autoridad de los SdS con oportunidades para la toma de decisiones inteligente y nuevas capacidades para abordar los desafíos de capacidad y requisitos, proporcionando un medio para aprender y adaptarse a los cambios. Por otro lado, la naturaleza opaca de algunos modelos de IA/ML, a menudo denominada el problema de la “caja negra”, puede dificultar la comprensión y la predicción del comportamiento del sistema.

El **análisis de big data** ofrece la oportunidad de gestionar y analizar grandes volúmenes de datos de diferentes sistemas, lo que proporciona información para el diseño y la operación de los SdS, facilita la toma de decisiones basada en datos y ofrece oportunidades para mejorar la eficiencia y el rendimiento general. Estas oportunidades potenciales conllevan los desafíos de gestionar tanto el gran volumen, como la velocidad y la variedad de datos, incluyendo la gestión, el almacenamiento, el análisis, la calidad y la integridad de los

1. El material de esta sección se basa en la presentación del panel de INCOSE de la Conferencia IEEE SoS en junio de 2024 en Tacoma, Washington, utilizando MChat, el modelo LLM Microsoft Azure OpenAI GPT-4 32K disponible en MITRE.

mismos. Desde la perspectiva de los desafíos de los SdS, el análisis de big data puede proporcionar información sobre el comportamiento y el rendimiento de los sistemas, así como oportunidades para una mejor integración y gestión de los sistemas constituyentes. Sin embargo, la falta de una autoridad transversal en SdS puede plantear problemas de propiedad y privacidad de los datos y, dada la diversidad de perspectivas de los sistemas constituyentes, problemas asociados a la necesidad de integrar y gestionar grandes volúmenes de datos de diferentes sistemas.

La **tecnología 5G** proporciona comunicaciones de alta velocidad, fiables y de baja latencia, lo que es crucial en numerosos sistemas de sistemas. Como resultado, puede facilitar una mejor comunicación, así como la colaboración y el intercambio de datos en tiempo real. Esto conlleva el reto de una inversión significativa en infraestructura y nuevos desafíos en materia de seguridad y privacidad de los datos. En cuanto al desafío de los SdS de autonomía, interdependencia y emergencia, la ventaja de la conectividad de alta velocidad y el intercambio y la comunicación de datos en tiempo real de baja latencia entre sistemas conlleva el reto de una mayor complejidad y la posibilidad de interferencias entre sistemas.

Las **tecnologías de ciberseguridad** pueden ayudar a proteger el SdS de posibles amenazas y ataques. Aun así, el SdS sigue enfrentándose a los desafíos de la seguridad de cada sistema constituyente y su impacto en el SdS general. Dada la falta de autoridad del SdS, estos se enfrentan a los retos de garantizar la seguridad e integridad de los datos en todos los sistemas y generar confianza entre los diferentes sistemas y sus autoridades, incluyendo la necesidad de coordinación y cumplimiento de diferentes protocolos y estándares de seguridad en todos los sistemas.

Blockchain puede garantizar la transparencia y la seguridad en las transacciones y brindar la oportunidad de crear sistemas seguros y transparentes, mejorar la confianza entre diferentes sistemas y partes interesadas y garantizar la integridad y seguridad de los datos, pero esto requiere una comprensión profunda de la tecnología.

Las tecnologías de **realidad virtual y aumentada** permiten la simulación de escenarios del mundo real. Esto aborda el desafío de pruebas, validación y aprendizaje de los SdS. Proporciona tecnología para emplear plataformas inmersivas e interactivas para probar y validar sistemas, brindando la oportunidad de crear sistemas seguros y transparentes y de fortalecer la confianza entre los diferentes sistemas y partes interesadas. Sin embargo, esta tecnología puede presentar desafíos técnicos, que requieren conocimientos especializados, y es propensa a generar expectativas poco realistas.

La **computación en la nube** permite la integración de diversos sistemas en una plataforma común, lo que facilita la interacción y la colaboración, la independencia operacional y la distribución geográfica. Las principales oportunidades de la computación en la nube para los SdS incluyen una mejor integración e interoperabilidad de los sistemas y proporciona plataformas escalables y flexibles que se adaptan a las necesidades cambiantes de los SdS. Los desafíos en materia de seguridad y privacidad de los datos, así como la gestión e integración de diversos sistemas, persisten a medida que el control de los datos se distribuye entre diferentes servicios en la nube y surgen problemas de soberanía y cumplimiento normativo de los datos debido a la falta de una autoridad centralizada del SdS.

Las tecnologías del **Internet de las Cosas (IoT)** proporcionan interconectividad entre diferentes sistemas, permitiéndoles comunicarse y compartir datos. Estas tecnologías facilitan la distribución geográfica, la recopilación y comunicación de datos en tiempo real, el diseño y la operación eficientes de los sistemas y una mayor adaptabilidad a las condiciones cambiantes. Al mismo tiempo, se enfrentan a los desafíos de la sobrecarga de datos, la gestión y el análisis de datos distribuidos y la seguridad de los dispositivos IoT, dada la complejidad derivada de la gran cantidad de dispositivos y sistemas que deben gestionarse y coordinarse, lo que introduce nuevas vulnerabilidades de seguridad.

4. CONCLUSIONES

Se puede afirmar con seguridad que los sistemas de sistemas han llegado para quedarse y, en todo caso, se están reconociendo como una clase de sistemas que merece especial consideración. Con la llegada de la modularidad en el diseño de sistemas, lo que antes se consideraban sistemas tecnológicos únicos, ahora exhiben muchas de las características de los SdS, ya que cada vez menos sistemas se desarrollan completamente desde cero. Cada vez más componentes de sistemas desarrollados por organizaciones independientes se integran como componentes de nuevos sistemas, lo que conlleva las ventajas de la reutilización, junto con algunos de los desafíos de los SdS. A medida que la ingeniería de sistemas continúa aplicándose a organizaciones socio-técnicas de mayor tamaño, estos sistemas de sistemas organizativos forman parte integral del futuro panorama de sistemas y del ámbito de competencias de la ingeniería de sistemas.

Esta sección ha sentado las bases para los temas de ingeniería de SdS que se abordan en el resto de esta monografía. Se ha revisado la historia de SdS, así como las características críticas que distinguen a este tipo de sistema y cómo estas características impulsan su complejidad. El capítulo ha mostrado cómo los SdS se encuentran en una amplia gama de dominios y es objeto de investigación a nivel mundial. Las tecnologías avanzadas son impulsores clave de los SdS, cuya importancia será cada vez mayor en el futuro. Estas tecnologías proporcionan un contexto importante para comprender los desafíos y los enfoques de la ingeniería de sistemas de sistemas.

REFERENCIAS

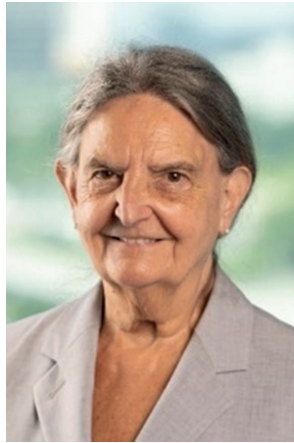
1. Gorod, Alex, Brian Sauser, and John Boardman. System-of-Systems Engineering Management: A Review of Modern History and a Path Forward. 2015 IEEE SYSTEMS JOURNAL, VOL. 2, NO. 4, DECEMBER 2008
2. Neilsen, Claus Ballegaard, Peter Gorm Larsen, John Fitzgerald, Jim Woodcock, and Jan Peleska. 2015. Systems of systems engineering: Basic concepts, model-based techniques, and research directions. ACM Comput. Surv. 48, 2, Article 18 (September 2015)
3. Boulding, Kenneth. 1956. General systems theory—The skeleton of science. Management Science 2, 3 (April 1956).
4. Von Bertalanffy, Ludwig. 1969. General Systems Theory. New York: George Braziller.
5. Berry, Brian J. L.. 1964. Cites as systems within system of cites. Papers and Proceedings of the Regional Science Association 13, 1 (Jan. 1964), 149–163.
6. Ackoff, Russell L.. 1971. Towards a system of systems concept. Management Science 17, 11 (July 1971), 661–671.
7. Jacob Francois. 1974. The Logic of Living Systems: A History of Heredity. Allen Lane.
8. Keating, C., R. Rogers, R. Unal, et al. 2003. Systems of systems engineering. Eng. Management J. 15-3: 36–45.
9. Jamshidi, M. 2005. System of systems engineering – A definition. Piscataway, NJ: IEEE SMC.
10. Abeer Sharawi, Serge N. Sala-Diakanda, Sergio Quijada, Nabeel Yousef, Luis Rabelo, and Jose Sepulveda. 2006. A distributed simulation approach for modeling and analyzing system of systems. In 2006 Winter Simulation Conference.
11. Lane, J. A. and R. Valerdi, “Synthesizing SoS concepts for use in cost estimation,” presented at the IEEE Conf. Syst., Man, Cybern. Waikoloa, HI, 2005.
12. Gorod, A., B. Sauser, and J. Boardman. 2008. System-of-Systems Engineering Management: A Review of Modern History and a Path Forward. IEEE Systems Journal, 2-4: 484-99. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4682611&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4682611.
13. Ed. Jamshidi, M. 2008. System of System Engineering – Innovations for the 21st Century. Hoboken, NJ: Wiley.
14. Maier, M. 1998. Architecting principles of systems-of-systems. 6th Ann. Int. Symp. Int. Council Syst. Eng. Boston, MA.
15. Owens, William A.. “The Emerging US Systems-of-Systems” in Strategic Forum, Institute for National Strategic Studies, The National Defense University, Number 63, February 1996.
16. United States Air Force Scientific Advisory Board. Report on System-of-Systems Engineering for Air Force Capability Development Executive Summary and Annotated Brief. SAB-TR-05-04. July 2005.
17. Department of Defense. 2001. Army software blocking policy: Version 11.4E.
18. DoD. 2009. System of systems, systems engineering guide: Considerations for systems engineering in system of systems environment. U. S. Department of Defense.
19. J. Dahmann. 2014. Systems of systems pain points. In INCOSE International Symposium on Systems Engineering 2014.
20. INCOSE. 2018. Systems of Systems Primer. INCOSE-TP-2018-003-01.0.
21. INCOSE. 2018. Guide to SoS Standards
22. Dahmann, Judith. Current Landscape of Systems of Systems Engineering, IEEE SoSE Conference, Tacoma, WA, June 2024.
23. Henson, S.A, M.J.D. Henshaw, V. Barot, C.E. Siemieniuch, M.A. Sinclair M. Jamshidi H. Dogan, S.L. Lim, C. Ncube D. DeLaurentis, Towards a Systems of Systems Engineering EU Strategic Research Agenda. Proc. of the 2013 8th International Conference on System of Systems Engineering, Maui, Hawaii, USA - June 2-6, 2013
24. DAU, “Defense acquisition guidebook,” 2004.
25. ISO/IEC/IEEE 21839 (ISO, 2019)
26. ISO SoS Standards study group report (ISO, 2016)
27. Quick Reference Guide to SoS Standards, INCOSE, 2020.
28. <https://www.sesarju.eu/MasterPlan2025> Accessed March 2025
29. <https://www.itecskynex.com/> Accessed March 2025
30. https://transport.ec.europa.eu/transport-themes/infrastructure-and-investment/trans-european-transport-network-ten-t_en Accessed March 2025
31. <https://vision2030.entsoe.eu/> Accessed March 2025
32. https://health.ec.europa.eu/document/download/3dc615dc-e94c-4691-b655-a2877a6aa57c_en?filename=2016_ehealthleaflet_vertical_en.pdf Accessed March 2025
33. https://www.nato.int/cps/fr/natohq/topics_111830.htm?selectedLocale=en Accessed March 2025
34. https://www.nato.int/nato_static_fl2014/assets/pdf/2020/7/pdf/200701-Factsheet_Alliance_Future_Surveil-1.pdf Accessed March 2025

35. <https://www.airbus.com/en/newsroom/stories/2023-11-future-combat-air-system-fcas-enter-the-internet-of-military-things> Accessed March 2025
36. <https://www.act.nato.int/activities/federated-mission-networking/> Accessed March 2025
37. https://defence-industry-space.ec.europa.eu/eu-space/galileo-satellite-navigation_en Accessed March 2025
38. https://commission.europa.eu/eu-regional-and-urban-development/topics/cities-and-urban-development/city-initiatives/smart-cities_en Accessed March 2025
39. <https://www.intelligentcitieschallenge.eu/about-icc> Accessed March 2025
40. https://defence-industry-space.ec.europa.eu/eu-space/iris2-secure-connectivity_en Accessed March 2025 https://civil-protection-humanitarian-aid.ec.europa.eu/what/civil-protection/eu-civil-protection-mechanism_en Accessed March 2025
41. <https://www.copernicus.eu/en> Accessed March 2025
42. https://civil-protection-humanitarian-aid.ec.europa.eu/what/civil-protection/eu-civil-protection-mechanism_en Accessed March 2025
43. https://joint-research-centre.ec.europa.eu/jrc-science-and-knowledge-activities/crisis-management_en Accessed March 2025
44. https://home-affairs.ec.europa.eu/policies/schengen-borders-and-visa/border-crossing/eurosur_en Accessed March 2025
45. <https://www.glowconsortium.de/index.php/en/lofar-about-new> Accessed March 2025
46. <https://evlbi.org/> Accessed March 2025
47. <https://bootesnetwork.com/> Accessed March 2025
48. <https://home.cern/science/accelerators> Accessed March 2025
49. https://eurohpc-ju.europa.eu/index_en Accessed March 2025
50. Axelsson, Jacob. What Systems Engineers Should Know About Emergence. First published: 26 September 2022; <https://doi.org/10.1002/iis2.12982>
51. Systems Engineering Body of Knowledge (SEBOK), [https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)](https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)), Accessed March 2025
52. INCOSE. 2015. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Version 3.2.2. (4th ed.). Wiley.
53. Dahmann, Judith and Kristen Baldwin. 2008. Understanding the current state of US defense systems of systems and the implications for systems engineering. In IEEE Systems Conference. IEEE.
54. ISO 21841, Taxonomy of systems of systems, 2019.
55. Henshaw, Michael. Emerging Strategic Research and Education Agenda in SoS: Trans-Atlantic Research and Education Agenda in Systems of Systems. NDIA presentation, April, 2013.
56. Bodner, Douglas. Mitigating Counterfeit Part Intrusions with Enterprise Simulation. *Procedia Computer Science* 61 (2015) 233 – 239
57. Dahmann, Judith. Systems of Systems Characterization and Types. NATO Publication Ref NBR EN-SCI-276-01.2015.
58. Dahmann, Judith and Dan DeLaurentis, Unique Challenges in System of Systems Analysis, Architecting, and Engineering. In *Systems Engineering for the Digital Age*. Wiley. 2024.
59. SEBOK: Emerging Topics: SoS and Complexity. [https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_\(SEBoK\)](https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)) Accessed March 2025
60. Watson et al. "Appreciative Methods Applied to the Assessment of Complex Systems", 29th INCOSE international Symposium, Orlando, Florida, July 20-25, 2019
61. INCOSE Complexity Primer; INCOSE, 2016

BIOGRAFÍAS

DRA. JUDITH DAHMANN

La Dra. Judith Dahmann es miembro de MITRE, en MITRE Corporation, y líder de MITRE para actividades de Apoyo Técnico de Integración de Misión, en la Oficina del Subsecretario de Defensa para Investigación e Ingeniería del Departamento de Defensa de los EE.UU. Lidera el equipo que respalda las actividades de ingeniería de misión para misiones de defensa prioritarias seleccionadas y la aplicación de la ingeniería digital a la ingeniería de misión. Fue líder técnica en el desarrollo de la guía del Departamento de Defensa para ingeniería de sistemas aplicada a sistemas de sistemas (SdS) y lideró el proyecto de la Organización Internacional de Normalización (ISO) 21839, el primer estándar internacional ISO sobre "Consideraciones SdS para Sistemas a lo largo de su Ciclo de Vida". Antes de esto, la Dra. Dahmann fue científica jefa de la Oficina de Modelado y Simulación de Defensa de la Dirección de Investigación e Ingeniería de Defensa de los EE.UU. (1995-2000), donde dirigió el desarrollo de la Arquitectura de Alto Nivel, una arquitectura software distribuida de propósito general para simulaciones, ahora un estándar IEEE (IEEE 1516). La Dra. Dahmann es fellow del Consejo Internacional de Ingeniería de Sistemas (INCOSE) y copresidente del Grupo de Trabajo sobre Sistemas de Sistemas de INCOSE y del Comité SE de SdS de la División SE de la Asociación de la Industria de Defensa Nacional.



DAVID SÁNCHEZ GARCÍA

David Sánchez García es ingeniero de telecomunicaciones por la Universidad Politécnica de Madrid (UPM). Cuenta con más de 30 años de experiencia en Isdefe en ingeniería de sistemas, con especial dedicación a los sistemas de guerra electrónica e inteligencia de señales, donde aplica enfoques innovadores de sistemas de sistemas en el Ministerio de Defensa.



Actualmente, lidera la asistencia técnica de Isdefe a la Oficina del Programa Conjunto de Inteligencia de Señales SANTIAGO de la SDG-PROGRAMAS de la Dirección General de Armamento y Material (DGAM).

Está certificado como CSEP por INCOSE y como PMP por PMI. También participa como formador interno en ingeniería de sistemas en Isdefe.



De la Ingeniería de Sistemas a la Ingeniería de Sistemas de Sistemas: Una visión general

Dr. Michael Yokell, *INCOSE (US Aerospace and Defense, retirado)* (Michael.Yokell@INCOSE.net)

Dr. Alejandro Salado, *Universidad de Arizona* (alejandrosalado@arizona.edu)

Adolfo Sánchez Domínguez, *Isdefe* (asdominguez@isdefe.es)

Resumen

La Ingeniería de Sistemas de Sistemas (SdS) presenta desafíos únicos que exigen un cambio con respecto a las prácticas tradicionales de ingeniería de sistemas (SE). Mientras que la mayoría de las prácticas de SE proporcionan un marco estructurado para la gestión de los ciclos de vida de los sistemas, su aplicación directa a los SdS se ve limitada por los diversos grados de gobernanza, independencia de gestión, independencia operativa y restricciones de interoperabilidad entre los sistemas constituyentes (CS). A diferencia de los sistemas tradicionales, donde los mandatos y el control centralizado pueden garantizar el cumplimiento normativo, los SdS deben equilibrar el mando, la negociación y la influencia, en función de la autonomía de sus componentes. Este capítulo ofrece un resumen de las principales adaptaciones que la aplicación de la ingeniería de sistemas requiere para ser eficaz en la ingeniería de los SdS.

Palabras clave

Ingeniería de SdS, independencia de gobernanza, independencia de gestión.



1. INTRODUCCIÓN

La aplicación de los principios tradicionales de la SE al SdS está lejos de ser sencilla. A diferencia de los sistemas convencionales, donde los ingenieros diseñan y controlan cada elemento, el SdS presenta un entorno complejo, descentralizado y en constante evolución, como se describió en el capítulo anterior. El principal desafío radica en que un SdS no se construye desde cero. En cambio, surge de la interacción de múltiples sistemas operados y gestionados de forma independiente, cada uno con sus propias partes interesadas, objetivos y limitaciones operativas.

En este entorno, el rol de los ingenieros de sistemas se transforma de diseñadores y tomadores de decisiones a orquestadores e influenciadores, alineando diversos intereses, fomentando la interoperabilidad y guiando las interacciones del sistema para lograr una mayor capacidad emergente.

Uno de los cambios más fundamentales en la ingeniería de SdS es reconocer que los enfoques de la SE, que son altamente prescriptivos para los procesos de SE, no pueden aplicarse de manera uniforme a los SdS [1]. Para un sistema tradicional, la mayoría de los estándares de SE establecen las acciones necesarias para su desarrollo y operación, como son la gestión de los requisitos, la verificación y validación y la ejecución rigurosa y disciplinada de todos los procesos del ciclo de vida. Se asume implícitamente que existe un control centralizado, lo que significa que los ingenieros pueden imponer procesos según sea necesario para garantizar el éxito del sistema.

Sin embargo, al trabajar con SdS, casi todo depende del nivel de gobernanza, gestión e independencia operativa de sus sistemas constituyentes. A diferencia de un sistema convencional, los ingenieros no pueden asumir que los mandatos serán efectivos en todo el SdS. Algunos sistemas pueden controlarse directamente, mientras que otros operan con total independencia, siguiendo sus propias prioridades y estructuras de toma de decisiones. Esto significa que, si bien en algunos casos los mandatos aún pueden usarse (especialmente para CS desarrollados internamente u obligados contractualmente a cumplir con la estructura del SdS), en otros casos son inútiles (particularmente cuando se trata de sistemas autónomos que existen fuera de la jurisdicción del propietario del SdS). Sin embargo, lo más común es que la solución se encuentre en un punto intermedio, con grados de negociación, influencia y mecanismos de gobernanza utilizados para alinear los sistemas sin imponer un control directo.

Este “depende” es la esencia de la ingeniería de SdS (SoSE, por sus siglas en inglés). El grado de gobernanza y de independencia de gestión y operativa implica que cada decisión de ingeniería, ya sea que involucre arquitectura, integración, verificación o mantenimiento, puede distribuirse debido a las estructuras específicas de gobernanza y gestión del SdS. Por lo tanto, el principal reto de la SoSE no reside únicamente en aplicar los principios y métodos de ingeniería de sistemas, sino en comprender dónde y cómo pueden aplicarse, dada la autonomía, la cooperación y las limitaciones de los CS involucrados.

A pesar de la creciente prevalencia de los SdS en las industrias, los métodos y prácticas para la ingeniería de SdS aún se encuentran en sus primeras etapas. Si bien la ingeniería de sistemas tradicional se ha beneficiado de décadas de perfeccionamiento, con estándares, metodologías y mejores prácticas bien establecidos [2], la SoSE sigue siendo una disciplina emergente, en la que los conceptos fundamentales aún están evolucionando y muchos desafíos siguen sin resolverse.

La ingeniería de sistemas tradicional se ha formalizado y continúa formalizándose mediante aplicaciones en los sectores aeroespacial, de defensa, automotriz e industrial. Los procesos descritos en normas como ISO/IEC/IEEE 15288 [3] y el Manual de Ingeniería de Sistemas INCOSE [4] proporcionan una guía clara sobre cómo abordar el diseño, la integración, la verificación, la validación y el mantenimiento de sistemas. Esto indica que estos métodos están profundamente arraigados en las prácticas de la industria y se están perfeccionando activamente a través de la investigación teórica y la aplicación en el mundo real.

En cambio, la SoSE aún no tiene el mismo nivel de madurez. Si bien se ha realizado un trabajo fundamental, especialmente en defensa e infraestructura a gran escala, aún se está acumulando un amplio corpus de conocimientos, conceptos, métodos y prácticas consolidados. A diferencia de los sistemas tradicionales, que han sido abundantes y con diversas clases de sistemas del mismo tipo, no sólo contamos con un número muy limitado de SdS, sino que estos son inherentemente heterogéneos y dinámicos. De hecho, los desafíos de gobernanza, autonomía y comportamiento emergente varían significativamente entre dominios, lo que dificulta establecer directrices claras (como se mencionó anteriormente, “depende”). Además, si bien la formación en SE está creciendo significativamente, los SdS no son centrales en el contenido educativo. Si los hay, se tratan como una nota al margen o en un curso o asignatura específico, lo que ejemplifica la diferente profundidad con la que se abordan los sistemas tradicionales y los SdS en estos programas. La brecha se ve agravada por el hecho de que la ingeniería de SdS requiere un cambio de paradigma en los métodos de ingeniería, ya que los supuestos fundamentales de las prácticas de ingeniería son diferentes: del control directo a la coordinación, de la previsibilidad a la adaptación y del diseño cerrado de componentes individuales a la habilitación de capacidades emergentes. Estos cambios requieren métodos y enfoques que tradicionalmente han sido ajenos a la ingeniería.

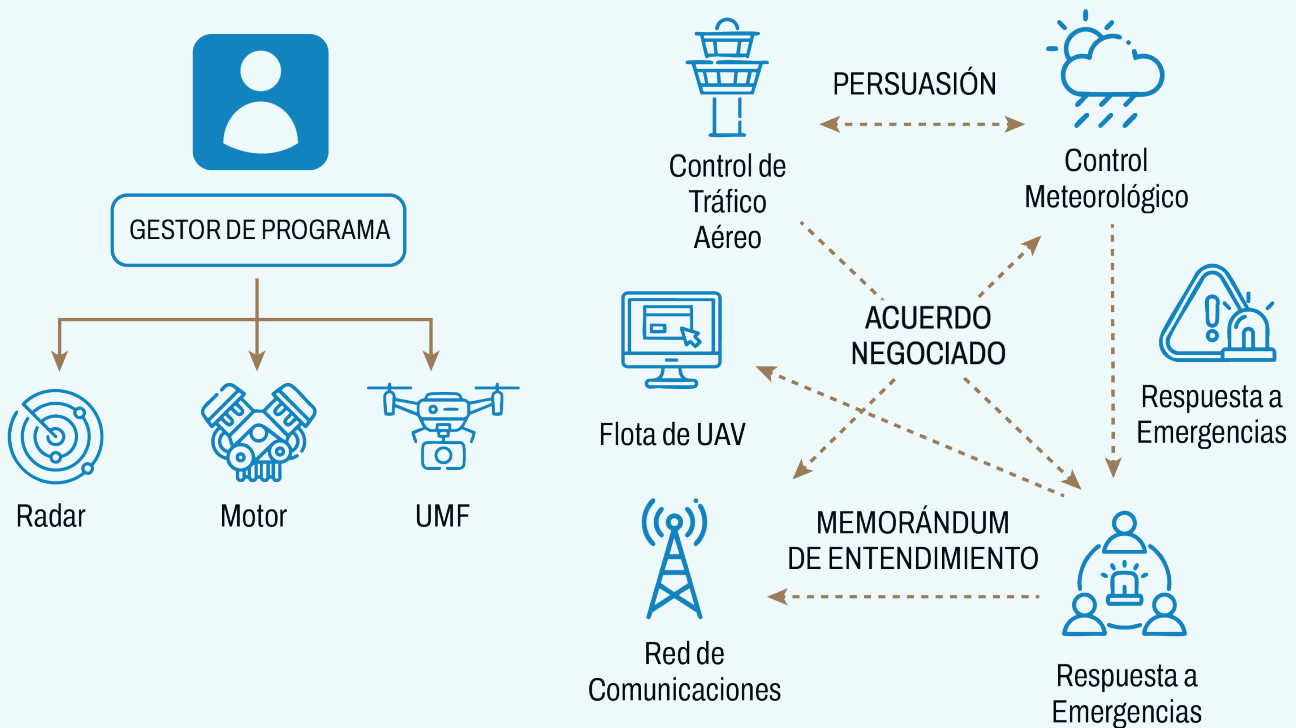


Figura 1. La variabilidad de la independencia de gobernanza, gestión y operación requiere nuevos enfoques de ingeniería de sistemas (Izquierda: mando y control tradicional; Derecha: coordinación en SdS).

Aunque la SoSE aún se encuentra en sus primeras etapas, se están logrando avances. La actividad investigadora está en aumento y están surgiendo diversos marcos y metodologías. Actualmente, podemos aprovechar las directrices, los principios y algunos métodos aislados que se consideran útiles para la SoSE. Estos se han recopilado en tres normas internacionales:

- **ISO/IEC/IEEE 21839 [5]** se centra en consideraciones críticas para el SdS durante las etapas del ciclo de vida de los sistemas constituyentes, alineada con la ISO/IEC/IEEE 15288, y aborda la interacción y el funcionamiento eficaz de los sistemas constituyentes dentro de un SdS.
- **ISO/IEC/IEEE 21840 [6]** ofrece pautas para la aplicación de la ISO/IEC/IEEE 15288 en el contexto de los SdS, explorando las similitudes y diferencias entre los sistemas y los SdS, y detallando prácticas de ingeniería para los SdS.
- **ISO/IEC/IEEE 21841 [7]** define una taxonomía para los SdS con el fin de mejorar la comprensión y la comunicación entre las partes interesadas, que también sirve como base para caracterizar el SdS y guiar mejor la selección de prácticas de la SoSE.

En las siguientes secciones se presenta una visión general de estos principios rectores. En particular, en la Sección 2 se detallan tres aspectos clave de la SoSE que ofrecen una visión general de algunos de los cambios fundamentales de la SE tradicional a la SoSE. Posteriormente, en las Secciones 3 a 6, se expone una visión general de la guía para aplicar los procesos de SE tradicionales a la SdS según [6].

Nota: A lo largo del capítulo, cuando el texto se refiera simplemente a sistemas, esta referencia es a sistemas tradicionales que no incluyen a los SdS.

2. CONSIDERACIONES CLAVE DE LA SOSE

Como se describió anteriormente, la gobernanza, la independencia de gestión y la independencia operativa son las diferencias cruciales que exigen un cambio en las prácticas de SE para los SdS. Éstas dan lugar a la interoperabilidad, la previsibilidad y la evolución como consideraciones clave para definir las prácticas de la SoSE.

2.1. Interoperabilidad: un proceso negociado, no una especificación definida

Dado que los SdS rara vez se diseñan como un todo unificado, uno de los mayores desafíos es lograr que sistemas desarrollados de forma independiente funcionen juntos. La SE tradicional asume que la integración puede planificarse con antelación, pero en los SdS, la integración suele ser un proceso continuo y dinámico.

Un SdS completamente gobernado puede imponer requisitos estrictos de interoperabilidad, obligando a todos los CS a cumplirlos. Sin embargo, en un SdS con gobernanza limitada, los ingenieros deben negociar la interoperabilidad, a veces convenciendo a los propietarios de los sistemas para que adopten estándares o proporcionen acuerdos de intercambio de datos.

En este entorno, las herramientas tradicionales de SE, como los documentos de control de interfaz (ICD), pueden seguir siendo útiles, pero no son suficientes por sí solas. Los ingenieros también pueden necesitar establecer estándares y marcos comunes que los sistemas adopten voluntariamente, utilizar incentivos para fomentar el alineamiento (por ejemplo, proporcionando financiación, soporte técnico o ventajas operativas) y diseñar software intermedio (*middleware*) y arquitecturas adaptativas que permitan la integración de sistemas no conformes sin una estandarización completa.

Por lo tanto, en lugar de imponer la interoperabilidad como un requisito, los ingenieros de SdS a menudo guían la interoperabilidad como un proceso de alineación y adaptación.

2.2. El comportamiento emergente y los límites de la previsibilidad

Una característica definitoria de los SdS es que las capacidades y los comportamientos surgen de las interacciones del sistema, en lugar de estar diseñados explícitamente. En un sistema tradicional, los ingenieros definen cuidadosamente las funciones del sistema. En un SdS, las capacidades suelen surgir como un efecto secundario de la colaboración entre sistemas, a veces beneficioso, a veces problemático.

Debido a esta imprevisibilidad y a que muchos de sus CS están operativos en entornos reales, el SdS no siempre puede probarse en su totalidad, lo que dificulta la adopción de la verificación y validación (V&V) de sistemas tradicional, donde se espera que los sistemas se sometan a una rigurosa campaña mediante prototipado, simulación y pruebas controladas. En cambio, el modelado, la simulación y la experimentación se convierten en herramientas esenciales para evaluar la adaptación del SdS en su conjunto al cambio, en lugar de simplemente verificar si cada CS individual cumple con sus requisitos.

2.3. Ingeniería para la evolución continua

A diferencia de los sistemas tradicionales, que suelen tener ciclos de vida claramente definidos, un SdS está en constante cambio. Se incorporan nuevos sistemas, los antiguos se retiran y los avances tecnológicos transforman constantemente el entorno. La idea de un SdS “finalizado” es un mito. En cambio, los ingenieros deben diseñar para una evolución continua.

Esto significa que los modelos de desarrollo rígidos y secuenciales no funcionan. La SoSE requiere procesos incrementales y adaptativos que permitan el cambio sin interrumpir las capacidades existentes. Para gestionar esto, los ingenieros deben pensar más allá de las soluciones técnicas. Los marcos políticos, legales y contractuales también deben evolucionar junto con los sistemas técnicos. La gestión de riesgos, tradicionalmente centrada en fallos técnicos, ahora debe abarcar también las incertidumbres organizativas, económicas y geopolíticas.

3. PROCESOS DE ACUERDO

3.1. Proceso de adquisición

En un sistema tradicional, la organización propietaria del sistema se denomina generalmente cliente, mientras que las organizaciones que lo proporcionan se denominan proveedores. En el contexto de los SdS, términos como «consumidor», «participante» o «socio» probablemente reflejen mejor la esencia de su relación. En el contexto de los SdS, un consumidor obtiene las capacidades del CS, con o sin acuerdo explícito, y sin adquirir realmente el CS que las generó. A diferencia de los sistemas tradicionales, los proveedores mantienen su independencia administrativa y el CS mantiene su independencia operativa.

Además de los enfoques formales, como los contratos, otros menos formales, como los memorandos de acuerdo y los memorandos de entendimiento, pueden ser eficaces en la gestión de algunos tipos de SdS. Sin embargo, en algunos casos, los acuerdos pueden ser informales, tácitos o inexistentes. Aceptar las condiciones de uso de un producto o servicio es un tipo de acuerdo. Muchos SdS funcionan eficazmente incluso en ausencia de acuerdos.

3.2. Proceso de suministro

Dado que los CS mantienen su independencia de gestión y operativa, ya cuentan con clientes para su producto o servicio. Algunos CS pueden estar dispuestos a ampliar o reasignar recursos para atender las necesidades de un cliente de SdS o a aceptar no cambiar las características de los CS sin previo aviso, pero otros no.

Los acuerdos dentro de los SdS abarcan un amplio espectro de formalidades, desde contratos en un extremo, pasando por enfoques menos formales como memorandos de acuerdo, hasta memorandos de entendimiento o incluso la ausencia total de acuerdos. Los acuerdos de licencia de software, por ejemplo, implican que el uso del software puede constituir un acuerdo a ciertos términos y condiciones. En SdS, donde no existe un acuerdo explícito, el proveedor no tiene obligaciones directas. Sin embargo, los proveedores generalmente quieren que sus productos y servicios sean utilizados, por lo que tienen cierta motivación para ayudar a los consumidores a encontrar utilidad en ellos.

4. PROCESOS FACILITADORES DE PROYECTOS ORGANIZACIONALES

4.1. Proceso de gestión del modelo de ciclo de vida

Los CS tienen sus propios ciclos de vida, modelos de ciclo de vida y pueden encontrarse en distintos momentos dentro de dichos modelos. El modelo de ciclo de vida de un SdS puede ser diferente al del CS. El uso eficaz del proceso de gestión del ciclo de vida proporciona un contexto para la incorporación y salida del CS del SdS. Las organizaciones que gestionan el CS podrían tener poco conocimiento o interés en las políticas, procedimientos y modelos de ciclo de vida del SdS o de otro CS. Sin embargo, es necesario definir y comprender los acuerdos de responsabilidad, rendición de cuentas y autoridad con el CS.

La evaluación de los modelos y procesos del ciclo de vida del SdS para su uso por la organización debe considerar que las preocupaciones, modelos y procesos del ciclo de vida pueden diferir de los de los CS. El modelo y proceso del ciclo de vida del SdS debe reconocer y acomodar el modelo y los procesos del ciclo de vida de los CS. Para

algunos tipos de SdS, los CS pueden ser incapaces o no estar dispuestos a realizar dichas adaptaciones. Sin embargo, la adaptación a las preocupaciones del SdS puede incluir hojas de ruta o estrategias generales. Las mejoras en los procesos, modelos y procedimientos del SdS deben reconocer que las mejoras necesarias para respaldar los SoS pueden ser diferentes de las necesarias para respaldar cada CS por sí solo y que las prioridades del SdS pueden diferir de las de los CS.

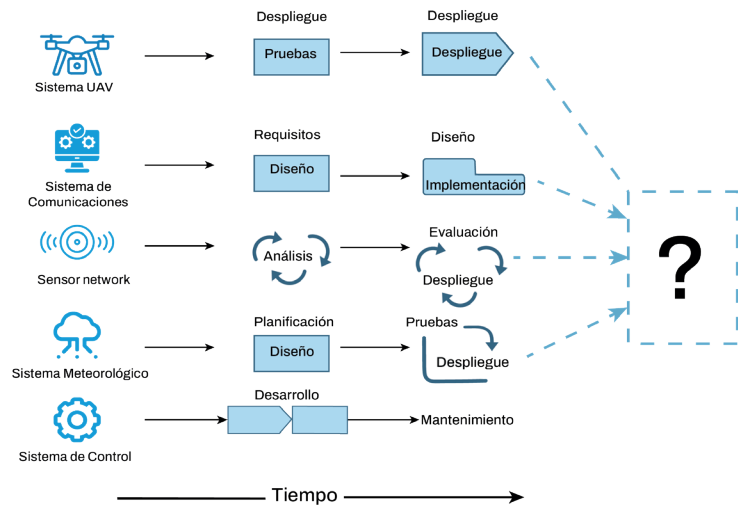


Figura 2. Cada sistema constituyente tiene su propio modelo de ciclo de vida y pueden encontrarse en diferentes etapas.

4.2. Proceso de gestión de infraestructura

Los requisitos de infraestructura para proyectos de SdS pueden ser diferentes a los de proyectos de CS. Por ejemplo, el SdS puede necesitar instalaciones físicas como laboratorios de integración y de pruebas para realizar pruebas de interoperabilidad y abordar problemas del SdS. El SdS puede necesitar elementos de infraestructura de proyecto que ningún CS necesita individualmente. El desarrollo y la adquisición de elementos de infraestructura de proyecto SdS pueden ser muy diferentes a los de los sistemas. Dado que el SdS y los CS pueden encontrarse en diferentes etapas de sus ciclos de vida, la disponibilidad, robustez y resiliencia de la infraestructura de proyecto del SdS puede ser más importante que la de cada CS por sí mismo.

4.3. Proceso de gestión de cartera

La gestión de cartera es más compleja para las organizaciones involucradas en SdS. Para los SdS, la falta de control sobre los CS, que mantienen su independencia de gestión y operativa, puede ser especialmente problemática. Asimismo, la participación de los CS en el SdS puede ser problemática debido al consumo de recursos que exceden los que cada CS necesitaba originalmente para satisfacer las necesidades de sus propios interesados. Para algunos tipos de SdS, la influencia para alcanzar las capacidades del SdS puede generarse destacando los beneficios mutuos con las del SdS y sus interesados, así como con cada CS y sus interesados. Los desincentivos deben abordarse y minimizarse siempre que sea posible.

Dependiendo del grado de independencia operativa y de gestión, pueden requerirse recursos y presupuestos adicionales para respaldar un SdS, además de los necesarios para cada CS. La contabilidad de estos recursos y presupuestos puede ser diferente a la de cada CS debido a su independencia. Las organizaciones del SdS deben considerar que otras organizaciones participantes en el SdS podrían optar por mantener o no proyectos, independientemente del cumplimiento de los acuerdos y los requisitos de las partes interesadas.

4.4. Proceso de gestión de recursos humanos

Los recursos humanos para un SdS abordan actividades y procesos distintos de los procesos de los CS. Dado que la SoSE difiere de la SE, las habilidades requeridas para los proyectos SdS difieren de otros tipos de habilidades necesarias para cada CS. Por ejemplo, dado que cada CS mantiene independencia de gestión, las habilidades relacionadas con la influencia, en lugar de la dirección, pueden ser especialmente importantes. Los métodos para evaluar el retorno de la inversión (ROI) en un proyecto SdS son inmaduros, por lo que se debe tener cuidado al resolver conflictos.

4.5. Proceso de gestión de calidad

Se establece un enfoque de gestión de calidad tanto para el CS como para el SdS. El SdS depende del CS para la gestión de calidad del CS. La gestión de calidad por parte del CS puede variar, desde una gestión muy prescriptiva en todos los CS hasta una ausencia total en algunos. Las organizaciones que participan en un SdS deben ajustar sus políticas, objetivos y procedimientos para adaptarse a estas realidades. Los criterios de evaluación de calidad del SdS deben centrarse en las capacidades del SdS, que pueden ser diferentes de las del CS. Las organizaciones que participan en un SdS deben considerar cómo alinear los enfoques para alcanzar los objetivos integrados del SdS en presencia de sistemas de calidad variables.

4.6. Proceso de gestión del conocimiento

Dado que los CS son independientes en términos de gestión, cuentan con sus propias taxonomías y activos de conocimiento para sus sistemas. Es posible que los CS puedan colaborar con otros CS o con el SdS, pero el SdS no debe asumir que lo harán. El SdS puede utilizar el proceso de gestión del conocimiento de ISO/IEC/IEEE para definir los activos de conocimiento específicos del SdS que puedan ser necesarios. Los activos de conocimiento del SdS deben sobrevivir a cada CS, especialmente cuando los CS entran o salen de un SdS. Dado que los CS residen en diferentes organizaciones, el conocimiento, las habilidades y los activos de conocimiento organizacionales pueden distribuirse entre diferentes organizaciones.

5. PROCESOS DE GESTIÓN TÉCNICA

5.1. Proceso de planificación del proyecto

Las organizaciones crean proyectos para cumplir con los objetivos cambiantes del SdS o las deficiencias de capacidad. Los proyectos pueden crear nuevos SdS y elementos del SdS, aprovechando CS existentes para desarrollar las capacidades. La planificación debe reconocer que el CS mantiene su independencia de gestión y operativa respecto del SdS y de la organización del SdS. La planificación del SdS debe definir las funciones, responsabilidades y autoridades de la organización del SdS (si las hubiera), así como de las organizaciones de los CS en los planes del SdS.

5.2. Proceso de evaluación y control de proyectos

En el contexto de los SdS, la evaluación es compleja y, a veces, el control es imposible. Las habilidades de coordinación, colaboración e influencia pueden ser especialmente importantes. En algunos SdS, no siempre es posible realizar revisiones técnicas de algunos CS. Sin embargo, las revisiones indirectas (es decir, sin la presencia del responsable del CS) pueden aportar información.

Siempre que sea posible, las medidas correctivas deben acordarse entre las autoridades rectoras y gestoras de cada CS y decidirse y ejecutarse de forma colaborativa, en lugar de ordenarse, como es habitual en los sistemas tradicionales. Se puede considerar el uso de incentivos o sanciones para algunos tipos de SdS. Los planes del

SdS deben ajustarse en función del estado actual y previsto de los CS.

5.3. Proceso de gestión de decisiones

Al igual que con los sistemas, las decisiones pueden afectar al sistema en su conjunto o solo a sus elementos. En el caso del SdS, las decisiones pueden afectar tanto al SdS como a los CS, por lo que a estos últimos les puede interesar participar en los procesos de decisión y análisis junto con el SdS. Esto no siempre es posible, ya que algunos CS desconocen su participación en un SdS. La independencia de gestión de los CS implica que estos gestionen sus propias decisiones, algunas de las cuales pueden ser contrarias a los intereses del SdS.

5.4. Proceso de gestión de riesgos

Los riesgos que enfrenta el SdS difieren de los que enfrenta cada CS. Los riesgos para el SdS pueden derivar de CS que cambian, se incorporan o abandonan el SdS, o que no operan de forma coherente con las expectativas del SdS. El análisis de riesgos del SdS puede requerir un análisis de la red de CS interoperables, así como de los CS candidatos a reemplazo en caso de que algunos CS cambien o se retiren inesperadamente. Dado que cada CS mantiene su independencia operativa y de gestión, se pueden considerar incentivos para facilitar el tratamiento adecuado de los riesgos del SdS. La aplicación del tratamiento de riesgos por parte de los CS no siempre es viable.

5.5. Proceso de gestión de la configuración

La gestión de la configuración del SdS se centra en los elementos del sistema que se relacionan específicamente con el SdS. En cambio, cada CS conserva la gestión de la configuración de sus sistemas. Dado que las funciones del SdS se asignan a los CS, las interfaces entre cada CS deben estar bien definidas, especialmente cuando el SdS no las controla. En el contexto del SdS, «controlado» significa monitorizado y registrado, no que se ejerza control sobre la aprobación, el rechazo o la implementación del cambio en sí.

Las bases para el SdS dependen de los acuerdos existentes, si los hay. Si bien la cooperación de cada CS es útil, no siempre es posible, especialmente en SdS con una organización flexible. Es posible establecer una base o definir la compatibilidad entre elementos para mapear las características del sistema. Los cambios en los elementos bajo gestión de configuración pueden ser controlados o no desde la perspectiva de otro CS o del SdS.

Algunos aspectos y atributos de los CS pueden estar relacionados con la certificación de interoperabilidad. Estos atributos pueden aprovecharse para establecer una línea base desde la perspectiva del SdS. Las especificaciones de interfaz pueden formar parte de esta línea base. La compatibilidad debe considerarse cuando las especificaciones cambian. Dependiendo de la independencia de gobernanza del SdS, las versiones y entregas formales de las capacidades del SdS pueden ser inviábiles. En cambio, pueden ocurrir cuando se lanzan y entregan las capacidades de los CS.

5.6. Proceso de gestión de la información

Los tipos de información que se deben gestionar pueden estar relacionados con el propio SdS, su gobernanza y los CS. La gestión de la información del SdS debe estar claramente definida y, siempre que sea posible, se deben establecer acuerdos con los CS para compartir la información necesaria. Cabe destacar que, en el caso de algunos tipos de SdS, los CS pueden desconocer o no estar dispuestos a compartir información. El SdS (y otros CS) debe respetar la confidencialidad, la seguridad y la titularidad de la propiedad intelectual de los CS, especialmente cuando los acuerdos son informales o inexistentes. Cierta información del CS no siempre está disponible para las partes interesadas del SdS u otras partes interesadas de los CS.

5.7. Proceso de medición

Se deben identificar las necesidades de información tanto del SdS como de los CS para su participación en el SdS. La definición de estas medidas debe ir acompañada de una descripción del impacto en el rendimiento o la capacidad de cada CS, si corresponde, y del SdS en general. Para algunos tipos de SdS, la recopilación, verificación y almacenamiento de datos no siempre es posible. Cuando estén disponibles, se deben definir enfoques alternativos. También se pueden añadir elementos adicionales del SdS para facilitar la supervisión del SdS. Dependiendo de la independencia de gobernanza del SdS, la información proporcionada por los CS puede ser sesgada, errónea o inexistente.

5.8. Proceso de aseguramiento de calidad

Idealmente, los criterios y métodos para las evaluaciones de aseguramiento de la calidad deberían acordarse. Al igual que con los sistemas y subsistemas en sistemas tradicionales, el acuerdo debe identificar las funciones y responsabilidades, así como la información, los datos y los supuestos relativos a las evaluaciones planificadas. En

algunos casos, las actividades de aseguramiento de la calidad pueden ser gestionadas por el SdS, o por separado por cada CS, sin acuerdo entre ellos. En el caso de algunos tipos de SdS, los CS pueden desconocer su participación. De ser posible, el tratamiento de los problemas y la definición de prioridades deben acordarse entre las autoridades rectoras y gestoras de cada CS y el SdS.

6. PROCESOS TÉCNICOS

6.1. Proceso de análisis de negocio o misión

Dado que los problemas y oportunidades que aborda el SdS suelen ser más amplios y complejos que los que abordan los sistemas, el análisis de negocio o de misión puede requerir el uso e integración de técnicas y herramientas de modelado más heterogéneas que las de los sistemas. Idealmente, la caracterización de un espacio de soluciones para un SdS debería incluir el CS candidato, cómo cada CS respalda el nuevo problema u oportunidad y cualquier restricción que estos CS puedan imponer al espacio de soluciones.

Dado que los CS ya desplegados cuando se concibe el SdS presentan consideraciones y limitaciones operativas, el SdS puede verse afectado y limitado. Los enfoques sistémicos que asumen la capacidad de cambiar los CS probablemente no sean eficaces. Dependiendo del grado de independencia de gestión, pueden ser necesarios diferentes enfoques. Para el SdS, pueden ser viables múltiples soluciones alternativas.

6.2. Proceso de definición de necesidades y requisitos de las partes interesadas

El proceso de definición de las necesidades y requisitos de las partes interesadas para el SdS se centra en el mismo enfoque que para los sistemas, pero la identificación y el acceso a las partes interesadas pueden verse limitados. No siempre es posible identificar a todas las partes interesadas; la obtención de sus necesidades probablemente será incompleta y quizás incorrecta. El SdS debe planificarse para incorporar a las partes interesadas emergentes y sus necesidades, así como cualquier restricción que pueda afectar su funcionamiento y evolución. En el SdS, cada CS mantiene su independencia operativa y de gestión, así como su interdependencia. Es importante reconocer dónde

las necesidades clave de las partes interesadas de los CS se alinean o entran en conflicto con los objetivos del SdS. Para tener éxito, el SdS debe evitar infringir los objetivos de los CS.

Los CS tienen sus propios conceptos operativos y ciclos de vida que evolucionan de forma independiente, por lo que es importante documentar la situación de cada sistema dentro de su ciclo de vida. Puede haber diferentes periodos de estabilidad para algunos CS e inestabilidad para otros. Es importante identificar las limitaciones de los CS en el SdS.

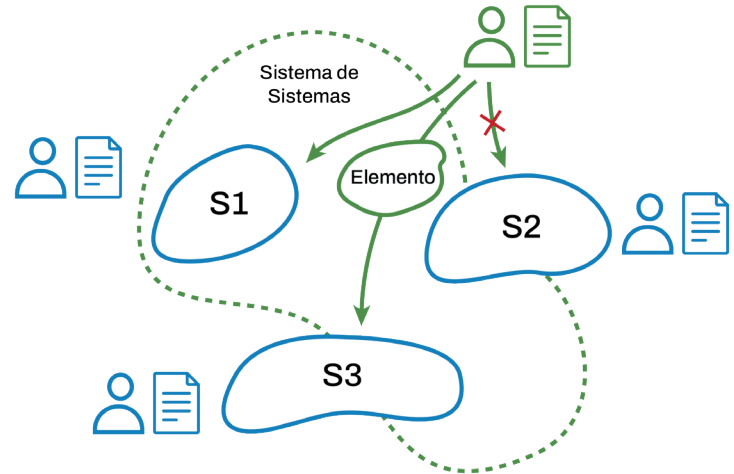


Figura 3. Cada CS tendrá sus propias partes interesadas con sus necesidades.

Las partes interesadas en el SdS y los CS deben esforzarse por alcanzar un acuerdo siempre que sea posible, pero no es necesario que todos los interesados de los CS estén de acuerdo para una definición eficaz del SdS. Sin embargo, se debe tener cuidado, ya que un desacuerdo puede provocar que algún CS abandone el SdS o dificultar los objetivos del SdS.

6.3. Proceso de definición de requisitos del sistema

Al igual que con los sistemas, el proceso de definición de requisitos del SdS transforma los resultados deseados por las partes interesadas en requisitos, características, atributos, funciones y rendimiento que el SdS debe poseer para satisfacer los requisitos de las partes interesadas. Para algunos tipos de SdS, los propietarios de los CS pueden dirigir o apoyar la definición de requisitos del SdS. Para otros tipos de SdS, el propietario del CS puede ser

reacio a divulgar esta información. Al igual que con los sistemas, los requisitos del SdS y las restricciones de diseño pueden cambiar con el tiempo. A medida que los CS cambian, se incorporan o abandonan el SdS, el conjunto de capacidades disponibles cambia, lo que afecta lo que es factible que el SdS proporcione. Requisitos que se cumplían previamente pueden dejar de cumplirse sin previo aviso.

Dado que los CS mantienen su independencia de gestión y operativa, no siempre aceptan ni implementan los requisitos del SdS. En consecuencia, los SdS a veces no pueden cumplir sus objetivos. Sin embargo, la definición de requisitos debe considerar medidas críticas de rendimiento para los CS y el SdS a fin de facilitar la implementación de las capacidades del SdS en diversas situaciones. Al igual que con los sistemas, los requisitos de las partes interesadas del SdS pueden vincularse a los objetivos de capacidad del SdS y a los requisitos de los CS o de los elementos del sistema. Desafortunadamente, los requisitos del SdS pueden entrar en conflicto con los atributos de los CS existentes.

6.4. Proceso de definición de la arquitectura del sistema

Al igual que con los sistemas, la definición de la arquitectura del SdS se centra en la funcionalidad crítica de los elementos. En SdS, estos incluyen los CS, los elementos del sistema y sus interacciones. Conocer la arquitectura interna de cada CS puede ser innecesario, o incluso imposible. Los sistemas suelen descomponerse en términos de jerarquías; los SdS suelen describirse como redes con interconexiones complejas. En consecuencia, la arquitectura del SdS puede verse limitada por información parcial de los CS, algunos de los cuales pueden desconocer su participación en el SdS.

La arquitectura del SdS debe prever la necesidad de definir elementos del sistema que no forman parte de ningún CS. Asimismo, los CS cambian, se incorporan y abandonan el SdS con el tiempo, de modo que la definición de la arquitectura puede mantenerse flexible. Al igual que con los sistemas, las arquitecturas del SdS deben reflejar las necesidades de las partes interesadas del SdS en cuanto a las capacidades del SdS. Sin embargo, también es importante reconocer en las arquitecturas las necesidades de las partes interesadas de los CS que pueden verse afectadas por la participación de los CS en el SdS. Idealmente, las arquitecturas de SdS deberían abordar el SdS, minimizando o evitando cualquier impacto negativo en los CS.

6.5. Proceso de definición del diseño

Al igual que con los sistemas, se requiere un diseño integral para el SdS. Sin embargo, a diferencia de los sistemas, los elementos de un SdS pueden ser propiedad de otras organizaciones y operados para sus propios fines. No obstante, los enfoques de sistema, como los de la norma ISO/IEC/IEEE 15288, pueden aplicarse a elementos específicos del sistema que no forman parte de ningún CS. Pueden ser necesarias múltiples definiciones de diseño para implementar cambios en el SdS de forma gradual, especialmente cuando los CS se incorporan o abandonan el SdS inesperadamente.

La definición del diseño del SdS debe reconocer que los cambios deseados en los CS pueden no ser posibles y que el diseño y las capacidades de los CS pueden tener que aceptarse y adaptarse tal como están. En la medida de lo posible, la definición del diseño del SdS debe considerar la implementación gradual de las implementaciones de los CS para que sus capacidades sean compatibles entre sí en todo momento.

Para el SdS, la definición del diseño enfatiza la selección y adaptación de los CS u otros elementos del sistema necesarios para facilitar la interacción de los CS en el SdS. Se debe evaluar la viabilidad de los CS alternativos, no sólo desde el punto de vista técnico, sino también en cuanto a su independencia operativa y de gestión. Un participante dispuesto puede ser mejor que un participante técnicamente superior, pero no dispuesto.

La selección del conjunto de CS para satisfacer las necesidades del SdS implica asignar sistemas, no solo elementos del sistema, a los requisitos del SdS. El proceso de definición del diseño del SdS se centra en evaluar la capacidad de los CS y los elementos del sistema, actuales o propuestos, para implementar las interfaces necesarias para satisfacer las necesidades del SdS.

Las organizaciones que gestionan CS pueden ser reacias a divulgar toda la información de diseño. Afortunadamente, los SdS no necesitan comprender los aspectos internos de los diseños de los CS, sino que pueden centrarse más en las interfaces y las características externas de los CS. Es decir, en la definición del diseño del SdS es importante contar con la información de diseño de CS suficiente para comprender su comportamiento, pero sin acceso a los detalles de cómo lo logra.

6.6. Proceso de análisis del sistema

Al igual que los sistemas, el análisis de Sistemas de Sistemas puede aprovechar datos reales y previstos. Dado que muchos CS ya existen antes que el SdS, se puede disponer de datos de su operación, aunque los propietarios de los CS no estén dispuestos a compartir esta información. Por esta razón, pueden ser necesarios datos predictivos separados de los CS.

Es posible que cada propietario de un CS pueda realizar el proceso de análisis del sistema para respaldar la gestión de decisiones a nivel de CS con impacto en el SdS. Sin embargo, los propietarios de CS pueden tener puntos de vista e intereses diferentes al del SdS, por lo que sus análisis y resultados podrían diferir de los del SdS. Desafortunadamente, las decisiones de los CS basadas en sus propios análisis pueden entrar en conflicto con los intereses y requisitos del SdS. La resolución de conflictos puede ser muy difícil o imposible.

Los análisis de SdS deben explorar y validar los supuestos y resultados relacionados con la independencia operativa y de gestión de los CS. El SdS debe considerar la posibilidad de añadir puntos de medición e instrumentación para proporcionar los datos necesarios para validar los análisis o modelos.

6.7. Proceso de implementación

Los SdS se implementan típicamente mediante la composición de CS existentes y potencialmente modificados, junto con otros elementos del sistema para proporcionar nuevas capacidades. Las limitaciones de los CS existentes influyen en los requisitos, la arquitectura o el diseño. Al igual que los sistemas, los elementos restantes del sistema deben implementarse y se puede utilizar el proceso de implementación de la norma ISO/IEC/IEEE 15288.

La gestión de la información de los SdS y la infraestructura de soporte pueden permitir que el SdS prospere y se desarrolle según sea necesario. La planificación temporal de los cambios en los CS debe planificarse cuidadosamente. Idealmente, se espera que cada CS continúe respaldando sus propias capacidades independientes y que cualquier cambio en un CS esté listo para respaldar la capacidad deseada del SdS. En la medida de lo posible, se debe tener cuidado para confirmar que las interacciones y capacidades deseadas no se vean afectadas por los cambios en los CS.

6.8. Proceso de integración

Los SdS suelen consistir en CS existentes que se integran para cumplir con los requisitos del SdS. Las restricciones de integración que influyen en los requisitos, la arquitectura o el diseño del SdS, incluidas las interfaces, pueden ser considerables. A diferencia de los sistemas, la integración del SdS suele realizarse con CS operativos en un entorno de SdS operativo. La integración del SdS suele ocurrir en el contexto de un SdS en constante evolución y operación. Detener la operación puede resultar imposible o poco práctico. Por lo tanto, la planificación de la integración debe facilitar la interacción con las operaciones en curso de los CS y del SdS.

Dado que los CS pueden cambiar, unirse o abandonar un SdS, se puede esperar una integración frecuente. La comprobación frecuente de las interfaces entre los CS y los elementos del sistema implementados puede ser importante.

6.9. Proceso de verificación

La verificación de los requisitos del SdS no suele ser posible con el mismo grado de certeza y fidelidad que con otros tipos de sistemas. Sin embargo, las actividades de verificación pueden ser valiosas para evaluar el grado de cumplimiento del SdS con los requisitos, siempre que estos se conozcan y comprendan. A diferencia de muchos sistemas tradicionales, la verificación del SdS casi siempre se realiza en el contexto de operaciones continuas y cambiantes. La verificación debe planificarse y ejecutarse teniendo en cuenta las operaciones en curso.

Para algunos SdS, los propietarios de CS pueden realizar alguna verificación de sus vistas parciales del SdS. Para otros, puede ser imposible probar algunas capacidades del SdS debido a cuestiones de seguridad o costo. La verificación del SdS puede basarse en modelado o análisis. A diferencia de los sistemas, la verificación de algunos CS no está garantizada. La verificación completa del SdS y la resolución de anomalías pueden ser inviables.

6.10. Proceso de transición

Al igual que con los sistemas, el proceso de transición establece las capacidades deseadas en un entorno operativo. Para algunos CS, las nuevas funciones y capacidades pueden estar disponibles con frecuencia para las partes interesadas, mientras que otros CS realizan transiciones a un ritmo mucho más lento. En consecuencia, las transiciones de los CS pueden ocurrir con la misma frecuencia para el SdS. Algunos propietarios del SdS pueden influir en la fase de los cambios de los CS, mientras que otros CS pueden no estar dispuestos a adaptarse a las necesidades del SdS. En la medida de lo posible, las transiciones deben planificarse y ejecutarse para coordinar las capacidades de los CS y del SdS.

Los CS suelen estar en funcionamiento antes de que la capacidad de SdS se transfiera a operaciones. Es posible que un CS se una y abandone un SdS inesperadamente. Por lo tanto, no se garantiza la entrega de las capacidades de SdS. La gobernanza de SdS puede considerar incentivar la creación de CS en ciertas áreas.

6.11. Proceso de validación

La validación del SdS puede ser un proceso continuo durante las operaciones, en lugar de un evento aislado, como suele ocurrir con los sistemas. El acceso a evidencia sobre algunos objetivos del SdS o requisitos de las partes interesadas puede ser limitado o incluso inexistente debido a consideraciones de seguridad, protección u operativas. En tales casos, la planificación de la validación del SdS debe considerar la disponibilidad de evidencia subjetiva o complementaria.

Al igual que con la validación de sistemas tradicionales, la validación del SdS permite identificar discrepancias entre las necesidades originales y las capacidades operativas disponibles. Las partes interesadas del SdS, así como las partes interesadas de los CS, pueden tener objetivos contradictorios. Por lo tanto, a diferencia de los sistemas, puede resultar imposible obtener la plena ratificación de la validación del SdS por parte de las partes interesadas. En algunos SdS, los propietarios de los CS pueden validar una vista parcial del SdS disponible para cada propietario de los CS, mientras que otros pueden no estar dispuestos a respaldar la validación del SdS. En muchos SdS, la disponibilidad del servicio no está garantizada.

Curiosamente, la validación de los CS puede ser irrelevante o innecesaria si la parte interesada valida el SdS. Sin embargo, un CS no validado y fuertemente independiente puede rechazar o abandonar el SdS, lo que supone un riesgo para la disponibilidad de los servicios y capacidades del SdS.

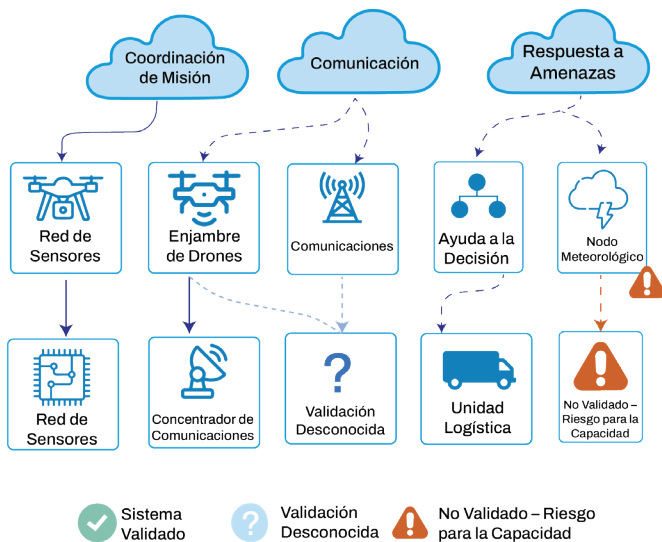


Figura 4. Algunos CS pueden validar vistas parciales del SdS, pero los CS no validados plantean riesgos para la disponibilidad del servicio.

6.12. Proceso de operación

Los SdS suelen ensamblarse u orquestarse a partir de los CS existentes, que tienen sus propias prioridades y limitaciones operativas. En ocasiones, no es posible incorporar estas prioridades y limitaciones a los demás procesos de manera oportuna. Los cambios operativos en los CS y sus interfaces afectan al SdS. Asimismo, la retroalimentación de las partes interesadas sobre el funcionamiento del SdS puede sugerir la necesidad de realizar cambios en los CS, pero estos pueden mostrarse reacios a hacerlo porque tienen intereses operativos propios, que pueden diferir de los del SdS.

6.13. Proceso de mantenimiento

El mantenimiento del SdS es particularmente complejo, ya que las organizaciones que poseen y gestionan los CS pueden modificarlos para sus propios fines, lo que podría afectar al SdS y a las interfaces con otros CS. Estos cambios pueden ocurrir sin previo aviso ni coordinación dentro del SdS. En algunos tipos de SdS, las organizaciones que gestionan los CS pueden ser reacias a informar sobre los cambios pendientes para abordar el mantenimiento correctivo, perfectivo o adaptativo. Asimismo, estas organizaciones pueden ser reacias a compartir datos relacionados con fallos. La planificación del SdS puede verse limitada por la falta de acceso a los datos que normalmente estarían disponibles en un sistema. Por consiguiente, la capacidad de un SdS para mantenerse a sí mismo puede requerir una planificación de contingencia proactiva.

6.14. Proceso de retirada

Dado que los CS dentro del SdS son gobernados y gestionados de forma independiente, es posible, y quizás probable, que un CS pueda abandonar el SdS sin mucha coordinación ni planificación. Una organización puede decidir deshacerse de un CS, lo que perjudica a cualquier SdS en el que participaba. En algunos tipos de SdS, el CS puede desconocer la participación. Por el contrario, un SdS puede retirarse y los resultados del proceso de eliminación se pueden lograr sin afectar negativamente al CS, que puede seguir operando para sus propios fines. A diferencia de los sistemas que no son SdS, el SdS puede simplemente desconectarse de un CS en lugar de deshacerse de éste. Sin embargo, si el SdS cuenta con sistemas, servicios o elementos de sistema habilitadores exclusivos del SdS que no son necesarios para ningún CS, se debe seguir el proceso de eliminación de la norma ISO/IEC/IEEE 15288, cláusula 6.4.14, y se deben lograr los resultados del proceso de retirada.

7. CONCLUSIONES

Aplicar la ingeniería de sistemas a SdS no es tratar únicamente de adaptar los procesos existentes, sino de adoptar una forma de pensar fundamentalmente diferente. En lugar de centrarse en el control, la predictibilidad y la optimización, la ingeniería de SdS se centra en la influencia, la adaptación y la resiliencia. Esto requiere un cambio en varias áreas:

- 1) El rol de los ingenieros de sistemas debe evolucionar de diseñador a orquestador.
- 2) La influencia debe reemplazar el control directo como el medio principal para garantizar la alineación del sistema.
- 3) La interoperabilidad debe convertirse en un requisito fundamental, no en una ocurrencia tardía.
- 4) El comportamiento emergente se anticipa y se gestiona en lugar de evitarse.
- 5) La verificación y validación cambia de pruebas estáticas, basadas en requisitos, a la evaluación continua en contextos del mundo real. Sin embargo, también es importante tener en cuenta que el desarrollo de enfoques y métodos específicos para respaldar este cambio está en sus primeras etapas.

El cambio es necesario, pero debe llevarse a cabo con cautela¹.

¹ La cautela no significa apegarse a las prácticas tradicionales de ingeniería de sistemas. En este caso, la cautela implica tener especial cuidado al abordar la ingeniería de SdS; un SdS no es simplemente un sistema más grande...

REFERENCIAS

1. McDermott, T. and V. Ramos, New Kinds of Systems, in Introduction to Systems Engineering in the 21st Century, A. Salado, Editor. 2024, Isdefe: Madrid, Spain. p. 35-55.
2. Salado, A., A. Sancehz, and D. Verma, Systems Engineering in the 21st Century, in Introduction to Systems Engineering in the 21st Century, A. Salado, Editor. 2024, Isdefe S.A.: Madrid, Spain. p. 15-33.
3. ISO, ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes. ISO/IEC/IEEE 15288 First edition 2015-05-15, 2015: p. 1-118.
4. INCOSE, Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities. 5th ed. 2023, Hoboken, NJ, USA: John Wiley and Sons, Inc.
5. ISO/IEEE/IEC, ISO/IEC/IEEE 21839, Systems and software engineering — System of systems (SoS) considerations in life cycle stages of a system. 2019.
6. ISO/IEEE/IEC, ISO/IEC/IEEE 21840, Systems and software engineering — Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems (SoS). 2019.
7. ISO/IEEE/IEC, ISO/IEC/IEEE 21841, Systems and software engineering — Taxonomy of systems of systems. 2019.

APÉNDICE: COMPARACIÓN ENTRE PRÁCTICAS DE INGENIERÍA DE SISTEMAS PARA SISTEMAS Y SDS

	Procesos	Sistema	SdS
Procesos de acuerdo	Adquisición	Un único comprador gestiona el sistema y contrata a los proveedores para su desarrollo y operación, con acuerdos formales.	Puede que no haya un único comprador; El CS puede permanecer independiente desde el punto de vista operativo y de gestión, con acuerdos que pueden ser formales, informales o incluso inexistentes.
	Suministro	El adquirente tiene pleno control administrativo y operativo sobre el sistema y sus proveedores.	El CS puede mantener su independencia con sus propios compradores. La adaptación a las necesidades del SdS varía, y los acuerdos pueden ir desde contratos formales hasta la ausencia de ellos. Los proveedores no tienen obligaciones directas con el SdS a menos que se acuerde expresamente.
Procesos facilitadores de proyectos organizacionales	Gestión del modelo de ciclo de vida	Sigue un modelo de ciclo de vida bien definido gestionado por el adquirente. Cuando se utilizan distintos modelos de ciclo de vida en diferentes partes del sistema siguen integrados a nivel de sistema.	Cada CS tiene su propio modelo de ciclo de vida y puede estar en diferentes etapas.
	Gestión de infraestructura	Los requisitos de infraestructura se definen para un solo sistema y sus necesidades.	Puede requerir infraestructura adicional, como laboratorios de integración y pruebas, para garantizar la interoperabilidad. La disponibilidad y la resiliencia son cruciales debido a las posibles variaciones en las etapas del ciclo de vida de los CS.
	Gestión de cartera	La gestión de cartera es más sencilla, ya que la organización tiene control total sobre su sistema y sus recursos.	La gestión de cartera puede ser más compleja debido a la falta de control sobre CS independientes, lo que puede requerir recursos y presupuestos adicionales.
	Gestión de recursos humanos	Los recursos humanos se centran en actividades y procesos específicos del sistema, con habilidades adaptadas a la gestión del sistema en sí.	Los recursos humanos en el SdS pueden requerir habilidades diferentes, particularmente relacionadas con la influencia en lugar del control directo, debido a la independencia de gestión de los CS.
	Gestión de calidad	Se establece y gestiona un enfoque de gestión de calidad para el propio sistema.	El SdS depende de la gestión de calidad de los CS, que puede variar considerablemente. La evaluación de la calidad del SdS se centra en las capacidades integradas, no en las capacidades individuales de cada CS.
	Gestión del conocimiento	Las taxonomías y los activos de conocimiento se definen para el propio sistema.	El SdS define sus propios activos de conocimiento, que deben perdurar durante las transiciones de CS. La colaboración entre CS o con el SdS no está garantizada, y el conocimiento se distribuye entre diferentes organizaciones.

Procesos de Gestión Técnica	Planificación de proyectos	Los proyectos se crean para cumplir objetivos específicos del sistema y se gestionan completamente dentro del control del sistema.	Los proyectos abordan las deficiencias de capacidad aprovechando CS existentes. La planificación debe considerar la independencia de los CS y definir las funciones, responsabilidades y autoridades tanto de las organizaciones del SdS como de los CS.
	Evaluación y control de proyectos	La evaluación y el control son sencillos y se gestionan directamente dentro del sistema.	La evaluación es compleja y tiene un control limitado. El éxito depende de la coordinación, la influencia y la colaboración. Las acciones correctivas son colaborativas y los planes deben adaptarse a la evolución del estado de la seguridad informática.
	Gestión de decisiones	Las decisiones impactan al sistema en su totalidad o a elementos específicos, con control total sobre las decisiones del sistema.	Las decisiones afectan tanto al SdS como a los CS. Es posible que los CS no siempre sean conscientes de su papel, y sus decisiones independientes pueden entrar en conflicto con los objetivos del SdS.
	Gestión de riesgos	Los riesgos están contenidos dentro del sistema y se gestionan directamente.	Los riesgos surgen de cambios, desviaciones o inconsistencias en los CS. El análisis de riesgos implica comprender la red de CS interoperables y sus posibles reemplazos. Implementar un tratamiento de riesgos podría no ser viable.
	Gestión de configuración	La gestión de la configuración se realiza internamente dentro del sistema, centrándose en sus elementos.	La gestión de la configuración se centra en los elementos relacionados con el SdS y los CS conservan el control de sus propios sistemas. Las especificaciones de interfaz y la compatibilidad son cruciales y los cambios pueden no estar controlados desde la perspectiva del SdS.
	Gestión de la información	La gestión de la información se centra en el sistema y su gobernanza interna.	La gestión de la información involucra tanto al SdS como a los CS, lo que podría requerir acuerdos claros para compartirla. Los CS pueden no estar dispuestos o desconocer la necesidad de compartir información y se deben respetar la confidencialidad, la seguridad y la propiedad intelectual. Es posible que alguna información no sea accesible para todas las partes interesadas.
	Medición	Las necesidades de información están claramente definidas dentro del sistema para su propio desempeño.	El SdS necesita definir y gestionar información tanto en el SdS como en los CS, pero los datos pueden ser incompletos, sesgados o no estar disponibles.
	Aseguramiento de la calidad	Los criterios y métodos de garantía de calidad están claramente definidos y acordados.	El control de calidad puede ser gestionado por el SdS o de forma independiente por cada CS, a veces sin acuerdos formales. Es posible que los CS desconozcan su participación y las prioridades y el tratamiento de los problemas deben acordarse entre las autoridades de los CS y del SdS.

Procesos técnicos	Análisis de negocio o misión	Los problemas suelen ser más simples y el análisis utiliza herramientas de modelado más homogéneas.	Los problemas son más amplios y complejos y requieren diversas herramientas de modelado e integración. Las limitaciones de los CS y las consideraciones operativas afectan el SdS y múltiples soluciones alternativas pueden ser viables.
	Definición de las necesidades y requisitos de las partes interesadas	Las necesidades y requisitos de las partes interesadas se identifican y abordan con un enfoque claro.	La identificación y el acceso a todas las partes interesadas del SdS pueden verse limitados y las necesidades pueden ser incompletas o incorrectas. Es necesario gestionar las partes interesadas emergentes y los conflictos con los objetivos de los CS.
	Definición de requisitos del sistema	Los requisitos se definen claramente para cumplir con los resultados de las partes interesadas, con características, funciones y desempeño estables.	Los requisitos están sujetos a cambios a medida que los CS se incorporan, se retiran o cambian, lo que afecta la viabilidad. Los CS podrían no aceptar o implementar los requisitos del SdS y pueden surgir conflictos entre los requisitos del SdS y de los CS.
	Definición de la arquitectura del sistema	La arquitectura del sistema suele ser jerárquica y se centra en la funcionalidad crítica y los componentes internos.	La arquitectura se basa en redes con interconexiones complejas entre CS y los elementos del sistema y puede estar limitada por información parcial de los CS.
	Definición del diseño	Los sistemas requieren un diseño integral que integre todos los componentes y elementos dentro de un alcance definido.	El diseño debe considerar los CS que pertenecen y son operados por diferentes organizaciones con fines independientes. El diseño debe considerar los cambios de forma gradual, reconociendo que el diseño y las capacidades de los CS no pueden modificarse. El diseño del SdS se centra en la selección y adaptación de los CS para garantizar su compatibilidad y funcionalidad.
	Análisis del sistema	El análisis del sistema utiliza datos reales y predictivos para evaluar y optimizar el rendimiento del sistema.	El análisis del SdS utiliza datos reales y predictivos, pero los CS podrían no estar dispuestos a compartir datos reales. El análisis del SdS debe considerar la independencia operativa y de gestión de los CS añadiendo puntos de medición para validar suposiciones y resultados.
	Implementación	Los sistemas normalmente se implementan mediante el desarrollo y la integración de nuevos elementos del sistema según los requisitos y diseños.	Se implementa mediante la composición de CS existentes, potencialmente modificados, junto con otros elementos del sistema para crear nuevas capacidades. Las limitaciones de los CS existentes afectan los requisitos, la arquitectura y el diseño del SdS.
	Integración	La integración normalmente se realiza en un entorno controlado, donde se integran elementos del sistema nuevos o modificados según una arquitectura y un diseño definidos.	La integración implica la integración de los CS existentes para cumplir con los requisitos del SdS, a menudo dentro de un entorno operativo. La integración ocurre dentro de un SdS en funcionamiento continuo y puede resultar impráctico detener las operaciones.

	Verificación	La verificación normalmente se realiza en un entorno controlado con un alto grado de certeza, lo que confirma que el sistema cumple con sus requisitos.	La verificación es más compleja e incierta y suele ocurrir en un entorno operativo en constante evolución. Es posible que no sea posible verificar completamente algunas capacidades debido a limitaciones de seguridad o de costo. La verificación suele depender del modelado, del análisis y de las pruebas parciales de los CS.
	Transición	La transición implica trasladar un sistema del entorno de desarrollo a uno operativo, con un enfoque claro en garantizar que cumpla con las capacidades deseadas.	La transición en SdS implica la coordinación de múltiples CS, que pueden tener diferentes cadencias de transición. La gobernanza del SdS puede incentivar a los CS a alinearse con los objetivos de SdS, pero las transiciones no están garantizadas. Se requiere planificación y coordinación frecuentes.
	Validación	La validación suele ser un conjunto de eventos que garantizan que el sistema cumpla con los requisitos y objetivos previstos. Implica la recopilación de evidencia objetiva para confirmar las capacidades del sistema.	La validación es un proceso continuo, influenciado por el contexto operativo. El acceso a la evidencia puede estar restringido y se pueden utilizar evidencias subjetivas o auxiliares. Pueden surgir discrepancias entre los requisitos y las capacidades operativas. La ratificación total de las partes interesadas puede que sea imposible debido a objetivos contradictorios entre las partes interesadas de SdS y CS.
	Operación	Los cambios operativos normalmente siguen un proceso planificado y centralizado que afecta al sistema en su totalidad.	Los CS tienen sus propias prioridades y limitaciones operativas. Los cambios operativos en los CS y sus interfaces impactan en el SdS. Los CS pueden resistirse a los cambios sugeridos por el SdS debido a diferentes intereses operativos.
	Mantenimiento	El mantenimiento normalmente se planifica y controla dentro de un entorno cerrado, con acceso directo a los datos.	El mantenimiento es complejo debido a la independencia de los CS, que puede realizar cambios sin coordinación ni notificación, lo que podría afectar al SdS. Es necesario contar con una planificación de contingencia proactiva.
	Retirada	La retirada se gestiona dentro de un entorno controlado y los sistemas normalmente se retiran o desmantelan según procedimientos establecidos.	Un CS puede salir del SdS sin coordinación, mientras que el SdS puede simplemente desconectarse del CS sin deshacerse de él.

DR. MIKE YOKELL

El Doctor Mike Yokell es un líder en ingeniería de sistemas retirado tras más de 40 años de experiencia en la industria aeroespacial y de defensa estadounidense. Es editor, coeditor y colaborador de numerosos estándares internacionales sobre ingeniería de sistemas y software. Mike está certificado como experto en ingeniería de sistemas por INCOSE.



Posee múltiples patentes estadounidenses y europeas de ingeniería de sistemas basada en modelos y realidad virtual inmersiva a gran escala. Contribuye regularmente a los grupos de trabajo de INCOSE sobre Agile y Sistemas de Sistemas.

DR. ALEJANDRO SALADO

El Dr. Alejandro Salado es profesor titular de ingeniería de sistemas en el Departamento de Ingeniería de Sistemas e Ingeniería Industrial de la Universidad de Arizona y director del programa de ingeniería de sistemas. Además, ofrece consultoría a tiempo parcial en áreas relacionadas con la transformación empresarial, el cambio cultural de los equipos técnicos, la ingeniería de sistemas y la estrategia de ingeniería. Alejandro investiga en formulación de problemas, diseño de estrategias de verificación y validación, ingeniería de sistemas basada en modelos y formación en ingeniería. Antes de incorporarse al mundo académico, ocupó puestos de ingeniero de sistemas, arquitecto jefe e ingeniero jefe de sistemas en sistemas espaciales tripulados y no tripulados con un coste de desarrollo de hasta 1.000 millones de dólares. Ha publicado más de 150 artículos técnicos y su investigación ha recibido financiación federal de la Fundación Nacional de Ciencias (NSF), el Comando de Guerra de Superficie Naval (NSWC), el Comando del Sistema Aéreo Naval (NAVAIR) y la Oficina de Investigación Naval (ONR), entre otros. Ha recibido el Premio NSF CAREER y el Premio Internacional Fulbright de Ciencia y Tecnología. El Dr. Salado es ingeniero de telecomunicación por la Universidad Politécnica de Valencia y tiene un máster en gestión de proyectos y un máster en ingeniería electrónica de la Universidad Politécnica de Cataluña, el máster SpaceTech MEng en ingeniería de sistemas espaciales de la Universidad Técnica de Delft y un doctorado en ingeniería de sistemas del Stevens Institute of Technology.



ADOLFO SÁNCHEZ DOMÍNGUEZ

Adolfo Sánchez es ingeniero en Informática por la Universidad de Zaragoza y licenciado en Historia por la UNED. Ha participado en la definición y el desarrollo de un evaluador de arquitectura de aviónica modular, el mantenimiento del software del Eurofighter, el desarrollo de software de control de calidad para programas estratégicos de guerra electrónica, y la adquisición, despliegue e implementación del Programa CIS para la Unidad Militar de Emergencias. Actualmente, es Coordinador en el Área CIS de Defensa en Isdefe, prestando apoyo a la Subdirección de Transformación Digital del Ministerio de Defensa. Es Ingeniero de Sistemas Certificado (CSEP) por INCOSE y formador interno en diversos cursos de ingeniería de sistemas en Isdefe.



Ingeniería de Misión

Dr. James Moreland JR, *Alto ejecutivo retirado del Departamento de Defensa de EE. UU. y vicepresidente de Raytheon Technologies (james.moreland.mei@gmail.com)*

Tte Coronel Víctor M. Sobrino García, *Ministerio de Defensa (victor@sobrinostre.com)*

Resumen

La Ingeniería de Misión (IM) es una metodología estructurada y centrada en la misión, diseñada para impulsar el desarrollo y la integración de Sistemas de Sistemas (SdS) complejos en los dominios operativos, considerando la misión y la experiencia operativa como el eje central de la disciplina. La IM proporciona un marco repetible y basado en datos que enlaza el rigor de la ingeniería con la perspectiva estratégica y operativa, lo que permite un desarrollo de capacidades alineado con los resultados de la misión. Mediante el uso de Hilos de Ingeniería de Misión (HIM), Redes de Efectos y marcos estructurados como DOTMLPF-P o MIRADO-I, la IM permite a los responsables de la toma de decisiones mapear, evaluar y desarrollar las interacciones entre sistemas, personas y procesos críticos para conseguir el éxito de la misión. La metodología de IM integra evaluaciones basadas en riesgos, análisis formales de integración e interoperabilidad y la generación de criterios de éxito de la misión —desde la intención estratégica hasta el rendimiento táctico—, creando una arquitectura de referencia común que garantiza el alineamiento entre las partes interesadas, los dominios y los socios de una coalición. La IM promueve un cambio desde una adquisición optimizada en base a la plataforma hacia un desarrollo basado en capacidades, lo que da lugar a arquitecturas de misión resilientes, interoperables y preparadas para el futuro que respaldan tanto la defensa como objetivos industriales más amplios.

Palabras clave

Ingeniería de Misión, Sistemas de Sistemas, Hilo de Ingeniería de Misión, Desarrollo de Capacidades, Efectividad Operacional, Redes de Efectos, Resiliencia, Integración, Interoperabilidad, DOTMLPF-P, MIRADO-I.



1. INTRODUCCIÓN

La Ingeniería de Misión (IM) es un enfoque para Sistemas de Sistemas (SdS) centrado en la misión, que requiere una sólida comprensión de las interacciones y de las dependencias existentes entre los múltiples sistemas que se requieren para ejecutar misiones operativas complejas y multidominio. Originalmente, la IM se desarrolló e implementó en el Departamento de Defensa de los Estados Unidos (DoD, por sus siglas en inglés) como una metodología orientada a lograr la eficacia de la colaboración entre SdS en base a los objetivos generales de la misión como medio para la definición del éxito de la misma. En esencia, la IM proporciona un enfoque sistemático, cuantificable y repetible que apoya la toma de decisiones estratégicas en todos los ámbitos, facilitando así la transición exitosa desde plataformas y sistemas a capacidades de SdS.

La disciplina de la IM enlaza el rigor de la ingeniería con el conocimiento operativo y de negocio necesario para identificar las soluciones y los requisitos relacionados con las capacidades y alineados con la misión de una organización global. Formalmente, la IM se ha definido como «un proceso interdisciplinar que abarca todo el esfuerzo técnico necesario para analizar, diseñar e integrar las necesidades y las capacidades operativas actuales y para conseguir los resultados de misión deseados» [1]. En la práctica, los ingenieros de misión planifican, analizan, organizan e integran conceptos operativos con el fin de desarrollar una arquitectura operativa integral y los atributos de la capacidad.

En el contexto de defensa, esto abarca el espectro de Doctrina, Organización, Entrenamiento, Material, Liderazgo y Educación, Personal, Instalaciones y Política (DOTMLPF-P) (en España, MIRADO-I, Material, Infraestructura, Recursos Humanos, Adiestramiento, Doctrina, Organización, e Integración), incluyendo el comportamiento de adversarios y competidores. A pesar de sus especificidades en cuanto al empleo de la fuerza militar en el ámbito operativo, la IM proporciona un conjunto de herramientas y métodos que permiten a las organizaciones de todos los sectores aprovechar sus beneficios.

El análisis de la IM integra datos fiables y un marco común para generar la arquitectura de un SdS y los principales atributos de la capacidad que fundamentan los requisitos y establecen las bases de referencia para una arquitectura técnica basadas en datos. Los resultados de la IM proporcionan información a las partes interesadas y a los responsables de la toma de decisiones en todo el ecosistema del sector industrial, como se muestra en la Figura 1 para el Departamento de Defensa de EE. UU., que muestra la metodología desde la definición del enunciado del problema, o de la necesidad operativa, hasta el desarrollo de posibles soluciones. Instituciones como la División Dahlgren del Centro de Guerra de Superficie Naval (NSWCDD) han sido pioneras en la implementación de la IM en EE. UU. Si bien Europa no cuenta actualmente con un equivalente exacto, iniciativas dentro de la Agencia Europea de Defensa (EDA) u organizaciones nacionales (por ejemplo, la DGAM de España) están comenzando a explorar enfoques similares.

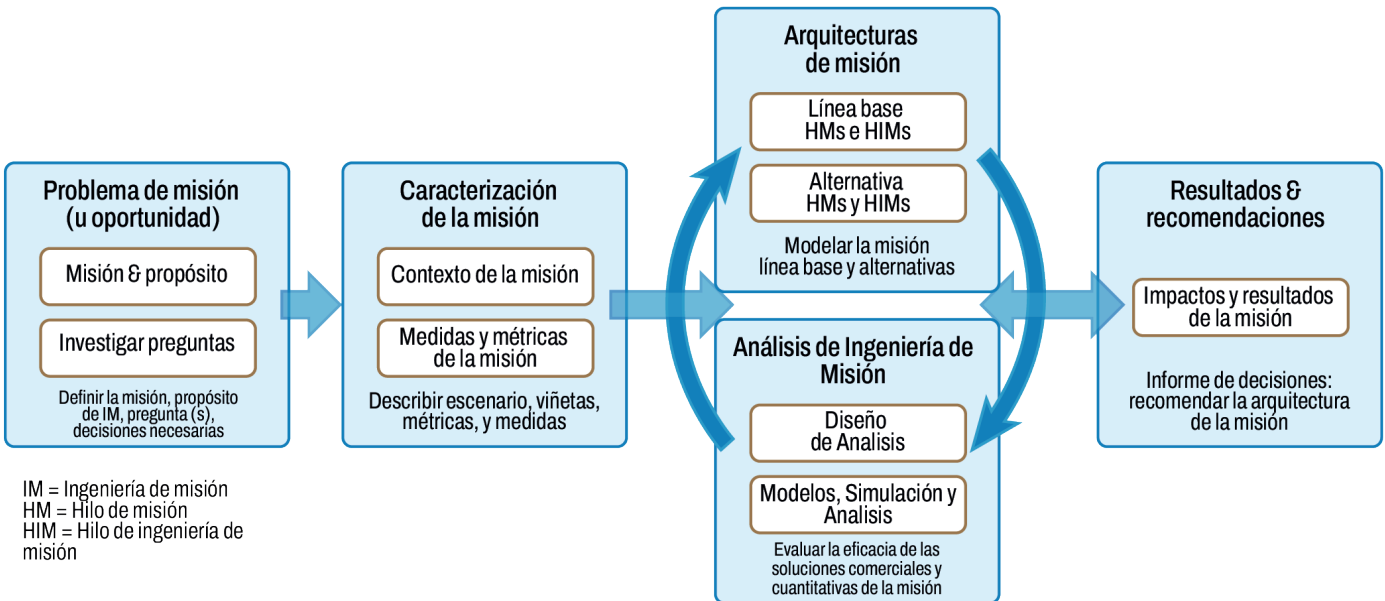


Figura 1. Metodología de IM de la oficina de investigación e ingeniería de la subsecretaría de defensa de EEUU (OUSD R&E).

Una vez que se ha comprendido el éxito de la misión en base a los criterios de éxito identificados, se puede determinar la capacidad operativa y la misión específica a ejecutar. Una capacidad operativa puede describirse desde perspectivas tanto estructurales como de comportamiento. La primera se centra en cómo se estructura una capacidad (DOTMLPF-P para EE. UU., DOTMLPF-I para la UE, MIRADO-I para España); la segunda define los resultados que proporciona una capacidad y, por tanto, define las actividades operativas (OA) que realiza dicha capacidad.

Este enfoque que combina el comportamiento del SdS con una implementación estructurada ha demostrado un gran éxito al aumentar la probabilidad de integración e interoperabilidad con respecto a la efectividad total de un SdS, alejándose de la optimización de sistemas individuales en base a enfoques centrados en la plataforma. Esto es importante para cualquier SdS complejo representado a través de un hilo de ingeniería de misión (HIM), tanto para la ruta principal de ejecución como para las rutas alternativas que proporcionan resiliencia y redundancia a las operaciones. Un HIM es una secuencia completa de pasos que ilustra la tecnología, el personal y los recursos necesarios para lograr un objetivo de misión en unas condiciones específicas.

Ayuda a los ingenieros a comprender cómo interactúan los diferentes sistemas dentro de un SdS a la hora de cumplir su misión general. En general, un HIM aporta información sobre las funciones, los actores y las interacciones involucradas en una misión, el flujo de datos y la toma de decisiones entre los diferentes sistemas, el impacto de los factores ambientales en el desempeño de la misión y la resiliencia y la disponibilidad de los sistemas críticos para la misión.

Dado que la combinación de estos HIM representa una cadena de actividades operativas que produce efectos para una determinada capacidad operativa, se denominarán «redes de efectos», a diferencia de, por ejemplo, la cadena de destrucción (kill chain), para capturar diferentes tipos de efectos graduales, como la distracción o la degradación, que no siempre dan lugar a la destrucción. Para abordar las diferentes taxonomías y ontologías en los distintos sectores industriales y servicios militares, se emplea un marco o arquitectura de referencia común que permite una comparación lógica entre los productos de la IM (véase la Figura 2).

Este enfoque operativo también es válido para aplicaciones comerciales, pudiendo ser definidos sus propios efectos deseados, así como sus HIM y las redes de efectos correspondientes.

Evaluación de amenazas y asignación de armas (ARMA) vs. (BLANCO)

Periodo de tiempo (aaaa-aaaa)

Parámetros del entorno operativo (ubicación) (hora del día) (condiciones) (otros)

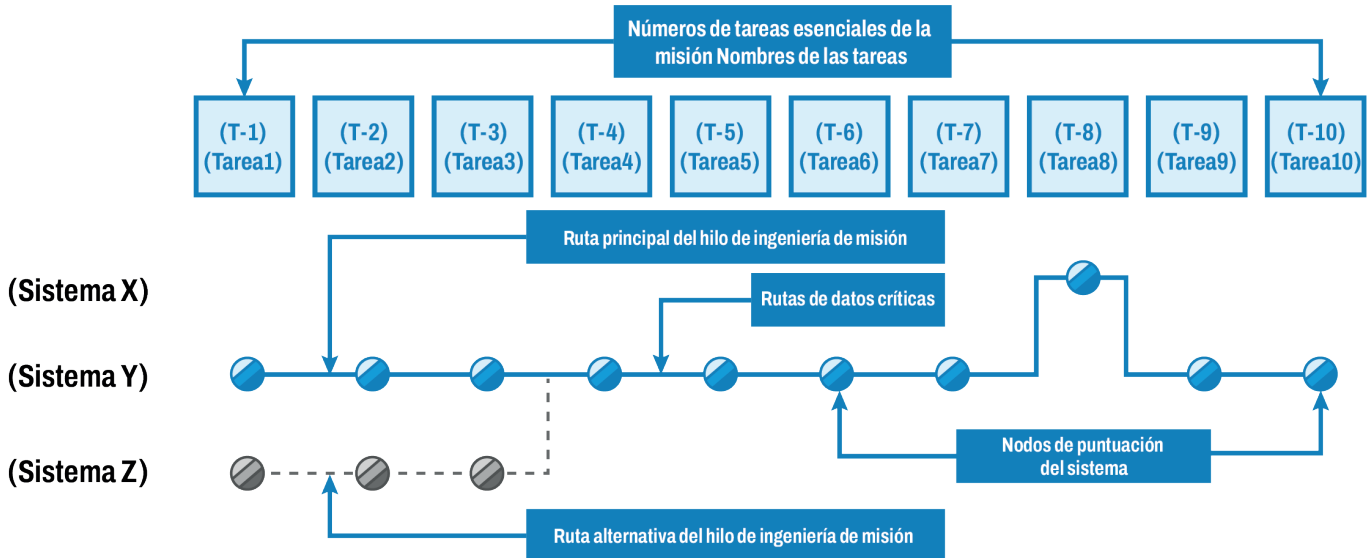


Figura 2. Marco de una red de efectos genérica que proporciona una terminología común [2].

2. RAZONES QUE IMPULSAN EL NUEVO ENFOQUE Y LA DISCIPLINA

Los avances en las capacidades de los adversarios (o competidores) suponen un reto significativo para la seguridad (o el éxito empresarial), ya que surgen nuevas amenazas que comprenden desde sistemas de armas de última generación y tácticas de ciberguerra hasta la toma de decisiones basada en inteligencia artificial (u otras tecnologías novedosas). Cualquiera de estas tecnologías, así como muchas otras, puede alterar rápidamente el equilibrio de poder (o el mercado). Estos avances permiten a los adversarios (o competidores) operar con mayor competitividad (alcance, persistencia, velocidad, precisión y letalidad en contextos de defensa), lo que podría erosionar las ventajas estratégicas y dar lugar a vulnerabilidades en sistemas de defensa críticos (o en la cartera de productos/mercados en sectores comerciales).

Un HIM fallido aparece cuando se producen interrupciones o fallos en la secuencia de actividades y en los sistemas necesarios para alcanzar un objetivo de la misión. Esta situación provoca un fallo en una actividad operacional (OA), comprometiendo la eficacia de la misión. Esto, a la larga, se traduce en una brecha o déficit de capacidad. Esta carencia puede deberse a diversos factores, como fallos técnicos, ciberataques, integración inadecuada entre sistemas y procesos, fallos logísticos, interrupciones de comunicación o interrupciones en la cadena de suministro. Por ejemplo, la entrega insuficiente de combustible o municiones, el deterioro de las comunicaciones satelitales o los retrasos en el movimiento de tropas pueden interrumpir eslabones críticos en la ejecución de la misión.

Para mitigar los riesgos asociados con los fallos de los HIM, generalmente se emplean criterios de redundancia y resiliencia en la planificación y ejecución de las misiones. La redundancia garantiza la disponibilidad de vías alternativas y sistemas de respaldo para realizar funciones críticas y mantener la continuidad de las operaciones cuando fallan los sistemas principales. La resiliencia se centra en la capacidad de adaptarse y recuperarse rápidamente ante interrupciones, permitiendo que las misiones resistan y se recuperen ante desafíos inesperados. En conjunto, la redundancia y la resiliencia crean una arquitectura operativa robusta que garantiza el éxito de la misión incluso en condiciones adversas, reforzando la integridad y la fiabilidad de los HIM.

El enfoque de la IM impulsa a considerar las posibles rutas de los hilos de la misión utilizando el Marco de Red de Efectos, basándose en la ejecución de las tareas esenciales de la misión (p. ej., cómo la fuerza planea combatir en un contexto militar). La definición de este marco proporciona un

mecanismo común para descomponer la misión en funciones esenciales —también llamadas actividades funcionales (AF) en el marco de la OTAN—, de modo que los sistemas conjuntos/de la coalición puedan mapearse de forma lógica. De lo contrario, cuando cada fuerza conjunta/de la coalición aporte su propio marco y terminología a la evaluación, realizar un mapeo real desde la función al sistema en toda la fuerza resulta muy difícil, o incluso inviable. Anteriormente, esto ha conllevado problemas de integración e interoperabilidad que han provocado el fallo del HIM y la falta de una capacidad del SdS para misiones específicas, debido principalmente a la aplicación aislada del análisis funcional a sistemas independientes que, en última instancia, no cumplían plenamente con una capacidad operativa.

Las evaluaciones integrales de misión (extremo a extremo) se realizan para garantizar que un SdS es resistente y resiliente. Éstas se basan en evaluaciones operacionales que comienzan con una capacidad y que extienden el análisis operacional (OA) y el análisis funcional (FA) a todo el SdS, a la vez que impulsan la integración e interoperabilidad de múltiples sistemas para garantizar la precisión en la determinación de las deficiencias de la capacidad. Este enfoque integral, que abarca toda la misión, permite una toma de decisiones basada en datos que no sólo es válida para proporcionar soluciones materiales para carencias o deficiencias operativas, sino que también plantea soluciones para todo el espectro de definición de una capacidad estructural (DOTMLPF-P para EE. UU., DOTMLPF-I para la UE, MIRADO-I para España), considerando tanto las áreas materiales como las inmateriales. Además, incluso aplicado únicamente a soluciones materiales, el enfoque propuesto ofrece mayores ventajas cuando se requieren soluciones tecnológicas holísticas que integren un SdS.

Resulta esencial comprender el espectro de la definición de una capacidad estructural a la hora de desarrollar una solución de material, ya que garantiza que ésta no sólo sea tecnológicamente viable, sino también que esté plenamente integrada en el marco operativo general. Una nueva capacidad debe alinearse con la doctrina, contar con el respaldo de estructuras organizativas, incorporar programas de capacitación eficaces y considerar los requisitos de personal, las necesidades de infraestructura y las restricciones políticas. Ignorar cualquiera de estos factores puede dar lugar a una solución material ineficaz, insostenible o incompatible con los sistemas existentes, lo que en última instancia reduce la efectividad de la misión. Comprender el SdS con este nivel de detalle reduce el riesgo de desplegar HIM fallidos al considerar soluciones en todos los dominios operativos y sectores industriales. La Figura 3 ilustra un HIM multi-dominio.

Motivación: Los hilos de ingeniería de misión/redes deben ser adaptables para la eficacia de la misión y las condiciones dinámicas.

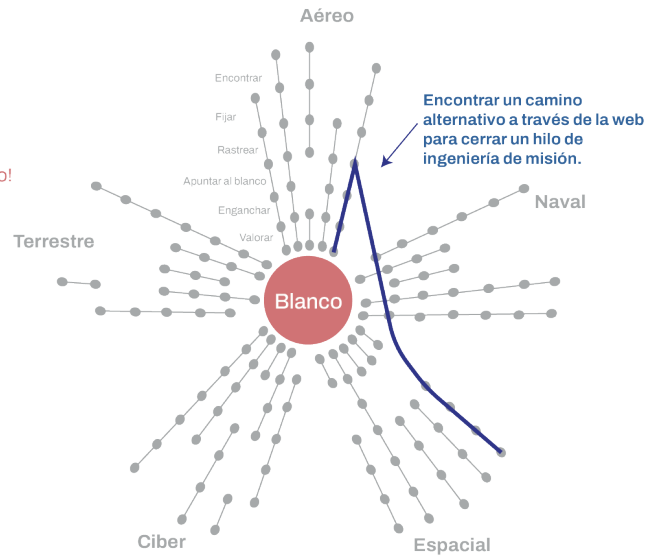
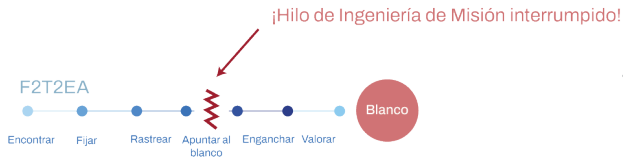


Figura 3. Hilo de Ingeniería de Misión multidominio.

Además, este enfoque permite considerar alternativas resilientes al ejecutar el HIM, a la vez que se reduce la redundancia innecesaria dentro y entre las fuerzas armadas. La Figura 4 representa el caso de una operación multidominio, considerando diferentes alternativas para la ejecución de las misiones, proporcionando resiliencia y redundancia. Para aprovechar esta metodología, es importante asegurar un análisis formal para evaluar la integración e interoperabilidad de los SdS necesarios para ejecutar los hilos de la misión.

El análisis y la evaluación formales de la integración e interoperabilidad de los SdS para ejecutar los hilos de la misión impulsan el desarrollo y la implementación del enfoque actual de la IM del Departamento de Defensa de EE. UU. El análisis mediante el Marco de Red de Efectos identifica brechas operativas, puntos de inserción de soluciones científicas y tecnológicas, conceptos de empleo y requisitos de investigación y desarrollo para el futuro espacio de soluciones y la futura arquitectura en las áreas de misión.

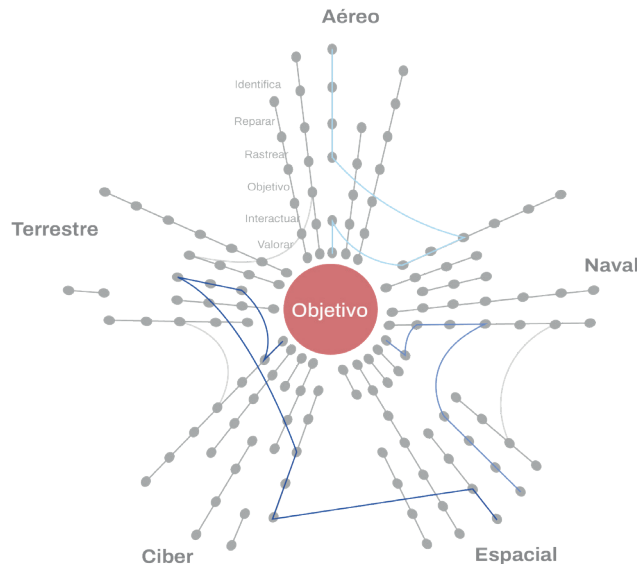


Figura 4. Red de Ingeniería de Misión multidominio.

Para garantizar que los sistemas cumplan con los requisitos operativos y mantengan su eficacia resulta crucial realizar una gestión eficaz de los riesgos de la misión durante todo el proceso de adquisición de defensa. Esto resulta relevante en cualquier industria que implemente capacidades críticas o para cualquier empresa que suministre sistemas esenciales para sus operaciones. Al evaluar el riesgo de la misión, los analistas deben considerar la misión en sí, incluyendo la unidad equipada con el sistema, el entorno relevante de operaciones, las vulnerabilidades del sistema frente a todo el espectro de amenazas previstas y sus efectos combinados en el contexto de sus misiones operativas previstas. En los programas del DoD, las Evaluaciones de Riesgo Basadas en la Misión (MBRA) se están integrando en la planificación de las adquisiciones siguiendo un enfoque basado en la medición de la misión para evaluar las funciones críticas de la misma frente a los posibles efectos de las amenazas en escenarios operacionalmente representativos (véase la Figura 5). Estas evaluaciones identifican los elementos e interfaces vulnerables del sistema, lo que orienta el alcance de las pruebas y de las evaluaciones operativas, incluyendo las pruebas y evaluaciones con fuego real, para garantizar que los sistemas puedan resistir amenazas reales. Mediante la realización del MBRA, los gestores de programa pueden supervisar y cuantificar los riesgos para los objetivos de las

pruebas, los programas de adquisición, los usuarios finales y las operaciones generales del DoD. Este enfoque basado en el riesgo garantiza que las estrategias de prueba estén dimensionadas y enfocadas adecuadamente, lo que facilita el apoyo a la toma de decisiones y mejora la resiliencia y la eficacia de los sistemas de defensa en entornos operativos complejos.

Cuando fallan los HIM, la consiguiente reelaboración y pérdida de capacidades puede degradar significativamente la preparación para la misión y la seguridad nacional. Es posible que sea necesario reasignar, retrasar o rediseñar por completo activos críticos, lo que genera un desperdicio de recursos, un aumento de costes y contratiempos operativos. Estos fallos también pueden generar deficiencias en la capacidad, reduciendo la efectividad de la fuerza y dejando vulnerabilidades que los adversarios pueden explotar. Los efectos en cascada de estas interrupciones pueden obstaculizar objetivos estratégicos, retrasar tiempos de respuesta y, en última instancia, debilitar la capacidad de una nación para proyectar poder, defender sus intereses y mantener una ventaja tecnológica en un panorama de amenazas en constante evolución.

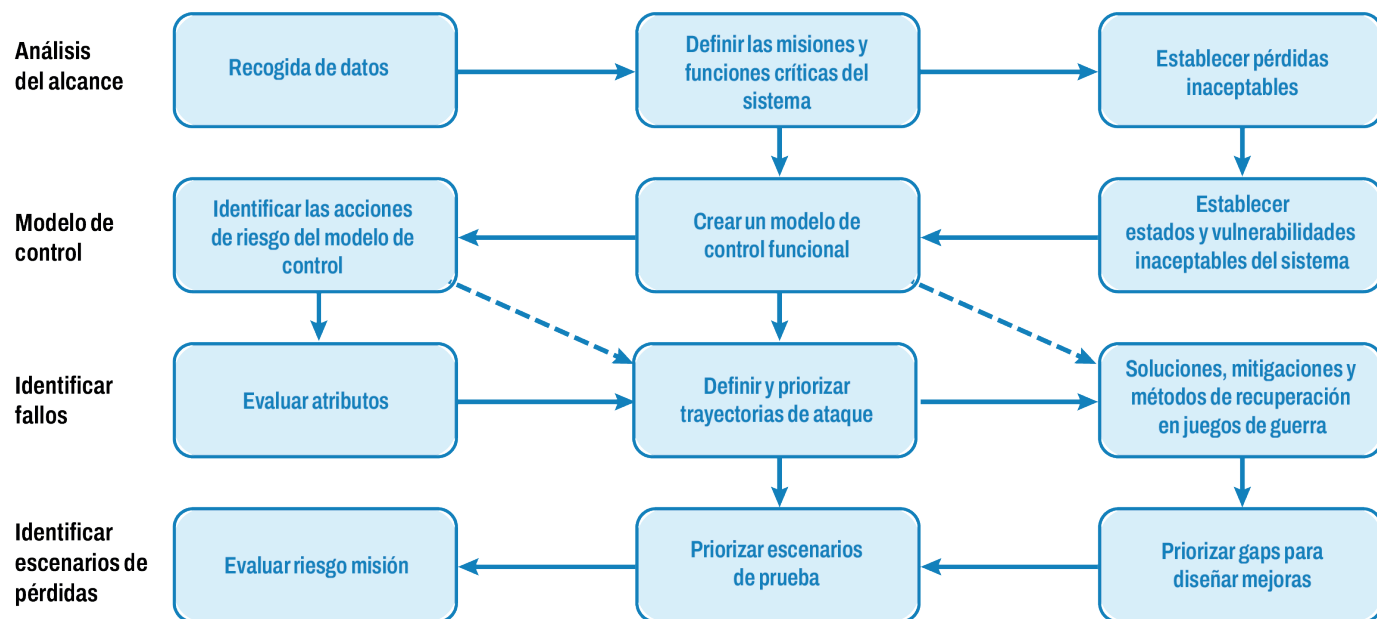


Figura 5. Análisis de riesgos basado en la misión.

3. PROPUESTA DE VALOR

La evaluación de las tecnologías, de los sistemas o de las capacidades del Departamento de Defensa de EE. UU. requiere un enfoque de SdS para analizar el impacto de estas complejas inversiones en los diversos dominios de superficie, submarinos, aéreos, terrestres, espaciales y cibernéticos, así como en la integración de las fuerzas de coalición. En el entorno actual de adquisiciones, los programas suelen madurar de forma independiente y la integración de los SdS se produce en el terreno, en lugar de durante las primeras etapas del desarrollo, lo que incrementa el coste y el tiempo necesarios para introducir una nueva capacidad al usuario final. La ingeniería de misión se centra en las evaluaciones basadas en capacidades para generar capacidades integradas de combate que puedan trasladarse a directrices programáticas específicas para traducir las necesidades operativas en SdS y en requisitos de sistema, impulsando así la preparación actual y las capacidades futuras de cada servicio militar. Este enfoque también proporciona información crucial sobre las deficiencias operativas para proporcionar información a las partes interesadas y permite asegurar que se están realizando las inversiones adecuadas.

Los complejos, y ahora altamente integrados, sistemas de operaciones continúan evolucionando, lo que permite una mayor precisión, una proyección de poder más efectiva y unas posturas defensivas más seguras para las fuerzas armadas. La interconectividad de la sociedad se está trasladando a los buques, aeronaves, submarinos, redes, activos espaciales, carros de combate y las armas que estos portan. Por ejemplo, determinar las inversiones adecuadas para el desarrollo de activos espaciales y la protección de aplicaciones de posicionamiento, navegación y referencia temporal es extremadamente complejo, lo que requiere este enfoque sistemático para identificar todas las interfaces principales y los puntos de intersección entre los numerosos sistemas/plataformas en un entorno multidominio y multifuncional. Comprender un SdS con este nivel de detalle brinda la oportunidad de reducir el riesgo de desplegar hilos/capacidades de misión fallidos al analizar de forma individual los dominios de combate u operativos y las fuerzas armadas. Además, este enfoque permite prestar atención a alternativas resilientes para la ejecución de los hilos de misión críticos, a la vez que reduce la redundancia innecesaria dentro y fuera de la fuerza conjunta/de coalición, según la forma en que las fuerzas planifiquen el combate, considerando la integración y la interoperabilidad

4. ASPECTOS CARACTERÍSTICOS DE LA INGENIERÍA DE MISIÓN

4.1. La continuidad a través de los criterios de éxito para la misión

Los SdS que se requieren para ejecutar misiones operativas suelen optimizarse de forma fragmentada, sin considerar su utilidad operativa total. Los análisis de IM suelen utilizar simulaciones y otras herramientas para desarrollar Medidas de Éxito (MoS, por sus siglas en inglés), Medidas de Efectividad (MoE, por sus siglas en inglés) y Medidas de Rendimiento (MoP, por sus siglas en inglés) para los sistemas que los componen, a la vez que evalúan los criterios de éxito de la misión para las tareas esenciales, con el fin de alcanzar los objetivos generales de la misión mediante la experimentación. Esto también es aplicable a las empresas.

Vincular la efectividad de la misión desde los niveles estratégico, táctico y operativo requiere un enfoque estructurado que alinee los objetivos generales con criterios viables y medibles en cada nivel. A nivel estratégico, las organizaciones definen las MoS que articulan el resultado final deseado y el impacto a largo plazo de la misión. Estos criterios generales sirven como principios rectores que garantizan que todos los esfuerzos subsiguientes contribuyan a los objetivos generales, como la seguridad nacional o la autonomía estratégica, la estabilidad económica o la influencia regional. Los líderes estratégicos determinan las MoS evaluando factores como la capacidad de disuasión, la presencia global o la preparación de la fuerza, garantizando la alineación con los objetivos de política y los imperativos estratégicos más amplios.

A nivel operativo, estas medidas de éxito estratégico se traducen en MoE que evalúan la eficacia de las tareas esenciales de la misión (a menudo estructuradas como un hilo conductor de la misión) para alcanzar los objetivos estratégicos. Las MoE permiten evaluar si las acciones operativas clave contribuyen a los resultados deseados, centrándose en la eficacia en lugar de la eficiencia. Por ejemplo, en un contexto militar, las MoE para una misión de superioridad aérea podrían incluir el control sostenido del espacio aéreo designado, el tiempo de respuesta ante amenazas emergentes o las tasas de desgaste del enemigo. Estas medidas garantizan que los comandantes puedan ajustar las tácticas, la asignación de recursos y la ejecución de la misión en respuesta a las condiciones reales, manteniendo la alineación con la intención estratégica.

Finalmente, a nivel táctico, las MoP se establecen para evaluar el funcionamiento de los sistemas, el personal y los procesos individuales en la ejecución de las tareas

esenciales de la misión. Las MoP son cuantitativas y se centran en las capacidades específicas del sistema, como la tasa de generación de salidas de las aeronaves, el estado de preparación, la precisión de las armas o la disponibilidad de los sistemas de comunicación críticos. Estas medidas proporcionan información directa sobre si los activos y procesos a nivel táctico funcionan según los estándares requeridos para respaldar la eficacia operativa.

En resumen, los criterios de las MoE miden si se está haciendo lo correcto y los criterios de las MoP miden si se está haciendo correctamente. Mediante la monitorización continua de las MoP, las unidades tácticas pueden realizar los ajustes necesarios, garantizando que su desempeño contribuya directamente al éxito de la misión a niveles superiores. Las MoP se redactan tradicionalmente como Parámetros Clave de Rendimiento y Atributos Clave del Sistema, ya que son los criterios integrados en los documentos de requisitos para el diseño o la modificación de sistemas y donde se define el espacio de diseño. La vinculación entre estos tres niveles garantiza que los objetivos estratégicos impulsen la planificación operativa, y la eficacia operativa dependa de

la ejecución táctica. Al definir claramente las MoS, las MoE y las MoP, las organizaciones crean un marco coherente que alinea la toma de decisiones, la asignación de recursos y la evaluación del desempeño, mejorando en última instancia el éxito general de la misión. La Figura 6 muestra un mapeo representativo de estos niveles de criterios de puntuación, que existe para crear continuidad en todos los niveles de operaciones con el objetivo de lograr la unidad de esfuerzos para el éxito de la misión definida.

Las Operaciones Basadas en Efectos (OBE) son un ejemplo de cómo los indicadores y las métricas transformaron las operaciones militares tras la Primera Guerra del Golfo. Anteriormente, la reducción de los activos adversarios era la norma. El objetivo era eliminar las fuerzas enemigas en busca de una retirada o de una rendición, pero el enfoque de las OBE se basaba en el análisis de los Centros de Gravedad (CdG) del enemigo para diseñar planes que permitan denegar el acceso a dichos centros bajo la premisa del mínimo coste. Por lo tanto, las OBE se basan en efectos, los cuales son generados por las actividades operativas que, en definitiva, constituyen los elementos funcionales más básicos

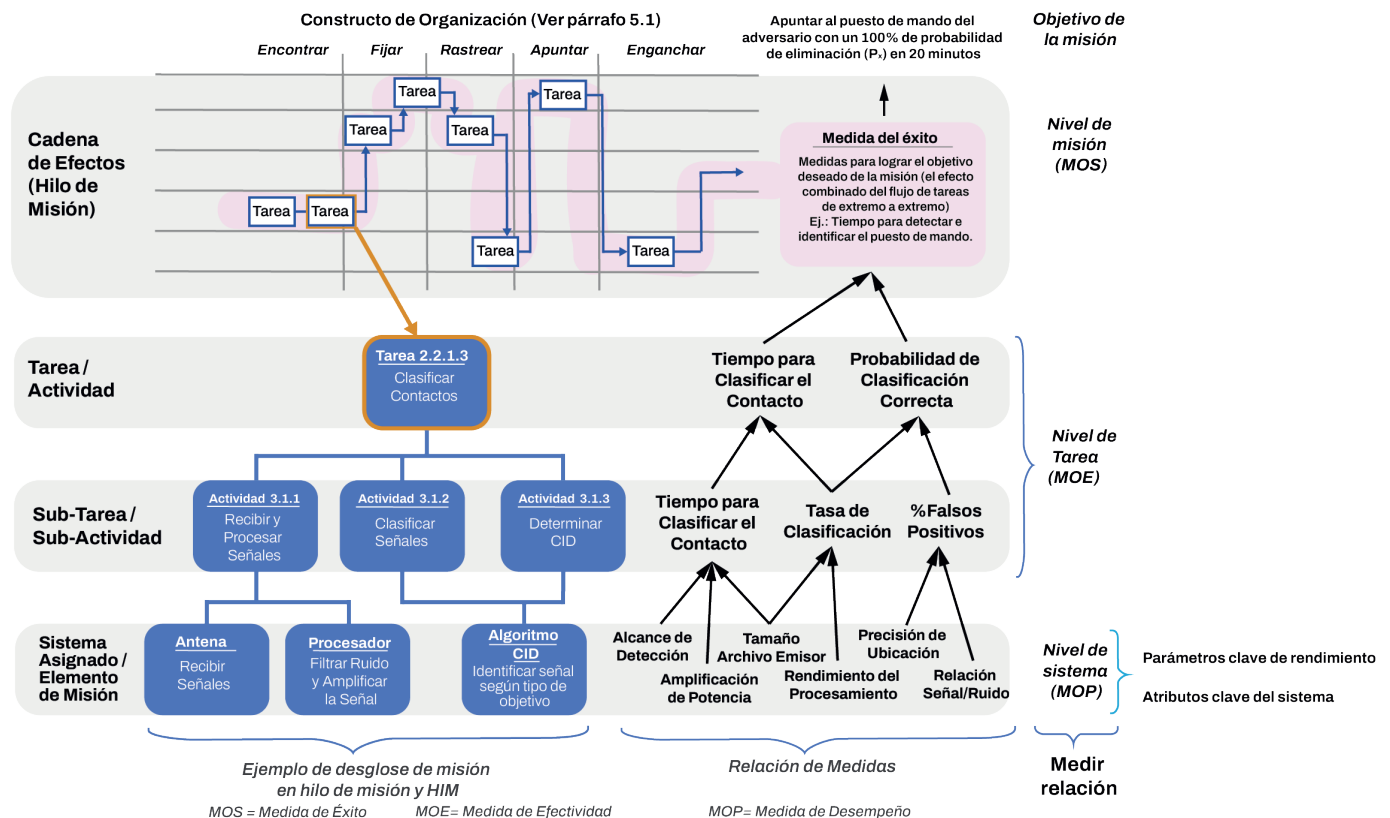


Figura 6. Desarrollo de los criterios de éxito de la misión en todos los niveles de operación.

de una capacidad según la definición de la OTAN. Por tanto, los efectos se sustentan en las capacidades, vinculando todo el conjunto de niveles de ejecución operativa, desde el estratégico hasta el táctico. Dado que los CdG están actualmente dispersos por todo el ámbito de la misión, el SdS permite sincronizar la consecución de efectos combinados para que generen valor para la operación. La IM sigue este enfoque para extender este vínculo al nivel técnico.

En la ejecución actual de los procesos de planificación de las adquisiciones, los sistemas se diseñan y desarrollan en función de una necesidad específica para cumplir un requisito, no necesariamente para respaldar los efectos de un SdS. Esto es típico del proceso de adquisición a través del Sistema Conjunto de Desarrollo Integrado de Capacidades (JCIDS) del DoD estadounidense, diseñado para desarrollar un sistema dentro de un contexto específico, en lugar de en un dominio o contexto integrado basado en la misión. Los sistemas individuales no se desarrollan con la idea de integrarse en la arquitectura de un SdS para ejecutar HIM operativos. En otras palabras, no son programas centrados en capacidades que persigan la generación de efectos operativos notables a nivel de SdS. Es comprensible que los gestores del programa optimicen el diseño del producto en función de las necesidades específicas de sus clientes, sin evaluar cómo interactúa el sistema con otros sistemas implementados para apoyar de forma colaborativa los efectos deseados durante la ejecución de los hilos críticos de la misión. Este diseño de sistema de compartimentos estancos “intraóptimo” resulta subóptimo

cuando se evalúa posteriormente desde una perspectiva de un SdS “interóptima”. Desafortunadamente, se requeriría un considerable rediseño para modificar el sistema con respecto al éxito de la misión de un SdS. La ingeniería y la implementación de productos teniendo en mente el éxito de la misión del SdS durante el diseño del sistema es la forma de evitar problemas de integración e interoperabilidad en toda la fuerza (ver Figura 7).

Un desafío incontrolable es que el número de escenarios a analizar para respaldar un diseño robusto de un SdS aumenta exponencialmente con la incorporación de cada misión operativa, cada parámetro de rendimiento, cada criterio de diseño, cada amenaza y cada factor ambiental. Esta explosión combinatoria desafía el uso de muchos procesos y herramientas tradicionales en una IM realista para el diseño de SdS complejos. Es necesario desarrollar principios, procesos y herramientas para evaluar y respaldar el diseño de SdS complejos utilizando el enfoque de la IM para que resulte manejable. Dicho esto, los principios y fundamentos de la ingeniería de sistemas son necesarios, pero no suficientes, para abordar estas complejidades actuales.

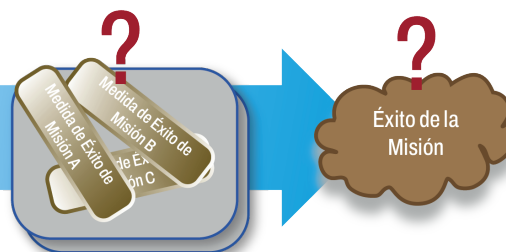
Si bien el JCIDS (Sistema Conjunto de Integración y Desarrollo de Capacidades) ofrece un enfoque de adquisición estructurado dentro del Departamento de Defensa de EE.UU., actualmente no existe un equivalente directo en Europa. Marcos como el AAP-20 de la OTAN o enfoques nacionales (por ejemplo, el MIRADO-I de España) aún están

Ruta de adquisición actual con diseño centrado en el sistema/plataforma

La aportación de las plataformas a la misión se define después de que el sistema se implemente; los resultados se mezclan



La utilidad se evalúa después de que el sistema ha sido entregado y se han tomado todas las decisiones de compromiso



Ruta diseñada en torno a la misión y el éxito de la misma

La aportación de las plataformas a la misión se concibe para contribuir a criterios específicos de éxito de la misión



Las medidas de la misión alinean la utilidad y las decisiones de compromiso



- Ejemplo: eliminación de todos los objetivos en la incursión
- Ejemplo: ningún aliado involucrado



Figura 7. Alineamiento de la Misión operativa del SdS con los criterios de éxito definidos para la Misión

en desarrollo y difieren en su madurez y gobernanza. Un enfoque de ingeniería basado en misiones contribuye a garantizar que el sistema diseñado se integre fácilmente con otros sistemas, a la vez que garantiza las capacidades necesarias para el éxito de la misión. Además, el sistema de adquisición no cuenta con la capacidad de gobernar un SdS; esto significa que, al realizar la ingeniería de misión (IM), es improbable que permanezca estable durante más tiempo que uno de los ciclos de actualización o mantenimiento de los sistemas. Este enfoque oportunista de la IM no proporciona necesariamente estabilidad a largo plazo para apoyar las misiones, pero sí que sitúa a las partes interesadas en el camino correcto para obtener una efectividad funcional desde el principio. Como se mencionó anteriormente, el enfoque de adquisición para el concepto, el diseño y la

experimentación debería desarrollarse para garantizar la gobernanza en cualquier SdS mediante la sincronización, que actualmente va en la dirección de la gestión de la cartera de capacidades.

4.2. Enfoque disciplinado para el despliegue de capacidades

El enfoque estructurado y descrito para la IM que se encuentra en MEG v2.0 puede mejorarse siguiendo el proceso de 10 pasos presentado en este apartado. Este proceso se relaciona con las categorías del enfoque MEG v2.0, como se muestra en la Figura 8.

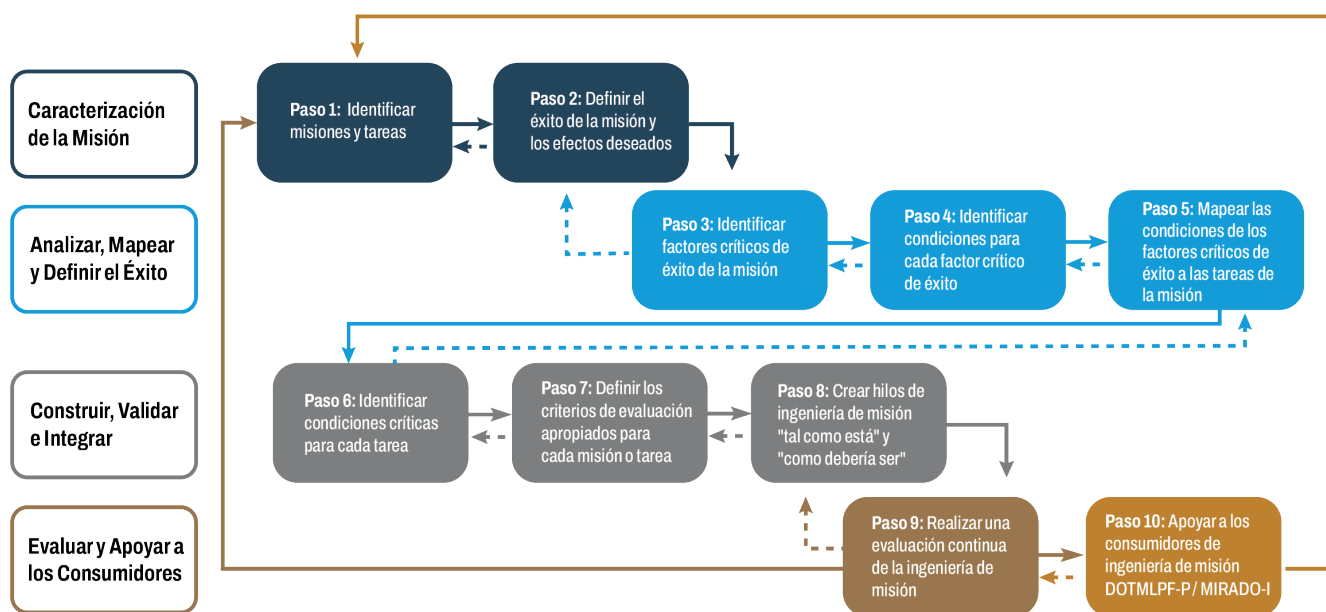


Figura 8. Proceso de 10 pasos de IM mapeado a MEG v2.0.

Este proceso comienza con la identificación de las misiones y las tareas y avanza a través de la definición de los criterios de éxito de la misión, los factores críticos de éxito y las condiciones asociadas que impactan el rendimiento. Al relacionar estas condiciones con las tareas de la misión y desarrollar criterios de puntuación claros, los responsables de la toma de decisiones pueden evaluar las capacidades actuales con las necesidades futuras, creando hilos conductores de ingeniería de misión "actual" (as-is) y "futura" (to-be) para guiar las mejoras. Las evaluaciones continuas de IM garantizan la adaptabilidad en entornos dinámicos, mientras que la alineación con DOTMLPF-P (DOTMLPF-I en la UE, MIRADO-I en España) garantiza que las iniciativas de la IM apoyen el desarrollo de una fuerza más amplia.

Mediante este enfoque estructurado, la IM proporciona una base sustentada en datos para el desarrollo de capacidades, la mitigación de riesgos y la eficacia operativa, garantizando que la planificación y la ejecución de la misión se mantengan resilientes ante las amenazas emergentes.

El orden de los 10 pasos de la IM se describe a continuación, comenzando con la priorización de las áreas de la misión y terminando con la gestión continua de los HIM de extremo a extremo (end-to-end) para mantener una correcta ejecución de las capacidades.

- 1) Identificar misiones y tareas.
- 2) Definir el éxito de la misión y el efecto deseado.
- 3) Identificar los factores críticos para el éxito de la misión.
- 4) Identificar las condiciones para cada factor crítico para el éxito de la misión.
- 5) Asignar las condiciones críticas para el éxito de la misión a las tareas de la misión.
- 6) Identificar las condiciones críticas para cada tarea.
- 7) Definir los criterios de valoración adecuados para cada misión o tarea.
- 8) Crear hilos de ingeniería de misión “actual” (as-is) y “futura” (to-be).
- 9) Realizar evaluaciones continuas de ingeniería de misión.
- 10) Apoyar el consumo de la Ingeniería de Misión por parte del DOTMLPF-P (DOTMLPF-I en la UE, MIRADO-I en España).

La ejecución de este proceso presenta desafíos asociados con la estructura de gobernanza, la disponibilidad y recopilación de datos, la coordinación de las partes interesadas en todo el Departamento de Defensa de EE. UU., los múltiples ciclos de vida del sistema debido a su maduración (de legado a nuevo) y el desarrollo del personal y de herramientas, entre otros. Sin embargo, el Marco de Red de Efectos (Figura 2) proporciona un mecanismo para traducir lo que el Departamento de Defensa de EE. UU. planea adquirir en la capacidad resultante. La Figura 9 ilustra un ejemplo del resultado de este proceso, que muestra la capacidad potencial de un SdS como un HIM evaluado.

4.3. Oportunidad de integrar capacidades entre naciones

Nuestras naciones se vuelven cada año más complejas a medida que se avanza hacia la implementación del “Internet de las Cosas” y se trabaja en la implementación de “Gemelos Digitales” para una mayor conectividad y una monitorización del estado del rendimiento. La falta de enfoques comunes para analizar estos complejos SdS con nuevas herramientas y procesos seguirá causando retrasos en la capacidad para ejecutar un desarrollo rápido y en la implementación de capacidades.

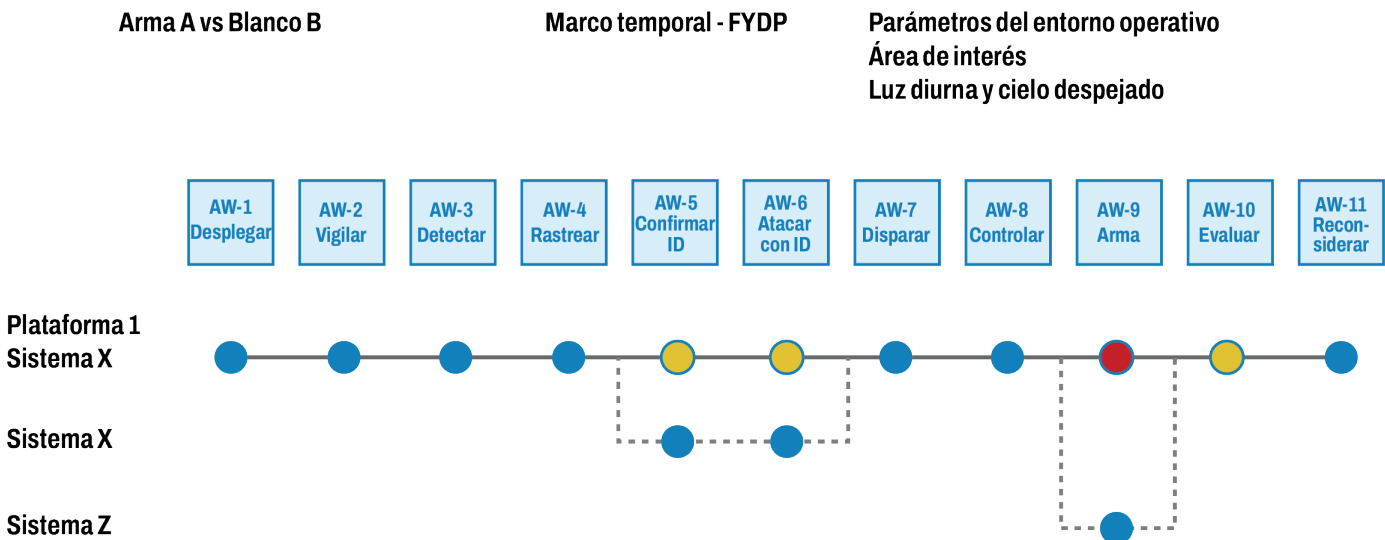


Figura 9. Ejemplo de criterios de valoración.

El enfoque proporcionado por la IM es tremendamente prometedor para reunir a la comunidad internacional aliada, con el ánimo de generar capacidades que permitan establecer de manera compartida e integrada una fuerza conjunta o de coalición. Concepto ilustrado en la Figura 10. Esta figura muestra solo a los socios aliados que han estado involucrados en discusiones recientes con el Departamento de Defensa de los Estados Unidos; sin embargo, esta metodología permitiría incorporar a otros socios y aliados en las discusiones para aumentar la coordinación y la colaboración utilizando los principios y procesos de la IM.

Mientras que la IM está madurando rápidamente dentro del Departamento de Defensa de EE. UU., instituciones europeas como la Agencia Europea de Defensa (EDA), la Dirección General de Armamento y Material (DGAM) de España y marcos nacionales como MIRADO-I están integrando progresivamente los principios de la IM en sus procesos de planificación y desarrollo de capacidades. Una mayor colaboración entre las iniciativas de IM de EE. UU. y de Europa podría contribuir a crear metodologías compartidas, estándares de interoperabilidad y evaluaciones conjuntas de SdS.

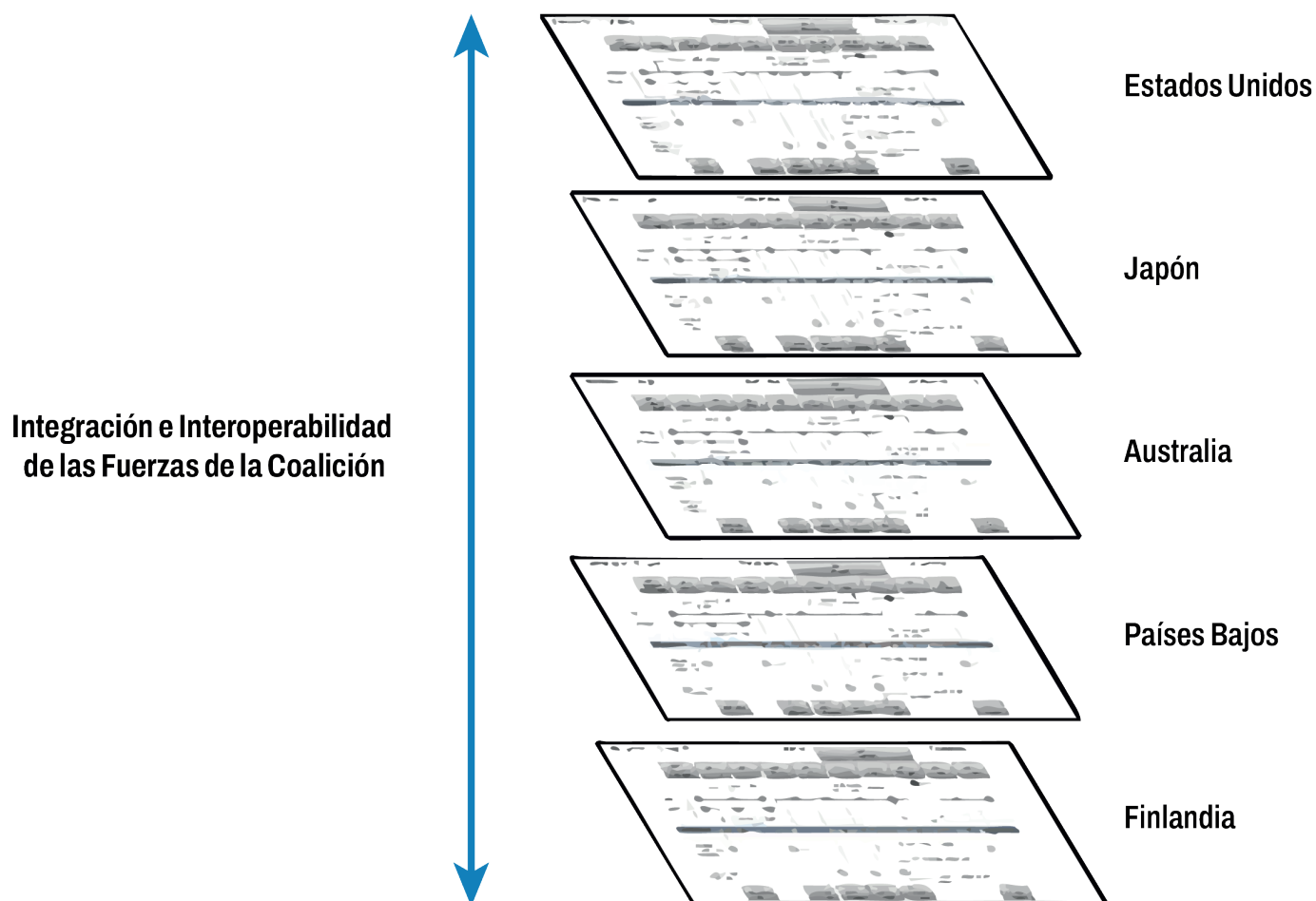


Figura 10. Integración vertical e interoperabilidad a través de socios y aliados

5. CONCLUSIONES

La Ingeniería de Misión ofrece un enfoque estratégico para el diseño y la gestión de SdS complejos que van más allá de las aplicaciones de defensa tradicionales. Si bien resulta esencial para la seguridad nacional y la autonomía estratégica, la IM también mejora la capacidad para abordar diversos desafíos, como la respuesta a desastres, la asistencia humanitaria, la resiliencia de las infraestructuras y la exploración espacial. Al aprovechar principios como la interoperabilidad, la arquitectura de sistemas abiertos y la toma de decisiones basada en el análisis de riesgos, una organización o nación puede optimizar los recursos y mejorar los resultados de las misiones en múltiples ámbitos.

La aplicación de la IM en la cooperación cívico-militar, la gestión de emergencias y la innovación tecnológica garantiza que una organización se mantenga ágil para responder ante desafíos, tanto previstos como imprevistos. La integración de consideraciones como los sistemas humanos, factores de sostenibilidad y la colaboración intersectorial fortalece aún más la capacidad de un país para desarrollar soluciones que equilibren la eficacia operativa con los beneficios sociales a largo plazo. En definitiva, la IM permite a una organización configurar proactivamente sus capacidades de misión de forma adaptable, rentable y tecnológicamente avanzada. Al fomentar un enfoque que abarque la defensa, la seguridad y las prioridades nacionales más amplias, una organización puede mejorar su resiliencia estratégica, contribuyendo a la estabilidad y el progreso a escala nacional e internacional.

REFERENCIAS

1. OSD R&E. (2023, October). Mission Engineering Guide (MEG) v2.0. Retrieved from https://ac.cto.mil/wp-content/uploads/2023/11/MEG_2_Oct2023.pdf.
2. Moreland Jr., James D. (2014). "Mission Engineering Integration and Interoperability." NSWCCD Leading Edge Technical Digest, NSWCCD-MP-15-00096 01/15: pp. 4-15.

BIOGRAFÍAS

DR. JAMES MORELAND JR.

El Dr. James Moreland Jr. se retiró del Servicio Ejecutivo Superior del Departamento de Defensa de EE.UU. en 2020 y posteriormente trabajó como Vicepresidente de Estrategia y Vicepresidente de Ingeniería de Misión e Integración en Raytheon Technologies, finalizando sus cinco años de servicio en la compañía en abril de 2025. Anteriormente, trabajó como



Director Ejecutivo de Ingeniería de Misión e Integración, Subsecretario Adjunto de Defensa para Sistemas de Guerra Táctica y Director Ejecutivo de Guerra Naval en la Oficina del Secretario de Defensa, así como Ingeniero Jefe del Centro de Guerra Naval de Superficie, División Dahlgren. Posee una licenciatura en Ingeniería Mecánica por la Universidad de Maryland, una maestría en Ingeniería de Sistemas por Virginia Tech, una maestría en Estrategia de Recursos Nacionales por el Colegio Industrial de las Fuerzas Armadas y un doctorado en Ingeniería de Sistemas por la Universidad George Washington. Ha sido asesor ejecutivo sénior de la Junta Nacional de Ciencias de EE.UU. y de la Oficina de Ciencia y Tecnología de la Casa Blanca, y formará parte del Consejo Asesor Científico de la Fuerza Aérea de EE.UU. a partir de septiembre de 2025.

TTE. CORONEL VÍCTOR M. SOBRINO

El Teniente Coronel Víctor M. Sobrino es piloto de F-18 del Ejército del Aire y del Espacio español actualmente adscrito a la Agencia Europa de Defensa como Oficial de Proyecto para el Programa Europeo de Mando y Control. Antes de este destino, fue Jefe del área operacional de la Oficina de Programa Española del Sistema de Armas de Última Generación (NGWS/FCAS), así como el jefe del 122 escuadrón de Ala 12 y oficial de Estado Mayor en el Mando Aéreo de Combate del Ejército del Aire y del Espacio. También sirvió en el Componente Aéreo de la Fuerza Conjunta (JFAC). Aparte de sus estudios militares, que incluyen el título de Oficial de Estado Mayor de las Fuerzas Armadas, es Graduado en Ingeniería Informática, Máster en Inteligencia Artificial Avanzada, Máster en Administración de Empresas y Máster en Dirección y Gestión de Adquisiciones de Sistemas para la Defensa. Actualmente está cursando un doctorado en Inteligencia Artificial por la Universidad Politécnica de Madrid (UPM) y el Centro Nacional de Investigaciones Oncológicas (CNIO).





Last Update 15:24
Date Analysing...
Date GMT+1

Modelado de la arquitectura de la misión

Matthew Gagliardi, *System Strategy, Inc. (SSI)* (MGagliardi@Systemxi.com)

Matthew C Hause, *System Strategy, Inc. (SSI)* (MHause@Systemxi.com)

Dr. James N Martin, *The Aerospace Corporation* (James.N.Martin@aero.org)

Víctor Ramos del Pozo, *Isdefe* (vramos@isdefe.es)

Resumen

Al planificar y llevar a cabo un estudio de ingeniería de misión (IM) es importante contar con un modelo completo, preciso y coherente de la arquitectura de la misión. Este capítulo explora algunas de las características y construcciones clave del modelado que permiten el desarrollo de un modelo de la arquitectura de misión que se utilizará en apoyo de un estudio de IM. En él se analizan las extensiones que se aplican al Marco de Arquitectura Unificado (UAF, *Unified Architecture Framework*) para respaldar mejor las actividades de IM.

Palabras clave

Ingeniería de Misión, Arquitectura de Misión, Hilos de Misión, Hilos de Ingeniería de Misión, Arquitectura Empresarial, Marco de Arquitectura Unificada.



1. INTRODUCCIÓN

La arquitectura de la misión puede involucrar múltiples empresas u organizaciones y relaciones complejas entre las entidades que las componen (incluidos los sistemas de sistemas y los sistemas individuales). El Marco de Arquitectura Unificada (UAF, por sus siglas en inglés) funciona bien en la captura de muchos de los aspectos “no físicos” de los sistemas de sistemas relevantes, ya que destaca las soluciones materiales y no materiales (es decir, doctrina, organización, adiestramiento, liderazgo, personal, instalaciones) que están involucradas en una arquitectura de misión. UAF especifica un conjunto de vistas de la arquitectura para describir varios aspectos de una empresa y las entidades principales de la empresa [1-3] y proporciona un lenguaje de modelado (UAFML) que está especialmente diseñado para modelar una empresa u organización. Como tal, es apropiado para modelar arquitectura de misión grande y compleja junto con su variedad de escenarios, viñetas, hilos de misión (HM), hilos de ingeniería de misión (HIM), etc. [4, 5]. Algunos de los elementos del modelo estructural relevantes para el modelado de misiones se ilustran en Figura 1 y se describe en la Tabla 1.

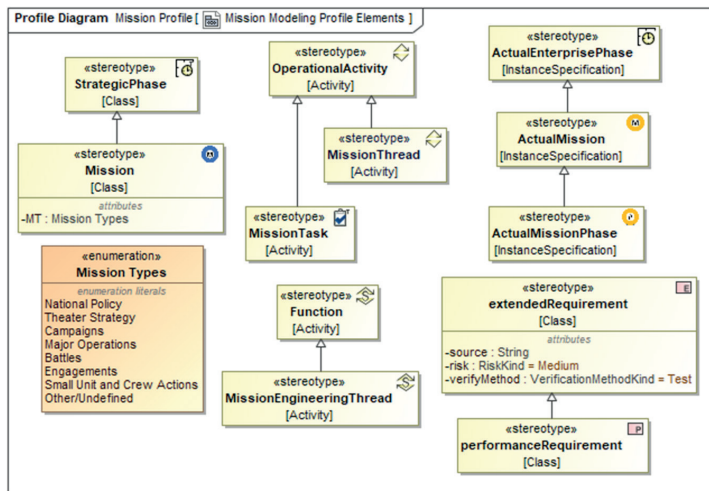


Figura 1. Vista de perfil de modelado de misión

Elemento de modelado	Descripción
Misión	Un elemento de misión es una generalización de un elemento de una fase empresarial en el metamodelo de dominio UAF.
Misión real	Una misión real es una generalización de un elemento de fase empresarial real en el metamodelo de dominio UAF.
Fase real de la misión	Una fase de misión real es una especialización de una misión real que proporciona una especificación de instancia de una misión y una fase de misión.
Hilo de la misión	Un hilo de misión es una generalización de una actividad operativa.
Tarea de la misión	Una tarea de misión también es una generalización de una actividad operativa y los hilos de misión se componen de otros hilos de misión o tareas de misión.
Hilo de ingeniería de misión	El hilo de ingeniería de misión es una generalización de una función y describe la implementación de las fases reales de una misión. La trazabilidad entre el HM y el HIM utiliza el estándar UAF para implementar relaciones.

Tabla 1. Elementos de modelado para su uso en un modelo de arquitectura de misión.

Si bien UAF proporciona un gran número de potenciales vistas de arquitectura, como se muestra en la Figura 2, el modelado de la arquitectura de misión sólo necesita un pequeño subconjunto de estas vistas. La definición del problema de la misión y los aspectos de caracterización de la misma, junto con los elementos y vistas del hilo de la misión (HM) que se utilizarán en la ingeniería de misión (IM), se corresponden principalmente con los puntos de vista estratégicos y operativos de UAF, como se ilustra en la Figura 2 [6]. Los subprocesos de ingeniería de misión (HIM) son una implementación de los HM, por lo que se representan principalmente desde el punto de vista de los recursos. Sin embargo, se debe tener en cuenta que hay otros puntos de vista de UAF y sus vistas de modelado asociadas que podrían usarse fácilmente en un estudio de IM y en actividades relacionadas, como la planificación de capacidades, la gestión de carteras empresariales, la formulación de presupuestos anuales, el análisis y evaluación de programas, el desarrollo de requisitos del sistema, etc.

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	Status St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr	
Architecture Management Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architecture Relationships Am-Cn	Architecture Development Method Am-Pr			Duration Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr	
Summary & Overview Sm-Ov													
Strategic St	Capacidades & Habilos			Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic Status St-St		Strategic Information St-If	MOE's & MOP's		Strategic Constraints St-Ct	Strategic Roadmap St-Rm	Strategic Traceability St-Tr
Operational Op		Hilos de Misión						Operational Information Op-If	MOE's & MOP's		Operational Constraints Op-Ct	Operational Roadmap Op-Rm	Operational Traceability Op-Tr
Services Sv	Requirements Rq-Mv	Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services Status Sv-St	Services Sequences Sv-Sq	Operational Information Model Op-If	MOE's & MOP's		Services Constraints Sv-Ct	Services Roadmap Sv-Rm	Services Traceability Sv-Tr
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel Status Ps-St	Personnel Sequences Ps-Sq	Operational Information Model Op-If	MOE's & MOP's		Personnel Constraints Ps-Ct	Personnel Roadmap Ps-Rm	Personnel Traceability Ps-Tr
Resources Rr		Hilos de Ingeniería de Misión						Resources Information Model Rr-If	MOE's & MOP's		Resources Constraints Rr-Ct	Resources Roadmap Rr-Rm	Resources Traceability Rr-Tr
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr			Resources Information Model Rr-If	MOE's & MOP's		Security Constraints Sc-Ct	Security Roadmap Sc-Rm	Security Traceability Sc-Tr
Projects Pj		Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr			Resources Information Model Rr-If	MOE's & MOP's		Projects Constraints Pj-Ct	Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr
Standards Sd		Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr					Resources Information Model Rr-If	MOE's & MOP's		Standards Constraints Sd-Ct	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actual Resources Ar			Actual Resources Structure Ar-Sr	Actual Resources Connectivity Ar-Cn		Simulation			MOE's & MOP's		Parametric Execution/Evaluation		

Figura 2. Vistas de ingeniería de misión en UAF.

Para ilustrar cómo se puede utilizar el lenguaje de modelado para definir una arquitectura de misión, en las siguientes secciones se presenta la aplicación de los conceptos de IM del capítulo anterior sobre un ejemplo de misión utilizando el modelado de misiones.

2. EJEMPLO DE MISIÓN

El ejemplo de misión utilizado en este capítulo es la Batalla de Hoth de la segunda película de Star Wars, “El Imperio Contraataca”. Esta misión se utiliza porque es bien conocida, contiene una rica fuente de sistemas, estrategias, misiones y comportamientos, e ilustra operaciones conjuntas¹. El ejemplo, adaptado de [4, 5], se concentra en las visiones Estratégica y Operativa, definiendo los conceptos de misiones, fases de misión, hilos de misión y arquitecturas operativas, así como la definición de los recursos y las estructuras organizativas y funcionales.

La misión se define como una sucesión de escenarios y viñetas a lo largo del tiempo, en una secuencia de fases de la misión. La arquitectura operativa, que se compone de varios HM y HIM asociados, se define para satisfacer estos factores motivacionales. La arquitectura operativa se utiliza como base para el análisis de la eficacia operativa, que a veces se realiza mediante herramientas y técnicas de modelado y simulación.

2.1. Propósito de la misión, partes interesadas, preocupaciones, objetivos e impulsores (drivers)

Las medidas de éxito (MOS, *Measures of Success*), los efectos deseados y las capacidades se capturan en el modelo de arquitectura de la misión y los elementos de la arquitectura operativa se remontan a estos elementos para garantizar una cobertura adecuada. Las Medidas de Efectividad (MOE, por sus siglas en inglés) se asignan a las tareas de la misión que componen cada uno de los hilos de la misión.

Las definiciones de los conceptos clave relacionados se muestran en Tabla 2 y los elementos de misión correspondientes se muestran en la Figura 3. El Comandante de la Legión está preocupado por su pérdida de posición, o posiblemente de su vida, lo que suele ocurrir cuando se produce un fracaso al servicio del Imperio. Desea evitar un resurgimiento rebelde y asegurar una victoria decisiva. Darth Sidius y Darth Vader desean controlar la Galaxia y establecer el dominio del lado oscuro. Darth Vader también desea proteger a Luke Skywalker. Estas preocupaciones se relacionan directamente con los objetivos de la misión, que a su vez se vinculan con los impulsores (*drivers*) que han obligado al Imperio a actuar. Estos se discutirán más adelante en el capítulo.

1. Para obtener más información, consulte [7] o, mejor aún, tome palomitas de maíz y vea la película.

Concepto	Descripción
Preocupación	Un asunto de relevancia o importancia para un grupo de interés con respecto a una entidad de interés.
Conductor	Cosa que obliga a trabajar o actuar; lo que te impulsa a seguir adelante.
Reto	Una situación exigente o estimulante; una llamada a participar en un concurso o pelea.
Estado de la empresa	Condición con respecto a las circunstancias o atributos.
Capacidad	Capacidad para lograr un efecto deseado en condiciones y entornos definidos.
Oportunidad	Una posibilidad debido a una combinación favorable de circunstancias.
Riesgo	Una fuente de peligro; la posibilidad de incurrir en pérdidas o infortunios.
Efecto	Un fenómeno que sigue y es causado por algún fenómeno previo.
Resultado	Algo que sucede o se produce como consecuencia o producto final.
Meta	Una declaración sobre un estado o condición de la empresa que se llevará a cabo o se mantendrá a través de los medios apropiados.
Objetivo	Una declaración de un objetivo alcanzable, dirigido en el tiempo y medible que la empresa busca cumplir para lograr sus objetivos.

Tabla 2. Elementos de motivación estratégica.

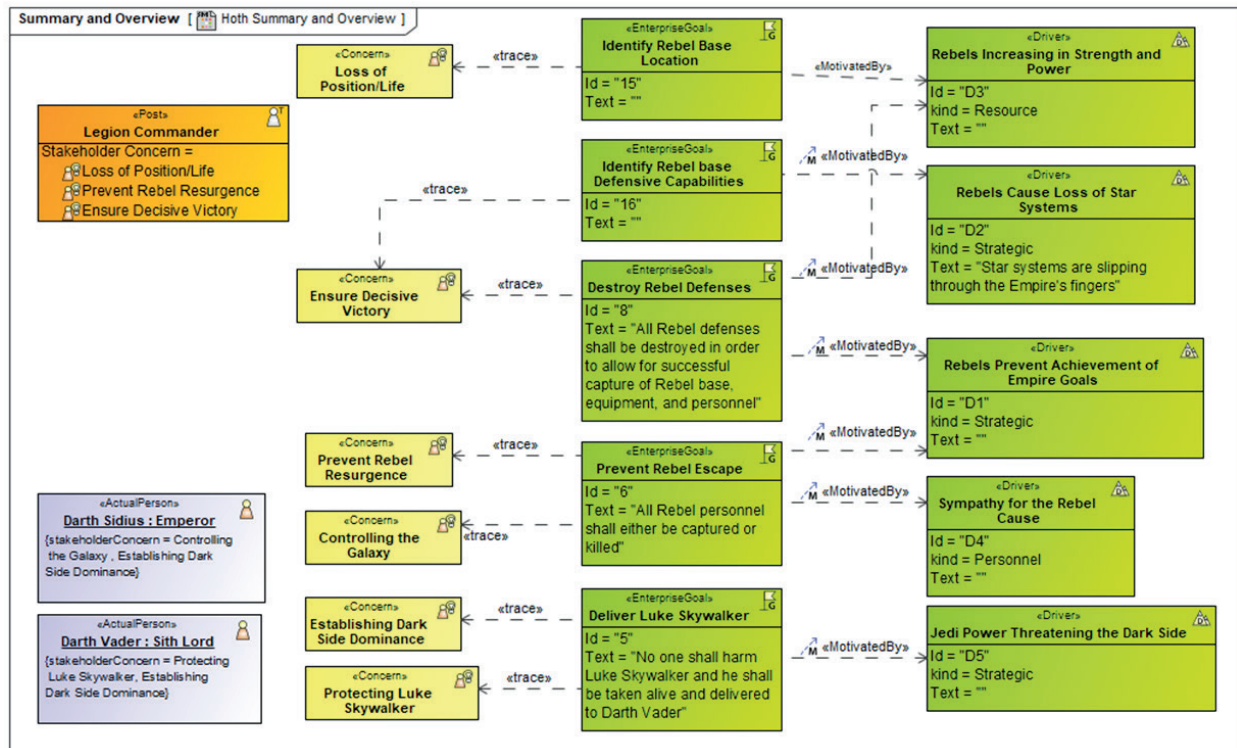


Figura 3. Resumen y descripción general de Hoth: Preocupaciones y objetivos de las partes interesadas.

2.2. Definición de la misión

La estructura de la Misión del Imperio que se muestra en la Figura 4 ilustra la complejidad requerida para modelar misiones. La doctrina del Imperio prescribe que toda misión militar tiene dos fases: Planificación y Ejecución. Una Misión de Invasión Planetaria se compone de Misiones de Exploración, Desembarco y Ataque separadas, cada una con sus propias Fases de Planificación y Ejecución. Estos son todos los tipos de misiones de invasión. Cada uno de ellos tiene un tipo de misión definido. Las fases de ejecución y planificación heredan los atributos tiempo de misión y fase de misión. Se han definido atributos para varios de los tipos de misión. Los tipos de escenario y viñetas se han vinculado a las misiones que, cuando se instancian, definen los parámetros y el contexto de la misión. Estos se detallan en la siguiente sección. Los hilos de misión específicos pueden y deben vincularse a las distintas misiones para definir los aspectos funcionales.

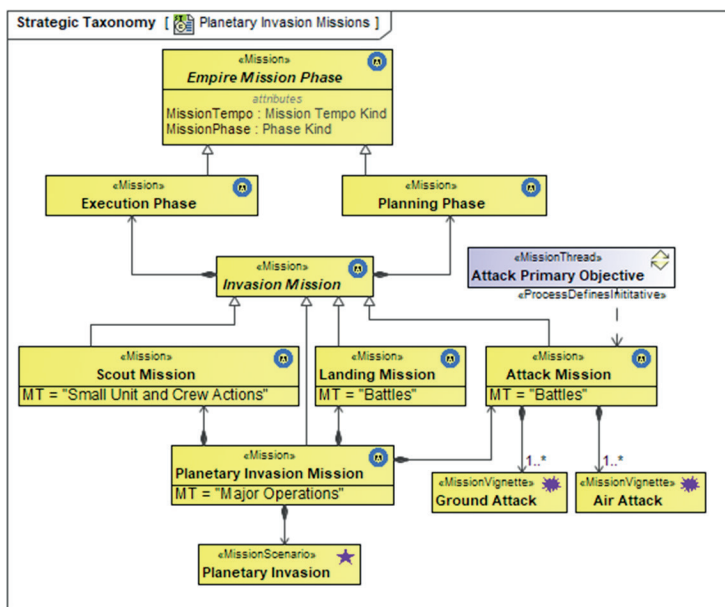


Figura 4. Definiciones de misión.

2.3. Campaña, escenario y viñeta

Dado que los nombres de las misiones pueden incluir el tipo de misión (por ejemplo, Campaña de Hoth, Operación de exploración de Hoth o Batalla terrestre de Hoth) y la jerarquía estructural de la misión puede mostrar qué misión se compone de otras misiones, los tipos de misión de la Figura 5 se definen en una enumeración y se modelan como un atributo para los tipos de misión de la Figura 3 en lugar de definir estereotipos específicos. Los valores elegidos para el ejemplo de Hoth se muestran en la Figura 4.



Figura 5. Niveles de guerra y tipos de misiones [8].

Para los conceptos de escenario y viñeta, se utilizan nuevos estereotipos para describir la información de contexto necesario para la(s) misión(es) que se describen en el modelo, como se muestra en la Figura 6. Los escenarios describen la ubicación geográfica y el marco temporal del conflicto general. Incluyen información como contextos y antecedentes político-militares favorables y de amenazas, suposiciones, restricciones, limitaciones, objetivos estratégicos y otras consideraciones de planificación [8]. Las viñetas describen conjuntos ordenados y específicos de eventos, comportamientos e interacciones para un conjunto específico de sistemas para incluir capacidades azules y amenazas rojas dentro del entorno operativo. Las viñetas pueden representar partes pequeñas, idealmente autónomas, de un escenario [8].

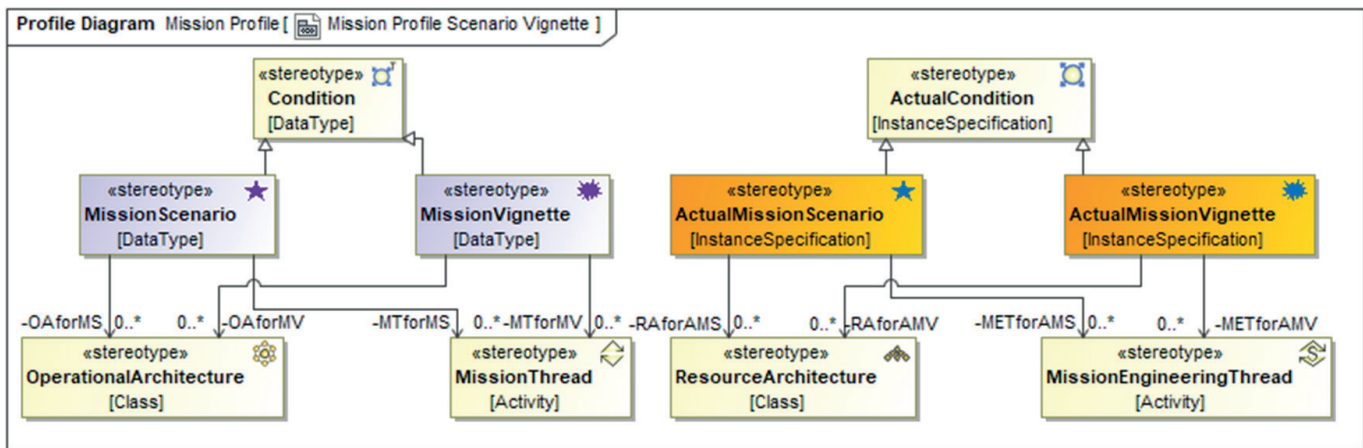


Figura 6. Extensiones del escenario y la viñeta a UAF.

Dado que ambos conceptos describen un conjunto de información relativa a una misión, el estereotipo más útil para extender es una condición, en la que cada uno de ellos tiene sus propios elementos que se relacionan con la información del contexto descrita en el la Guía de Ingeniería de Misión del Departamento de Defensa de EEUU. A continuación, los modeladores pueden crear condiciones reales que tengan valores específicos para el escenario y la viñeta adecuados, así como los umbrales para determinar el éxito, y luego aplicarlos a las misiones reales específicas dentro de su modelo, proporcionando la trazabilidad necesaria a sus misiones. Dado que el escenario y la viñeta pueden ser aplicables a cualquier tipo de misión (como se ve en la Figura 5), el escenario debe aplicarse a la misión real de nivel superior en el modelo y las viñetas deben crearse y aplicarse a cada misión real por debajo de la de nivel superior.

En el ejemplo de Hoth, en la Figura 7 se muestra cómo se definen y aplican los elementos Scenario y Vignette a las misiones adecuadas. A la izquierda hay un escenario de misión y una viñeta predeterminados. Estos elementos se incluirán en el perfil como ejemplos de la misma manera que los Objetivos de Capacidad Militar (OCM) y los factores MIRADO (Material, Infraestructura, Recursos, Adiestramiento, Doctrina y Organización). Estos se han extendido para el escenario de Batalla de Hoth para una invasión planetaria, la viñeta de ataque terrestre y la viñeta de ataque aéreo. Estos pueden incluir condiciones y valores adicionales. A lo largo de la parte inferior hay un conjunto de condiciones que se pueden utilizar en todo el modelo con respecto al entorno, la topografía y la situación política. Estos son utilizados por las instancias de escenario y viñeta de la derecha. A continuación, se vinculan a las definiciones de misión para que los datos reales de la misión puedan hacer referencia a los datos reales de la viñeta y del escenario. En este ejemplo, el escenario real contiene las viñetas.

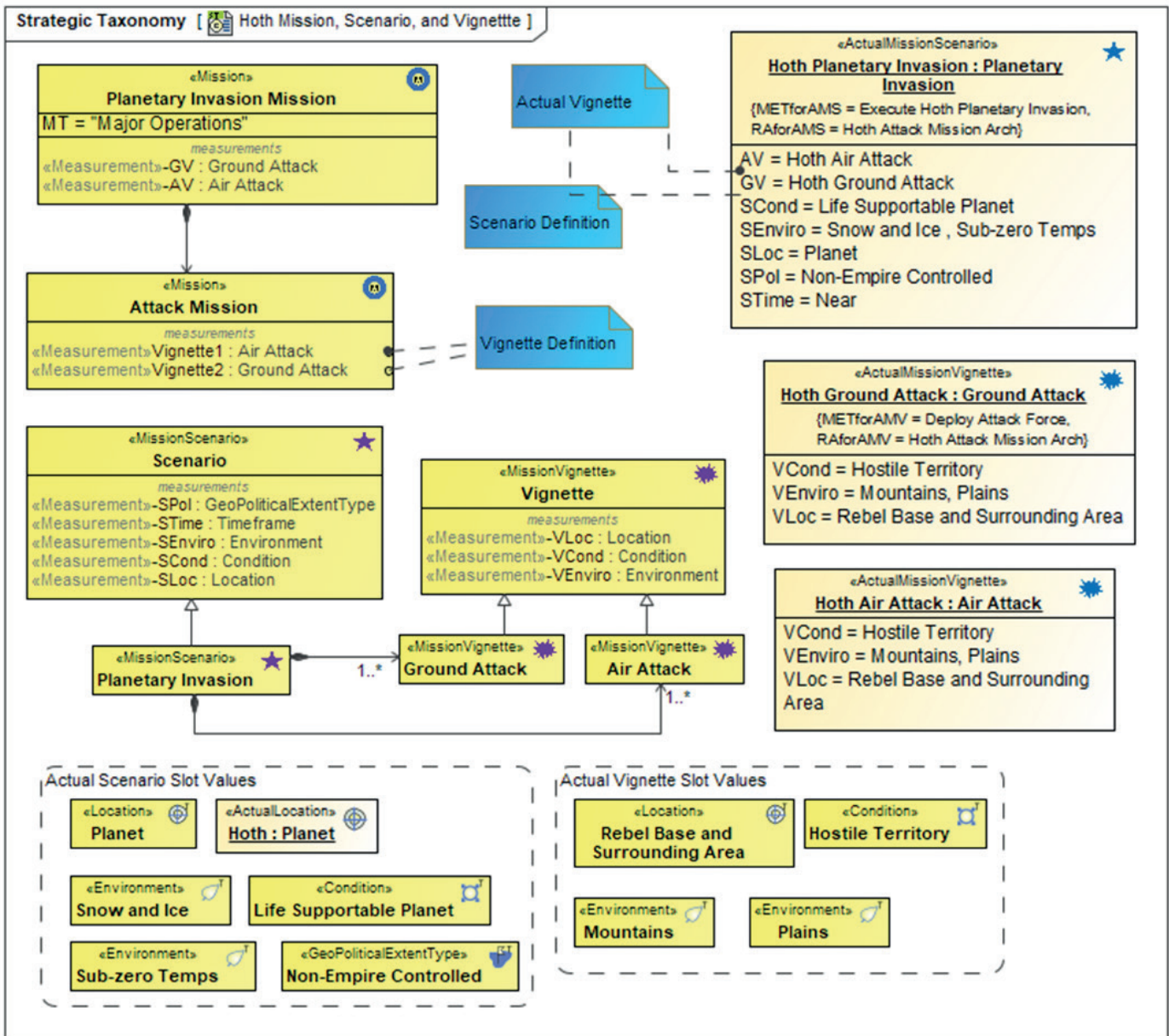


Figura 7. Definición de escenario y viñeta.

2.4. Relaciones de Misión

Se utilizan dos relaciones para conectar los tipos de misión a los hilos de misión (*Process Defines Initiative*), que se muestran en la Figura 8, y las misiones reales a los hilos de ingeniería de misión (*Process Adapts to Initiative*), que se muestran en la Figura 9. La Figura 9 muestra la estructura de la Invasión Hoth, que es una instancia de la Misión de

Invasión Planetaria definida en la Figura 4. Esta Misión Real se compone de la Fase de Planificación y Ejecución, así como de la Misión de Desembarco, la Misión de Ataque y la Misión de Exploración. Cada una de estas misiones tiene fases de planificación y ejecución. Todas las fases de ejecución tienen HIM asignados. La fase de ejecución de Hoth AMEP tiene objetivos definidos, así como una arquitectura operativa y de recursos.

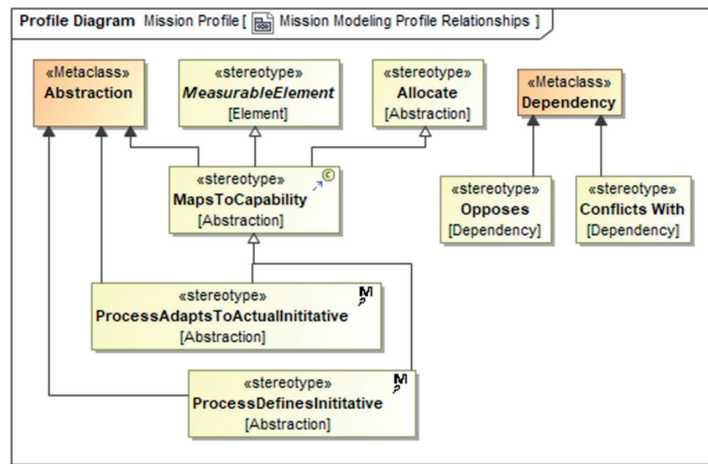


Figura 8. Relaciones de misión.

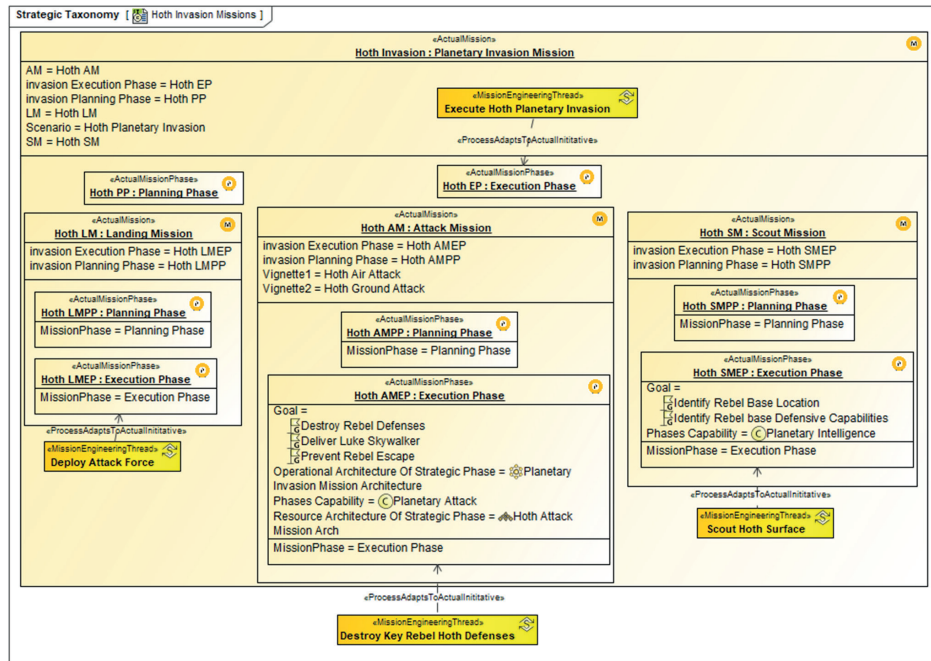


Figura 9. Misiones reales y fases de la misión.

2.4.1. Elementos conflictivos

La Oposición y el Conflicto son inherentes a la Ingeniería de Misión. Algunos de estos son obvios en el contexto de esta misión: las Fuerzas del Imperio atacan a las Fuerzas Rebeldes, los Cañones de Energía atacan al Escudo de Defensa, etc. Otros no son tan evidentes, como el conflicto dentro de los Objetivos de la Misión que se muestra en la Figura 10. El objetivo de capturar a Luke Skywalker reduce las posibilidades de destruir las defensas rebeldes y evitar la huida de los rebeldes.

Normalmente, el Imperio ejecuta sus misiones con extremo prejuicio, prefiriendo destruir un planeta en lugar de permitir que los enemigos escapen o que se libere información. Dado que tenían que atacar con fuerzas convencionales y hacerlo con mucho cuidado, no pudieron destruir las fuerzas ni impedir la huida de los rebeldes. Poner de relieve estos elementos contradictorios ayudaría a garantizar un resultado satisfactorio y proporcionaría un medio para mitigar el riesgo y otros aspectos. Cada meta se descompone a su vez en sus objetivos. Los objetivos definen los logros a corto plazo, mientras que las metas son a largo plazo.

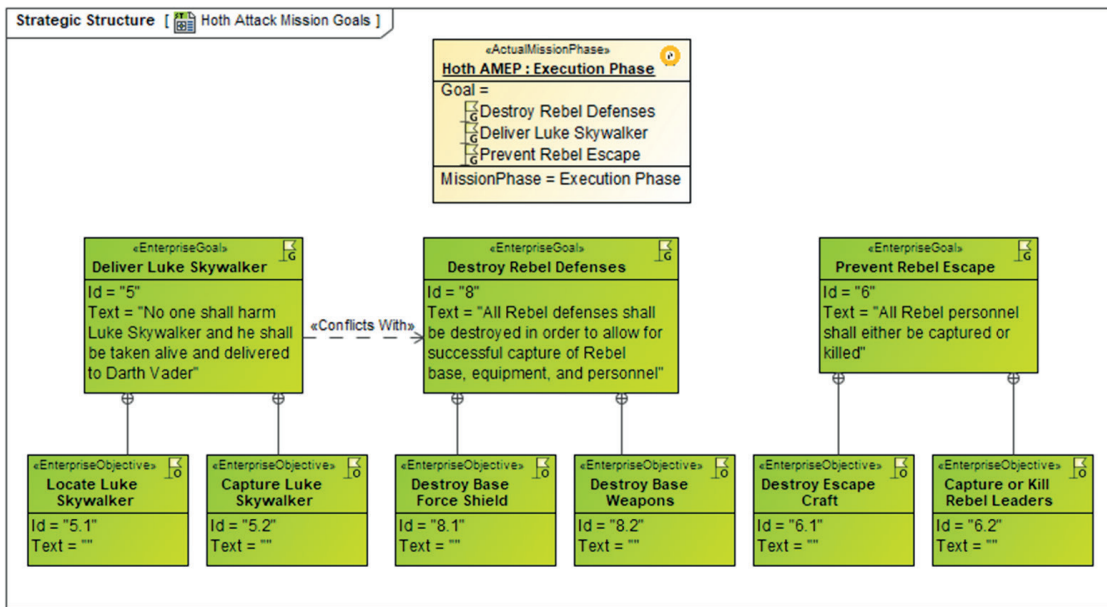


Figura 10. Metas y objetivos de la misión.

2.4.2. Vincular la estrategia con la ejecución

Los objetivos, los impulsores (*drivers*), los desafíos, las oportunidades, las fases de la misión, las capacidades y los sistemas están vinculados entre sí en la Figura 11. La fase de ejecución de Hoth AMEP superpone los objetivos de Destruir las defensas rebeldes, Prevenir la fuga rebelde y Liberar a

Luke Skywalker y la Capacidad de ataque planetario. Esto significa que se realizan durante esta fase. Las Arquitecturas de Recursos y Operativas implementan la fase de misión y se ejecuta el hilo de ingeniería de misión. Los riesgos de pérdida de las fuerzas del Imperio y la huida de las fuerzas rebeldes se identifican para las oportunidades. Se pueden desarrollar estrategias de mitigación para estos riesgos.

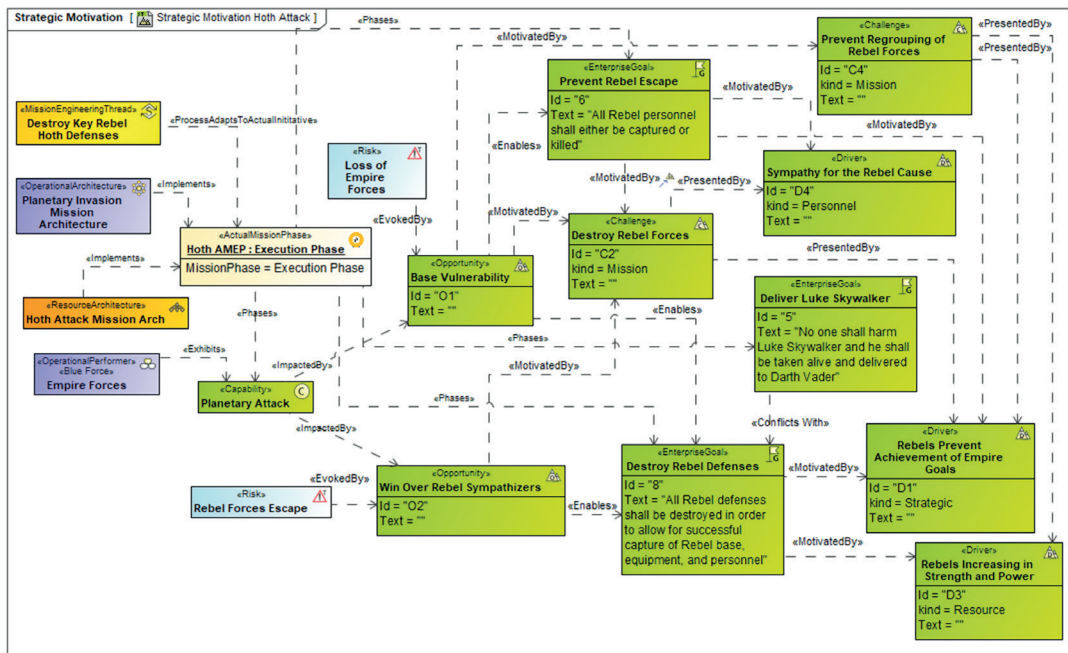


Figura 11. Impulsores de la misión, objetivos, desafíos, oportunidades y capacidades.

3. DESARROLLO DE UNA ARQUITECTURA OPERATIVA

3.1. La arquitectura operativa de la Fuerza Azul

La arquitectura operativa del Imperio, que se muestra en la Figura 12, establece a nivel lógico los principales elementos de las fuerzas del Imperio que se necesitan para ejecutar una Invasión Planetaria. Se debe tener en cuenta que el Imperio también debe estimar qué Fuerzas Rebeldes pueden estar presentes, para que puedan dar cuenta de sus interacciones con las fuerzas del Imperio.

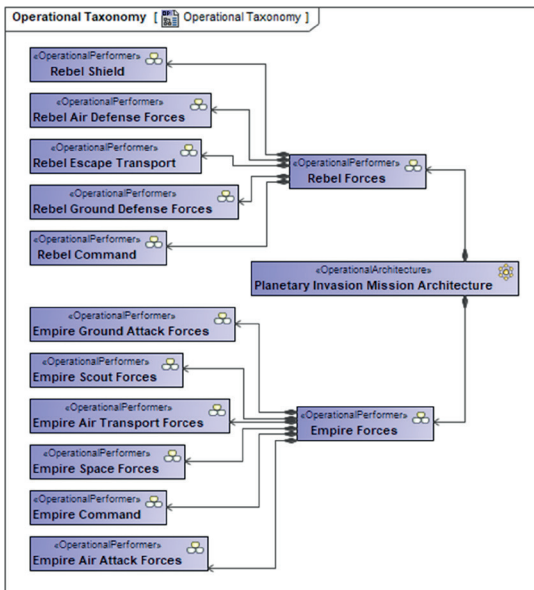


Figura 12. Taxonomía operativa de la fase de ataque.

Estos elementos lógicos candidatos se asignan a las capacidades necesarias para garantizar que todos se hayan considerado, como se muestra en la figura 13.

Legend		Strategic Taxonomy Capabilities							
↗	Exhibits								
↗	Exhibits (Implied)								
	Empire Operational Taxonomy		2	2	2	6	4	2	2
	Empire Air Attack Forces	3	3	↗		↗	↗		
	Empire Air Transport Forces								
	Empire Command	2	1			↗			↗
	Empire Forces	8	7	↗	↗	↗	↗	↗	↗
	Empire Ground Attack Forces	4	4		↗	↗	↗	↗	
	Empire Scout Forces								
	Empire Space Forces	3	3			↗	↗		

Figura 13. Tabla de mapeo de actores operativos a capacidades.

Se debe tener en cuenta que el Transporte Aéreo del Imperio y las Fuerzas Exploradoras del Imperio no están incluidos, ya que no son necesarios para esta fase. Una vez definidos estos elementos estructurales, sus interacciones se pueden definir utilizando el diagrama de conectividad interna, similar a un diagrama interno de bloques de SysML, como se muestra a continuación en la Figura 14.

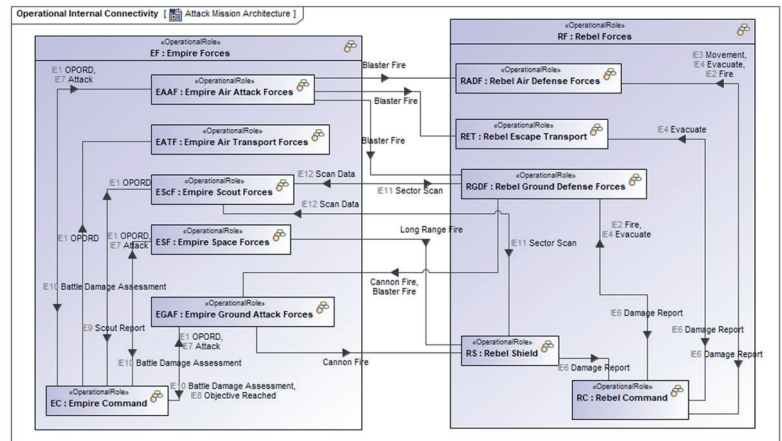


Figura 14. Arquitectura de la misión de ataque.

La arquitectura de la misión muestra los enlaces de comunicación interna, así como los datos de disparo y escaneo de armas entre las fuerzas opuestas. Las Fuerzas Rebeldes son aquellas que fueron identificadas durante la misión de reconocimiento. Mostrar estas interacciones garantiza que la potencia de fuego y los recursos tácticos necesarios estén disponibles para la misión.

3.2. Actividades operativas

El Imperio identifica las actividades operativas aplicables y que cada elemento de la Arquitectura Operativa podría realizar, en el contexto de una misión de exploración, desembarco o ataque en apoyo de una invasión planetaria. Estos se asignan entre sí y corresponden al comportamiento definido para estos elementos en la Figura 15. Los elementos rebeldes e imperios se modelan en diagramas separados.

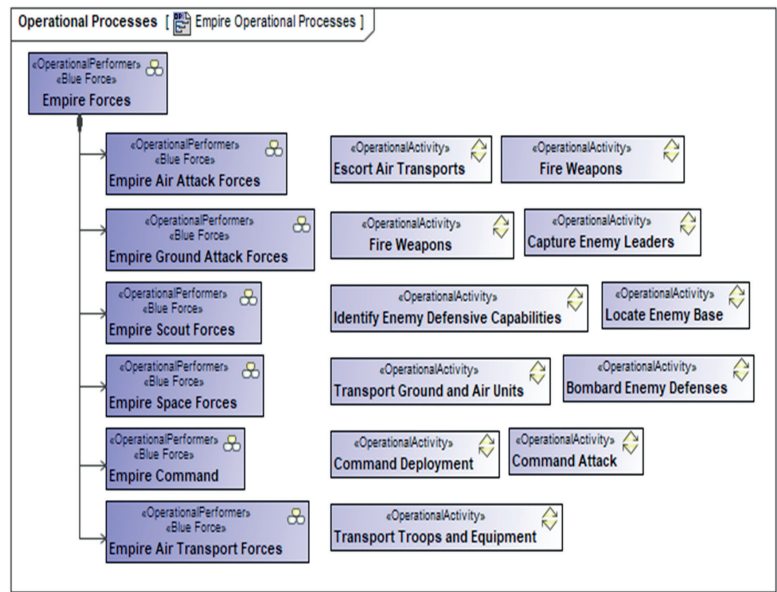


Figura 15. Fuerzas y actividades del Imperio.

Estos definen las actividades que pueden ser realizadas por las fuerzas del Imperio. Estos se utilizarán durante la ejecución de la misión. En la Figura 16 se define la funcionalidad para las Fuerzas Rebeldes. Este conjunto de actividades defensivas se evaluará frente a las actividades ofensivas del Imperio.

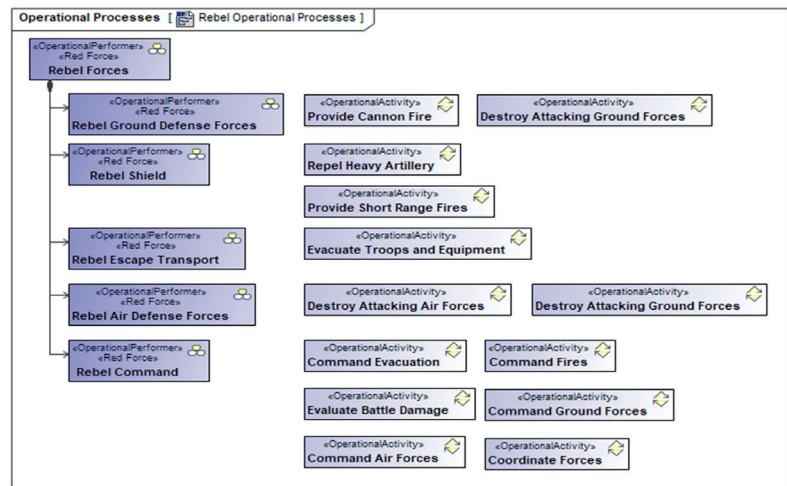


Figura 16. Actores y actividades rebeldes.

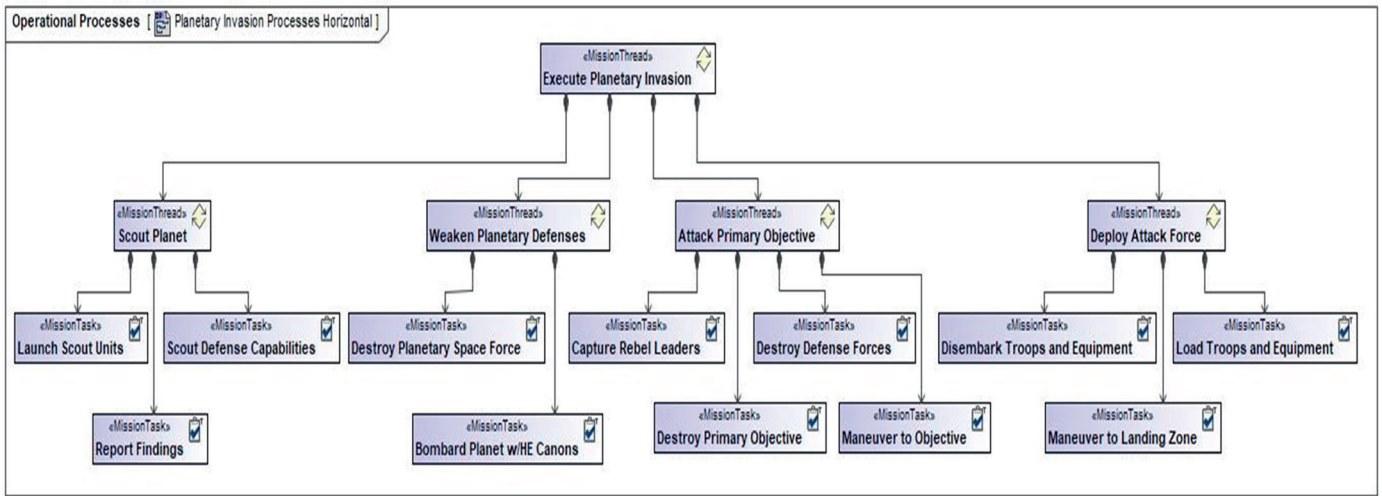


Figura 17. Hilos y tareas de la misión de la invasión planetaria.

3.3. Definición de subproceso de misión

La doctrina del Imperio establece los Hilos de Misión y las Tareas de Misión para cada Fase de la Misión. La Figura 17 muestra el desglose de los Hilos de Misión y las Tareas de Misión para la Fase de Ejecución de una Invasión Planetaria. Se divide en Hilos de Misión de Reconocer el Planeta, Debilitar las Defensas Planetarias, Atacar el Objetivo Principal y Desplegar Fuerza de Ataque. Cada una de ellas se descompone a su vez en Tareas de Misión.

Para cada Hilo de Misión y Tarea de Misión se desarrolla un Diagrama de Actividades Operativas para describir qué Actividades Operativas se necesitan para llevar a cabo cada Hilo de Misión, y qué parte de la Arquitectura Operativa se espera que las realice. Estos se definieron en la Figura 15. Progresivamente, se deben crear bibliotecas de estos para reducir al mínimo el tiempo necesario para la ingeniería de la misión y garantizar la exactitud y el cumplimiento de las normas y la doctrina. La Figura 18 muestra la descripción del diagrama del proceso operacional "Destruir las Fuerzas de Defensa".

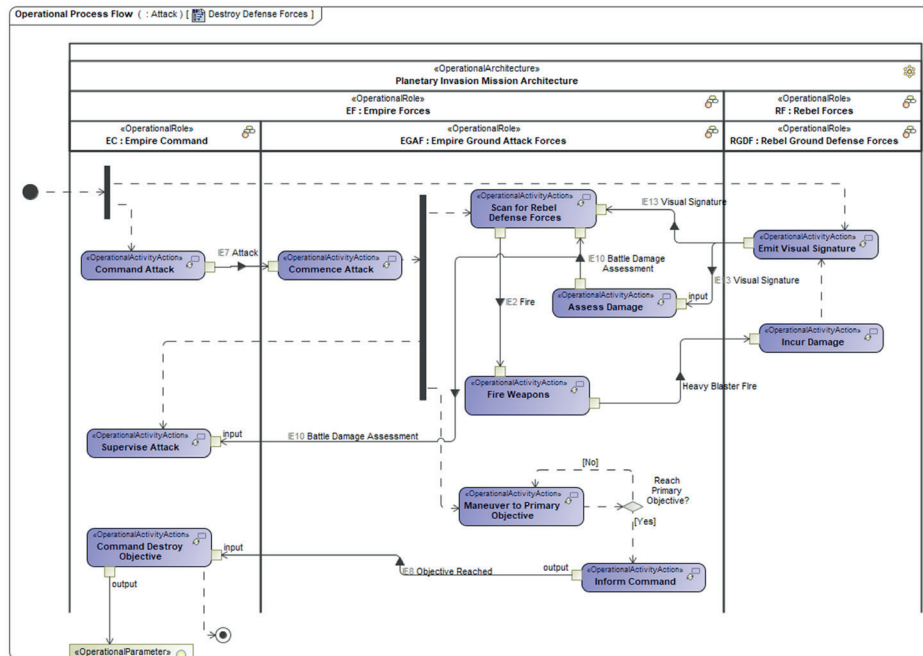


Figura 18. Destruir las fuerzas de defensa.

La Figura 19 muestra la Trazabilidad Estratégica entre los Hilos de Misión, las Tareas y las Misiones Reales. Esto demuestra que las capacidades se abordan con los elementos funcionales. Sólo se muestra la misión de ataque para simplificar.

3.4. Diferenciación entre Enemigo/Amigo/Neutral

Los modelos Ingeniería de Misión requieren la identificación de diferentes fuerzas como enemigas (rojas), amigas (azules), neutrales, etc. Esto puede corresponder a elementos individuales, así como a organizaciones y grupos. La forma más útil de lograr esto es a través de un conjunto de estereotipos que permitan rastrear estos elementos fácilmente dentro del modelo, así como permitir un formato único (por ejemplo, colores) que los identifique claramente en los diagramas. El Perfil Ingeniería de Misión integra 5 de estos estereotipos de Designación de Fuerza, con un estereotipo general de que se especializan, como se ve en la Figura 20. Esto también permite a los responsables del modelado añadir estereotipos adicionales simplemente heredando del estereotipo general de Designación de Fuerza. El término "Designación de Fuerza" fue elegido porque el término "Tipo de Fuerza" implica Ejército, Armada, Fuerza Aérea, etc., y podría ser confuso. Estos tipos podrían ser añadidos por un ingeniero para ampliar el perfil.

Legend		Attack Capabilities				
↗	Exhibits Capability	Close Air Attack	Enemy Force Capture	Ground Attack	Planetary Attack	Planetary Maneuver
↗	Maps To Capability					
↗	Maps To Capability (Implied)					
Hoth Missions					1	
○	Execute Hoth Invasion Mission : Execution Phase					
○	Hoth AMEP : Execution Phase	1			↗	
○	Hoth AMPP : Planning Phase					
○	Hoth EP : Execution Phase					
○	Hoth LMPEP : Execution Phase					
○	Hoth LMPP : Planning Phase					
○	Hoth PP : Planning Phase					
○	Hoth SMEP : Execution Phase					
○	Hoth SMPP : Planning Phase					
○	Plan Hoth Invasion Mission : Planning Phase					
Planetary Invasion Operational Processes [Operations]		6	3	4	9	3
⚡	Attack Primary Objective	5	↗	↗	↗	↗
☑	Bombard Planet w/HE Canons	2			↗	↗
☑	Capture Rebel Leaders	2		↗		
⚡	Deploy Attack Force					
☑	Destroy Defense Forces	3	↗		↗	
☑	Destroy Planetary Space Force	2	↗		↗	
☑	Destroy Primary Objective	3	↗		↗	
☑	Disembark Troops and Equipment					
⚡	Execute Planetary Invasion	6	↗	↗	↗	↗
☑	Launch Scout Units					
☑	Load Troops and Equipment					
☑	Maneuver to Landing Zone					
☑	Maneuver to Objective	2			↗	↗
☑	Report Findings					
☑	Scout Defense Capabilities					
⚡	Scout Planet					
⚡	Weaken Planetary Defenses	3	↗		↗	↗

Figura 19. Trazabilidad estratégica entre los hilos de la misión, las tareas y las misiones reales.

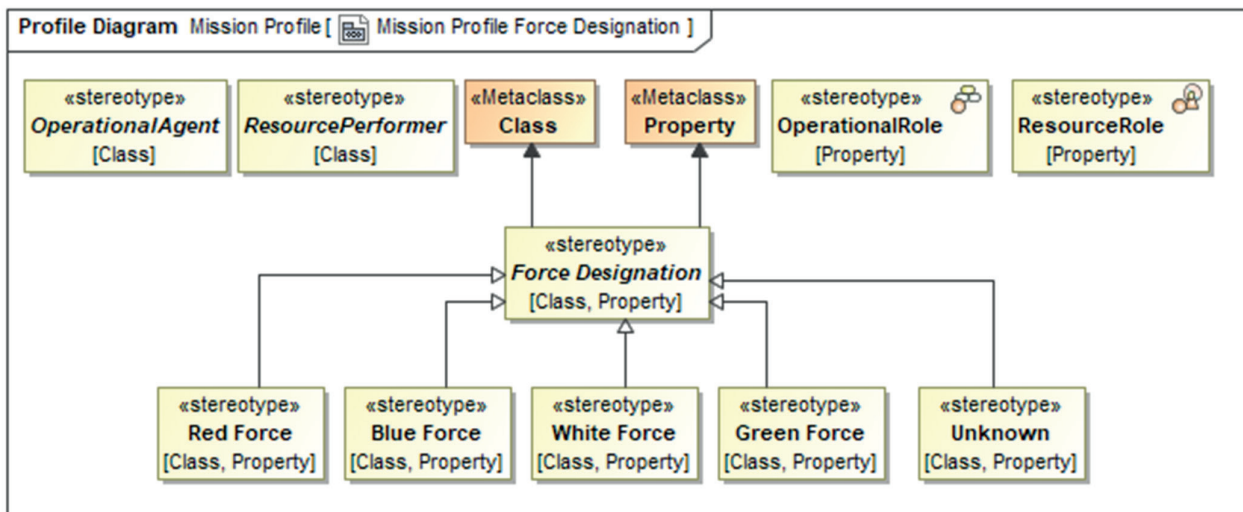


Figura 20. Diagrama de perfil de definición de designación de fuerza.

La Figura 21 muestra las fuerzas opuestas del Imperio y las de los Rebeldes. Las fuerzas rebeldes se muestran en la parte superior en rojo. Las fuerzas del Imperio se muestran en azul. Las designaciones de fuerza se pueden aplicar a la definición que se muestra aquí o a los elementos de rol en un diagrama de conectividad interna. Otras designaciones de fuerza pueden incluir civiles, operaciones comerciales, aliados, etc. La *relación de oposición* definida originalmente en la Figura 8 muestra los elementos de la misión que competirán/atacarán/lucharán entre sí.

3.5. Metas, objetivos, efectos y resultados

Como se mencionó anteriormente, las metas y los objetivos se modelan como tipos de requisitos. Como tales, poseen identificadores únicos, se pueden anidar y pueden hacer uso de todas las relaciones que se ofrecen a los requisitos. En la Figura 10 se enumeran las metas y objetivos de la misión de ataque. En la Figura 22 se muestra una parte del hilo de misión de Ejecutar Invasión Planetaria. Las diferentes tareas de la misión satisfacen sus resultados y objetivos. De esta manera, se demuestra que una solución de Ingeniería de Misión también debe ser capaz de alcanzar las metas y objetivos.

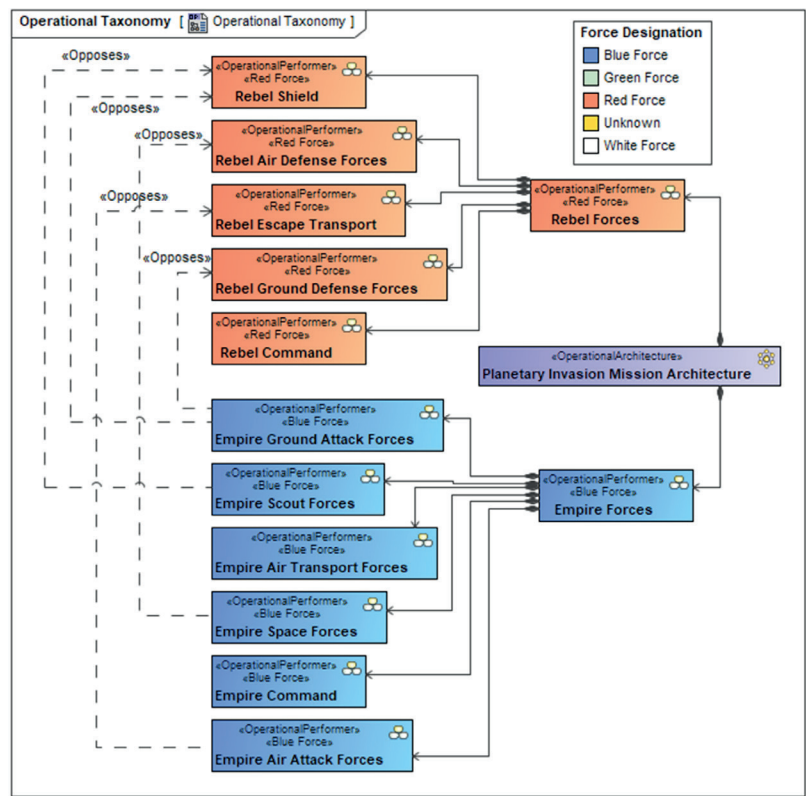


Figura 21. Elementos rojos y azules de los actores de la misión.

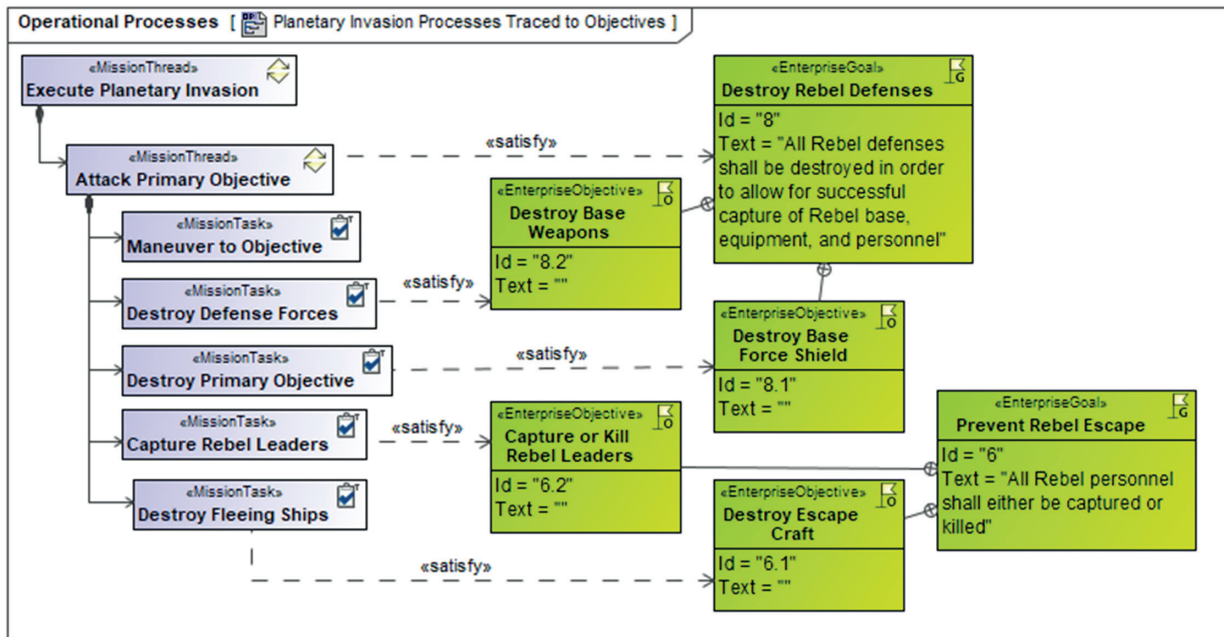


Figura 22. Hilos de misión y tareas vinculadas a metas y objetivos.

La Figura 23 muestra los objetivos de la misión de recopilación de inteligencia, así como los efectos, los resultados y los sistemas que generarán esos efectos y resultados. Los efectos y los resultados son secuencias correspondientes al despliegue de aviones no tripulados y espías, la recopilación de información, la detección de señales, el análisis de esas señales y, finalmente, la síntesis de esa inteligencia para mostrar la ubicación de la base rebelde.

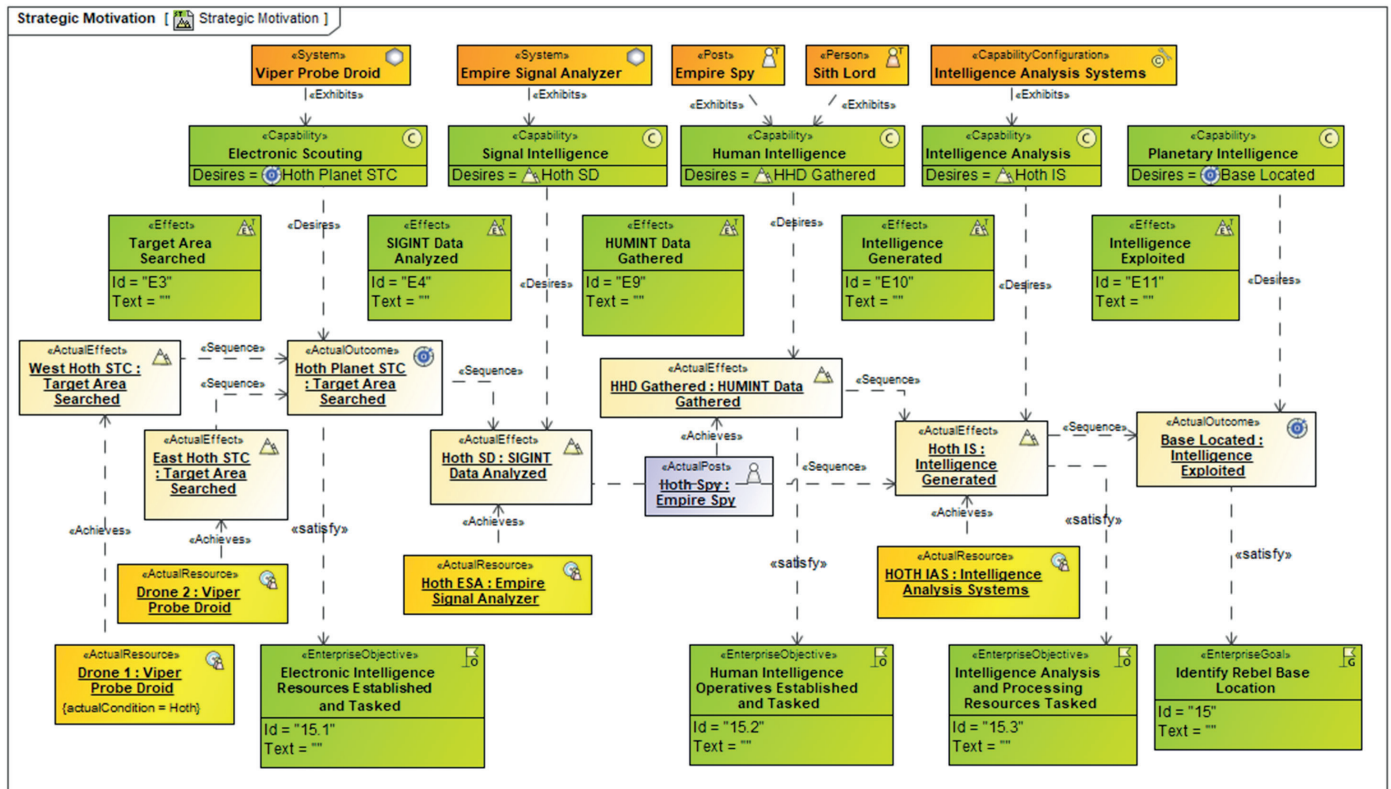


Figura 23. Objetivos, efectos, resultados y elementos de logro de la misión.

3.5.1. Procedencia/Confianza de los recursos enemigos

Las capacidades, las fuerzas, las actividades, la fuerza de combate, etc. de las fuerzas rebeldes han sido descubiertas a través de los servicios de inteligencia. Dos aspectos de cualquier inteligencia son la procedencia de la información y el nivel de confianza en la información, así como en la fuente. La estructura y los elementos conductuales creados en base a esa inteligencia deben referirse a la fuente (procedencia) y a la confianza correspondiente. La Figura 24 muestra una parte de las fuerzas rebeldes. La información sobre las fuerzas rebeldes ha sido recopilada por aeronaves no tripuladas y espías. Se han definido enumeraciones para el perfil que proporciona Intel Confidence e Intel Type. Los atributos corresponden a los elementos del recurso. Se crean instancias de ellas como una capacidad de campo y valores específicos asociados a las mismas. En este caso, la inteligencia fue recopilada por un droid de inteligencia, el tipo es desconocido y el nivel de confianza es medio.

3.5.2. Cumplimiento/Conformidad a la Doctrina/Normas.

El elemento de requisito de SysML se ha extendido para proporcionar conceptos Ref Doctrine, Ref Publication y Ref Standard (no se muestran por razones de espacio). Estos proporcionan la capacidad de vincular pasos específicos en un hilo de misión o en un hilo de ingeniería de misión, elementos de la misión o toda la misión con elementos atómicos de doctrina. Esto puede ser crucial para garantizar que se sigan los procedimientos adecuados al construir las misiones. El concepto de estándar de UAF también está disponible, pero normalmente se encuentra en un nivel más "macro" que se ocupa de un documento completo.

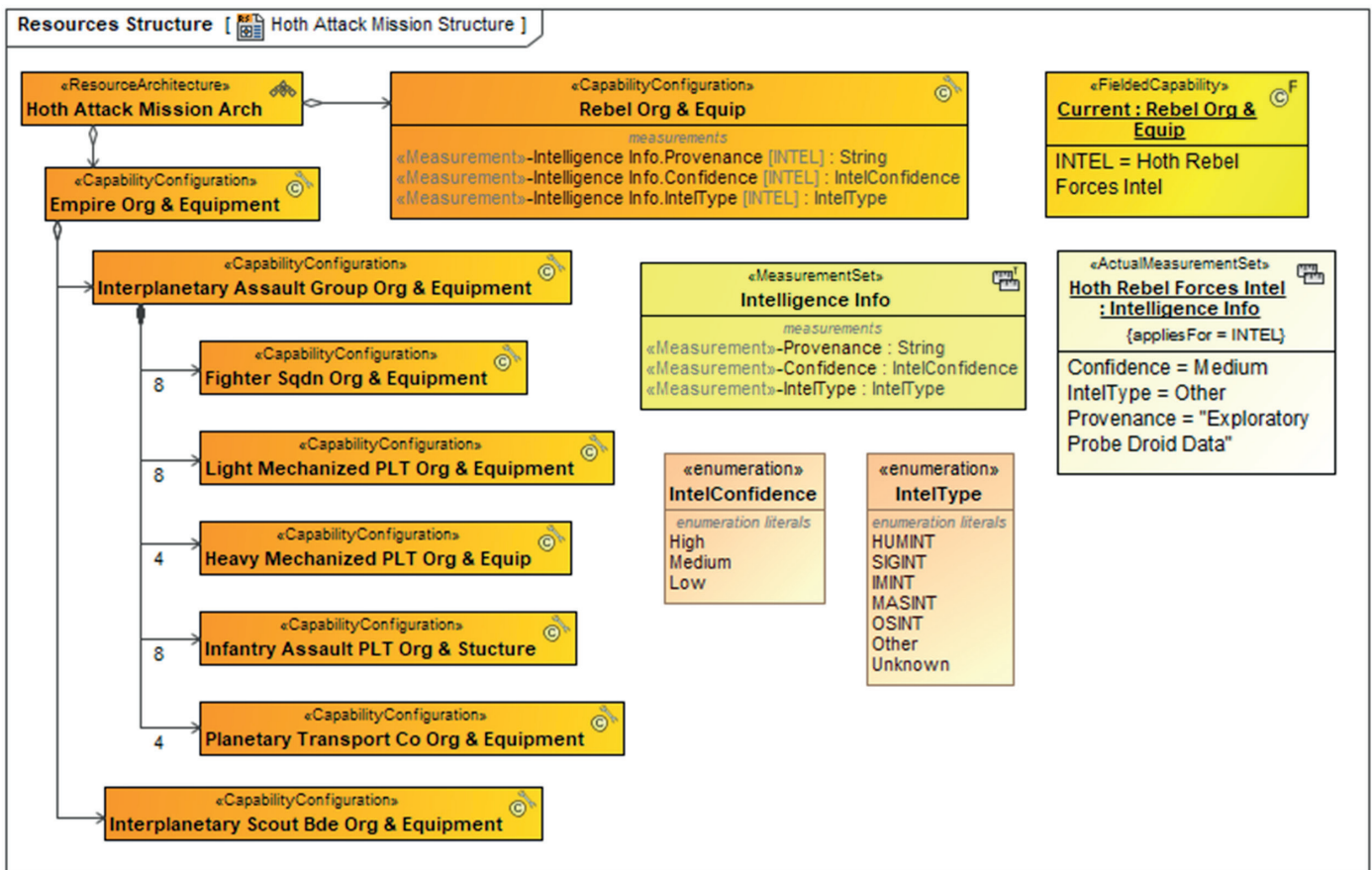


Figura 24. Elementos de recursos de la misión e información de inteligencia.

4. DESARROLLO DE LA ARQUITECTURA DE RECURSOS

4.1. La arquitectura de los recursos de las fuerzas amigas (Fuerzas Azules)

Los inicios de la Arquitectura de Recursos se vieron en la Figura 23, que muestra una parte de las fuerzas rebeldes involucradas en la batalla. Para comprender completamente cómo se desarrolla la misión, los diversos recursos (es decir, sistemas, software, personal) que se utilizan para respaldar las tareas de la misión en los subprocesos de la misión se capturan en los subprocesos de ingeniería de misión (HIM) que forman parte de la arquitectura de recursos. Esto se hace para ambos bandos de la batalla: para las Fuerzas Azules (de nuestro lado, el Imperio en este caso) y las Fuerzas Rojas (que son los adversarios, los Rebeldes en el caso de este modelo de ejemplo).

4.2. Funciones de subprocesos de ingeniería de misión

Los diversos pasos del hilo de ingeniería de misión están definidos para la Invasión Planetaria Ejecutar Hoth. Las funciones pueden ser parte del hilo de ingeniería de la misión, o pueden ser funciones realizadas por recursos, ahora que hemos identificado algunos de estos recursos. La Figura 24 muestra el subproceso de ingeniería de la misión a la izquierda y el subproceso de la misión a la derecha. Las funciones en el hilo de ingeniería de misión “implementarán” las actividades operativas en el hilo de misión. Este modelo muestra dos deficiencias de la arquitectura:

- 1) El hilo de misión para debilitar la defensa planetaria no tiene funciones de recursos (sistema) correspondientes que implementen esta tarea de la misión (es decir, la actividad operativa).
- 2) El hilo de ingeniería de misión para destruir las defensas rebeldes clave de Hoth tiene una función de proteger a las Fuerzas Terrestres del Imperio que no tiene una tarea de misión correspondiente, es decir, hay un proceso operativo faltante que debe realizarse pero que aún no se ha contabilizado.

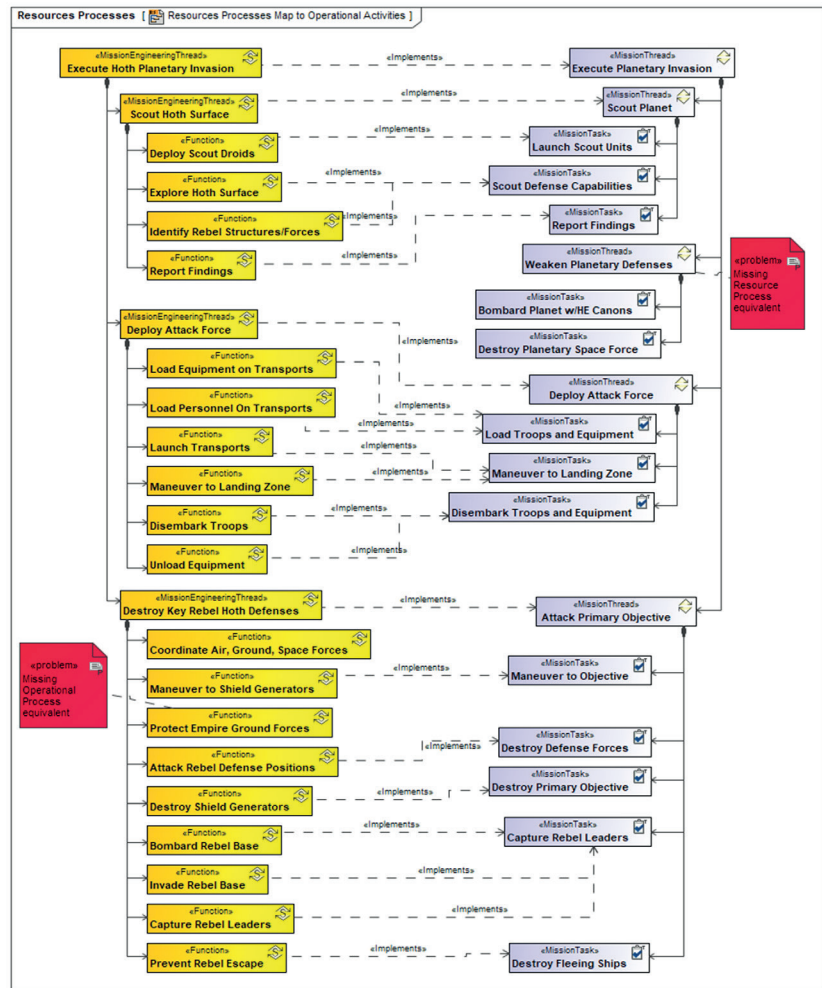


Figura 25. Hilo de ingeniería de misión para la fase de misión Ejecutar invasión planetaria.

4.3. Despliegue del flujo de proceso de Scout Droids

La Figura 26 ilustra un hilo de ingeniería de misión relativamente complicado para los diversos elementos de recursos que despliegan droides exploradores. En este flujo de proceso, los elementos humanos y técnicos se despliegan al recibir el Scout OPORD. La Figura 27 muestra un flujo de proceso relacionado que describe las actividades para prepararse para la exploración y el envío de informes de estado.

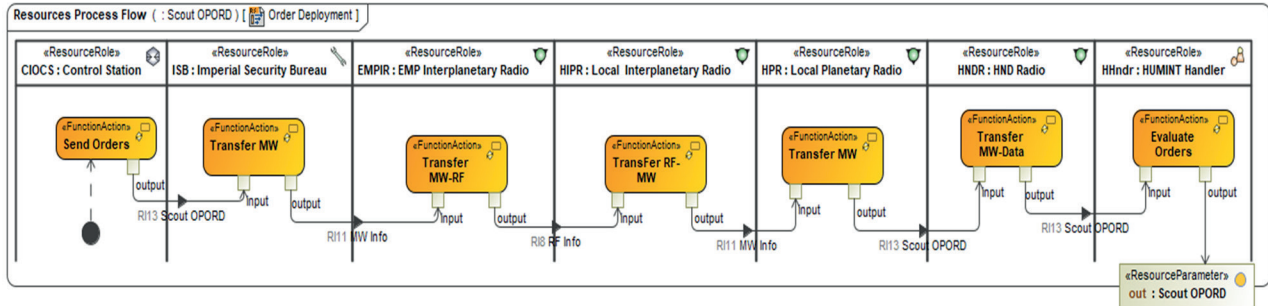


Figura 26. Despliegue del flujo de proceso de Scout Droids.

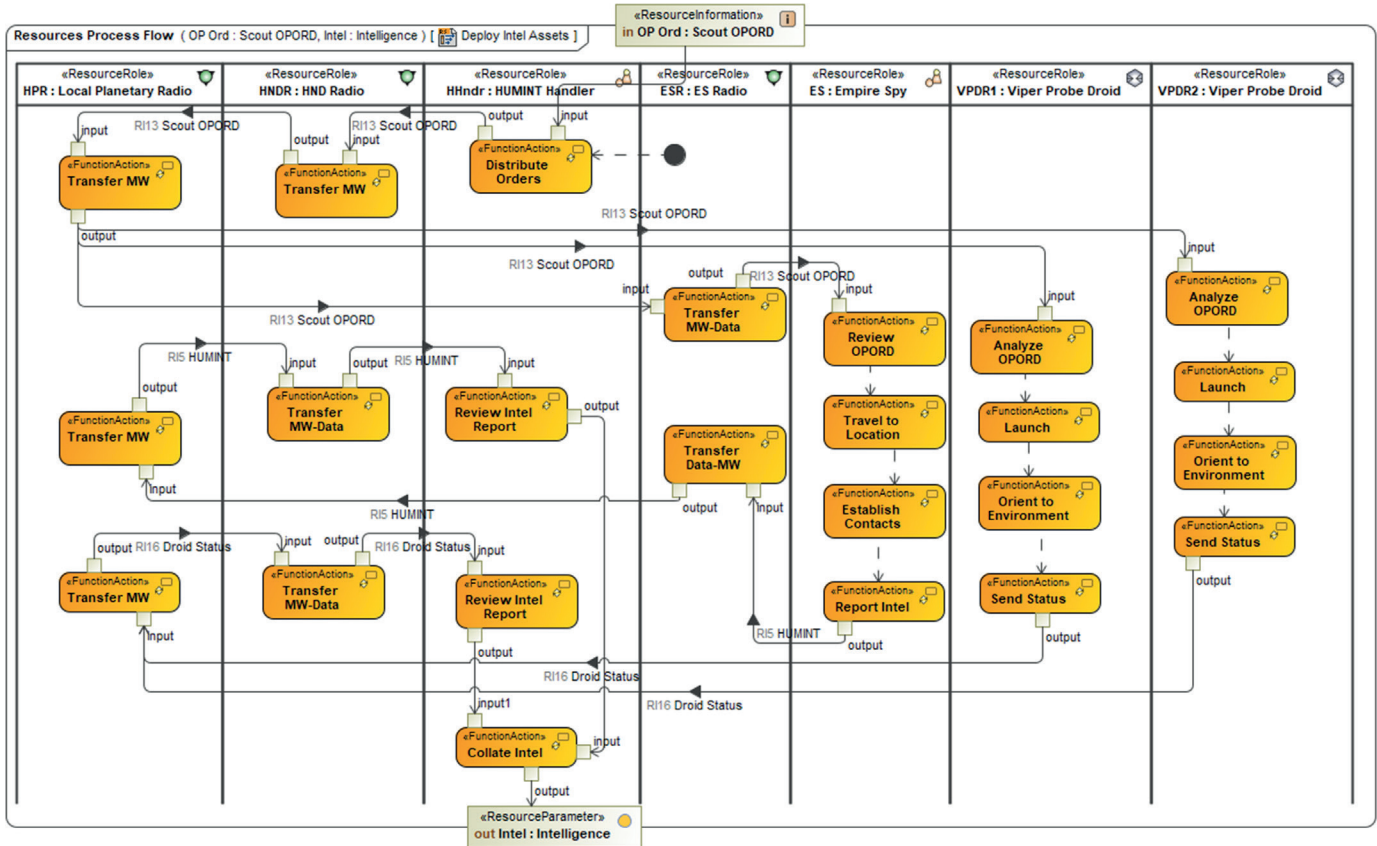


Figura 27. Flujo del proceso de exploración y presentación de informes de estado.

5. OTRAS CONSIDERACIONES

5.1. Problemas de los sistemas de sistemas

Los sistemas que participan en los hilos de ingeniería de misión que implementan los hilos de misión en la arquitectura de la misión suelen ser desarrollados por diferentes organizaciones y no siempre se desarrollan teniendo en cuenta estos hilos de misión en particular. Estos sistemas también suelen ser operados por diferentes organizaciones involucradas en la realización de la misión. Por lo tanto, desde la perspectiva de los sistemas de sistemas, esto ilustra la independencia gerencial y la independencia operativa de los sistemas que componen el sistema de sistemas para cada uno de los hilos de ingeniería.

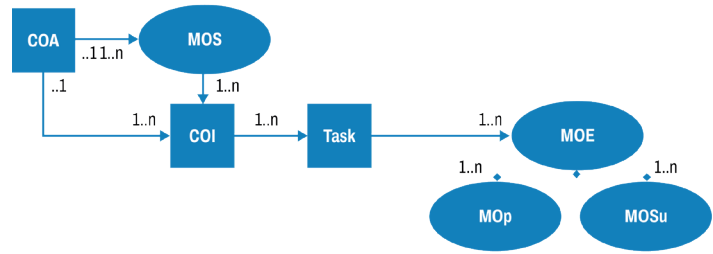


Figura 28. Relación entre las cuestiones operacionales críticas y las medidas de eficacia.

5.2. Mediciones clave y trazabilidad

Un modelo bien definido contendrá medidas cuantificables para el éxito. Como mínimo, describirá cómo medir un resultado exitoso para las diversas fases de la misión y las tareas esenciales de la misión. Estos se modelarán en los niveles de la misión, la fase de la misión, el hilo de ingeniería de misión y las tareas esenciales de la misión. Para un modelo bien formado, estas mediciones deben coincidir, vincularse y rastrearse entre sí, desde las medidas de rendimiento (MOP, por sus siglas en inglés) a nivel de sistema centradas en el rendimiento de los sistemas constituyentes individuales, hasta las MOE que definen el éxito de la misión en cada tarea esencial de la misión, hasta las medidas de éxito (MOS, por sus siglas en inglés) que definen el estado final deseado (consulte la Figura 27).

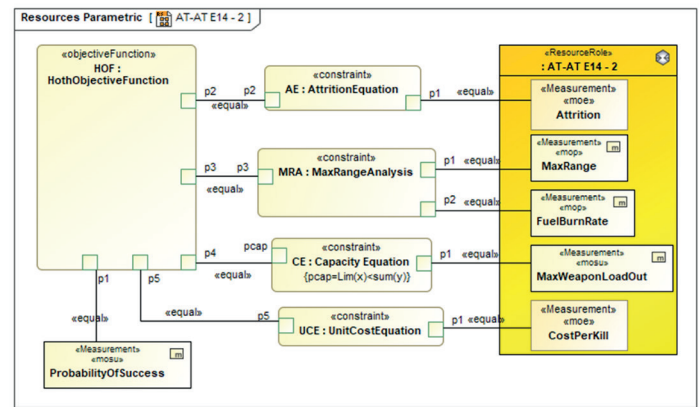


Figura 29. MOE típicas a través de la perspectiva de la Ingeniería de Misión en un Sistema de Sistemas.

Estas medidas clave deben fluir desde las MOS y desembocar en las MOE de los que se pueden derivar las MOP y las medidas de idoneidad (MOSu, por sus siglas en inglés). Sin embargo, para rastrear estas medidas a través de un sistema de sistemas, deben estar vinculadas a declaraciones de importancia. En este caso, dado que estamos en el ámbito militar, podemos referirnos a estas declaraciones como cuestiones operativas críticas (COI, por sus siglas en inglés). Para cada uno de estas COI puede haber de una a MOS, MOE y de una a muchas MOP y MOSu, como se muestra en la Figura 28. Este diagrama es puramente nocional, ya que estas métricas son en gran medida estocásticas, sólo estimables a través de una simulación avanzada, y a menudo representan propiedades emergentes de un sistema de sistemas complejo, lo que hace que un desarrollo (“roll-up”) determinista sea poco práctico o imposible. En la Figura 29 se da un ejemplo de la misión Hoth.

5.3. Todos los modelos son incorrectos, algunos modelos son útiles...

Incluso un modelo de arquitectura de recursos relativamente simple requiere mucho tiempo y esfuerzo para desarrollarse, si todo en la arquitectura está modelado. Al igual que con cualquier modelo, comprender qué preguntas debe responder el modelo, qué información está disponible para modelar y qué recursos tiene disponibles para realizar el modelado (personas, tiempo, dinero y herramientas) ayudará a enmarcar lo que se debe modelar. Por supuesto, una vez que se ha decidido el alcance del modelado, cualquier cambio en el alcance del modelado debe estar bien gestionado; de lo contrario, se introducirá un riesgo no deseado para el desarrollo de un modelo útil.

Es probable que no se conozca todo el alcance del esfuerzo de modelado, ya que es posible que la información necesaria no esté disponible al comienzo del esfuerzo de modelado o que se produzcan cambios significativos en la arquitectura no planeados. Identificar los riesgos de modelado desde el principio es clave para gestionar el esfuerzo de modelado y mantener su utilidad. Se recomienda encarecidamente que, antes de comenzar un esfuerzo de modelo, se dedique algún tiempo a realizar un ejercicio de definición de problemas.

Las organizaciones tendrán que determinar qué bibliotecas de modelos desean desarrollar, compartir y mantener. Aunque hay muchas maneras de separar los modelos de ingeniería de misión en información reutilizable y específica de cada caso, UAF ya segmenta la información del modelo de tal manera que simplemente se podrían crear modelos separados basados en los paquetes de nivel superior: Estrategia, Operativo, Servicios, Personal, Recursos, Seguridad, Proyectos, Estándares y Recursos Reales. Por supuesto, un enfoque de biblioteca de modelos tendrá que ser específico para la forma en que una organización quiere hacer el modelado. Se debe diseñar un plan de federación de modelos, incluso uno simple, antes del inicio del modelado para ayudar a dividir el modelo grande en proyectos de modelado más pequeños para facilitar la gestión y la gobernanza del modelo. Esto también ayuda a mejorar el tiempo de consulta del modelo, reducir los conflictos de acceso al modelo entre los miembros del equipo, permitir un mayor control sobre los cambios del modelo y el control de la configuración.

6. CONCLUSIONES

El modelado de la arquitectura de una misión es un facilitador fundamental para una ingeniería de misión eficaz. Este capítulo ha demostrado cómo una arquitectura de misión bien estructurada, basada en el Marco de Arquitectura Unificada (UAF), admite el análisis riguroso, la trazabilidad y la coordinación necesarios para ejecutar misiones complejas que involucran a múltiples partes interesadas, escenarios y sistemas de sistemas. Al integrar sistemáticamente la intención estratégica, las actividades operativas y las capacidades de los recursos a través de constructos como los hilos de misión (HM) y los hilos de ingeniería de misión (HIM), UAF proporciona un lenguaje de modelado cohesivo para describir y analizar arquitecturas de misión.

Usando la Batalla de Hoth como ejemplo, el capítulo ilustró cómo el modelado de ingeniería de misión no solo aclara las relaciones entre metas, objetivos y elementos operativos, sino que también revela riesgos, conflictos y dependencias inherentes a la misión. Se ha destacado la importancia de las construcciones contextuales, como los escenarios y las viñetas, y cómo se pueden ampliar dentro de la UAF para proporcionar conexiones significativas entre las condiciones, los comportamientos y la ejecución de la misión.

Además, el uso de medidas (MOS, MOE, MOP) y el énfasis en la trazabilidad desde los impulsores (drivers) de alto nivel hasta las actividades de recursos tácticos garantizan que el modelo de misión pueda servir como base para el análisis, la evaluación y la mejora iterativa. La incorporación de la procedencia y la confianza en los datos de inteligencia, el cumplimiento de la doctrina y la consideración de las designaciones de fuerza mejora aún más la fidelidad y el realismo del modelo.

En última instancia, el modelado de la arquitectura de una misión no trata de producir una representación perfecta, sino de construir una útil; una que pueda informar las decisiones, identificar vulnerabilidades y guiar acciones efectivas. Los principios y constructos presentados en este capítulo ofrecen un enfoque escalable y repetible para aplicar el modelado de arquitectura de misión en contextos de defensa, aeroespacial y empresarial.

REFERENCIAS

1. OMG 2022, Marco de arquitectura unificada, versión 1.2, Grupo de gestión de objetos, <https://www.omg.org/spec/UAF/About-UAF/>.
2. OMG 2022, Lenguaje de modelado de marco de arquitectura unificada, versión 1.2, Grupo de gestión de objetos.
3. OMG 2022, Guía de arquitectura empresarial para el marco de arquitectura unificada (informativo), versión 1.2, Grupo de gestión de objetos.
4. Gagliardi M., Hause M., "Implementación de la ingeniería de misión con UAF" en Actas del Simposio de Ingeniería y Tecnología de Sistemas de Vehículos Terrestres (GVSETS), NDIA, Novi, MI, 16-18 de agosto de 2022.
5. Gagliardi M., Hause M., Martin J., Phillips, M, 2024, "El arma secreta de Darth Vader: implementación de la ingeniería de misión con UAF" presentado en el Simposio Internacional INCOSE, Dublín, Irlanda.
6. Martín, J., & Álvarez, K. (2023). Uso del Marco de Arquitectura Unificada en apoyo de las actividades de ingeniería de la misión. Simposio Internacional INCOSE.
7. Fandom, 2023, Battle of Hoth, Información de referencia Disponible en línea en https://starwars.fandom.com/wiki/Star_Wars:_Episode_V_The_Empire_Strikes_Back#The_Battle_of_Hoth
8. DoD, 2023, DoD OSD Mission Engineering Guide (MEG), disponible en línea en https://ac.cto.mil/wp-content/uploads/2023/11/MEG_2_Oct2023.pdf consultado en noviembre de 2023.

BIOGRAFÍAS

MATTHEW GAGLIARDI

Matthew Gagliardi tiene más de veinte años de experiencia en ingeniería de sistemas en los sectores de automoción y defensa. Su experiencia incluye el desarrollo de vehículos, la integración de sistemas, los procesos y herramientas de ingeniería de sistemas y el desarrollo de nuevas tecnologías. Actualmente, Matt brinda apoyo y orientación de MBSE a múltiples programas de Sistemas de Combate Terrestre de PEO, así como al Centro de Sistemas de Vehículos Terrestres de DEVCOM. Matt recibió una licenciatura en Ingeniería Mecánica y una maestría en Ingeniería Mecánica en la Universidad de Purdue. Matt ha alcanzado el nivel 3 de SPRDE de la Universidad de Adquisición de Defensa y es un constructor de modelos profesional de modelado de sistemas certificado por OMG y ha sido un CSEP de INCOSE. Su papel en SSI incluye consultoría, mentoría, presentaciones en conferencias y desarrollo y presentación de cursos de capacitación.



MATTHEW HAUSE

Matthew Hause es ingeniero principal de SSI, presidente del grupo UAF, miembro del equipo de especificación OMG SysML, copresidente del equipo de especificación de activos reutilizables (RAS) de OMG y líder de opinión en el uso de MBSE. Lleva más de 45 años desarrollando sistemas complejos multinacionales como ingeniero de sistemas y software. Comenzó trabajando en la industria de sistemas de energía y luego pasó a sistemas de mando y control, control de procesos, comunicaciones, SCADA, control distribuido, sistemas militares y muchas otras áreas de sistemas técnicos y en tiempo real. Es autor de más de 100 artículos técnicos sobre una amplia gama de temas. Su papel en SSI incluye consultoría, mentoría, desarrollo de estándares, presentaciones en conferencias y desarrollo y presentación de cursos de capacitación. También es un orgulloso ganador del premio INCOSE MBSE Propeller Hat Award.



DR. JAMES N MARTIN

El Dr. James N Martin es miembro del Grupo de Trabajo de Revisión de la UAF con OMG y fue editor principal de la norma ISO 42020 sobre Procesos de Arquitectura. Es arquitecto empresarial e ingeniero de sistemas distinguido en The Aerospace Corporation, desarrollando soluciones para sistemas de información y para empresas de dominio espacial. Fue uno de los autores clave del proyecto BKCASE en el desarrollo de artículos de Ingeniería de Sistemas Empresariales para el SE Body of Knowledge (SEBOK). El Dr. Martin dirigió el grupo de trabajo responsable del desarrollo de ANSI/EIA 632, una norma nacional de EE. UU. que definió los procesos para la ingeniería de un sistema. Anteriormente trabajó para Raytheon Systems Company y AT&T Bell Labs en sistemas aéreos y submarinos y en sistemas de telecomunicaciones móviles. Su libro, Systems Engineering Guidebook, fue publicado por CRC Press en 1996. El Dr. Martin es fellow de INCOSE y fue líder del Comité Técnico de Normas. Fue el fundador y hasta hace poco líder del Grupo de Trabajo de Ciencias de Sistemas. Recibió de INCOSE el Premio Fundadores por sus largos y distinguidos logros en el campo.



VÍCTOR RAMOS DEL POZO

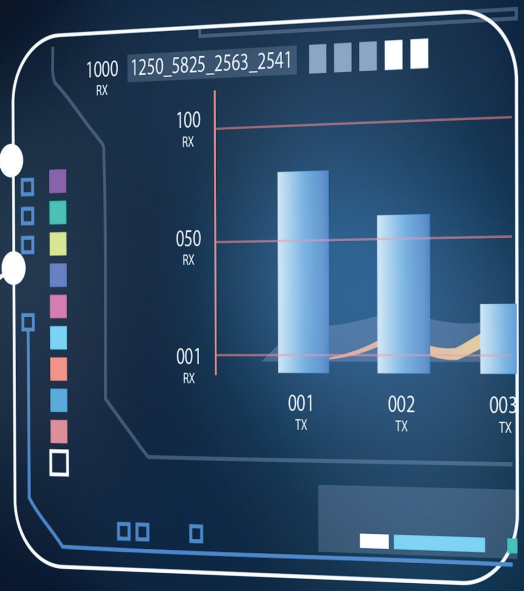
Víctor Ramos del Pozo es ingeniero de sistemas en Ingeniería de Sistemas para la Defensa de España (Isdefe) que actualmente presta asistencia técnica en la Oficina de Programa Española del Sistema de Armas de Última Generación (NGWS, Next Generation Weapon System) dentro del entorno de un Futuro Sistema de Combate Aéreo (FCAS, Future Combat Air System), especialmente en los ámbitos de la nube de combate, sensores colaborativos, simulación y sistemas de sistemas. Anteriormente prestó asistencia técnica en la Dirección General de Armamento y Material de la Secretaría de Estado de Defensa en los ámbitos del análisis de la base industrial de defensa, el planeamiento de la obtención de armamento y material, y en la gestión de grandes programas de defensa.



Antes de unirse a Isdefe, Víctor trabajó como ingeniero de sistemas en EADS-CASA (actual Airbus Defence and Space) en los programas de aviones de reabastecimiento en vuelo MRTT y FSTA, así como en Indra en el departamento de sistemas de identificación amigo-enemigo (IFF).

Es miembro del Consejo Internacional de Ingeniería de Sistemas (INCOSE) y es formador interno en Isdefe en los ámbitos de la ingeniería de sistemas y de la gestión de programas.

En 2015 le fue concedida la Cruz del Mérito Aeronáutico con distintivo blanco por su trabajo apoyando a la Dirección General de Armamento y Material.



Diseño e implementación de la gobernanza de Sistemas de Sistemas

Dr. Polinpapilinho F. Katina, *Universidad de Carolina del Sur Upstate* (pkatina@uscupstate.edu)

Dr. Charles B. Keating, *Universidad Old Dominion* (ckeating@odu.edu)

César Heras Menor de Gaspar, *Isdefe* (chmenor@isdefe.es)

Víctor Ramos del Pozo, *Isdefe* (vramos@isdefe.es)

Resumen

Dada la independencia y autonomía de los sistemas que constituyen un sistema de sistemas (SdS), la gobernanza supone un aspecto crucial de su funcionamiento. Básicamente, la gobernanza proporciona la dirección, la supervisión y la definición de responsabilidades de un SdS para el cumplimiento de su propósito/misión. Este capítulo ofrece una introducción al diseño e implementación de la gobernanza de un SdS, abordando tres temas principales. En primer lugar, se proporciona una perspectiva sistémica de la gobernanza de un SdS. En segundo lugar, se establece un marco de última generación basado en la teoría de sistemas para la gobernanza de un SdS: la Gobernanza de Sistemas Complejos. En tercer lugar, se explora la aplicación de este marco. El capítulo concluye con la identificación de los desafíos y las guías de aplicación para los profesionales responsables del diseño y la implementación de la gobernanza de los SdS.

Palabras clave

Gobernanza de Sistemas de Sistemas, SdS, Marco, Teoría de Sistemas, Sistemas Complejos, Gobernanza.



1. INTRODUCCIÓN

La “gobernanza” tiene por objetivo la gestión activa de los SdS a través de un diseño, una ejecución y una evolución de los mismos de forma ingeniosa e integrada [1]. Por tanto, la principal finalidad de este capítulo es describir el rol y las contribuciones que la gobernanza puede aportar al diseño y la implementación de los SdS.

En este capítulo, se abordan tres temas principales. Primero, se expone una perspectiva sistémica de la gobernanza de los SdS, la cual examina la naturaleza y el rol de la gobernanza en relación con el diseño y la implementación de la gobernanza de un SdS. En segundo lugar, se presenta la Gobernanza de Sistemas Complejos (GSC o GCS, por sus siglas en inglés, *Governance of Complex Systems*) como un marco basado en la teoría de sistemas para la gobernanza de un SdS. La GCS se constituye como una evolución de la Ingeniería de Sistemas, centrada en proporcionar un enfoque que respalde una dirección, una supervisión y una definición de responsabilidades más efectivas para unos SdS cada vez más complejos. En tercer lugar, se explora la aplicación de este marco. El capítulo concluye con la identificación de los desafíos y las guías de aplicación para los profesionales responsables del diseño y la implementación de la gobernanza de los SdS.

Con un diseño y una implementación de los SdS en evolución, se aportan ocho características distintivas para la gobernanza de los SdS [2] (Figura 1).

- 1. Holismo:** La gobernanza opera en todos los niveles de un SdS, desde cada profesional individual hasta el conjunto de la empresa. Se considera la necesidad de evitar tanto los fallos técnicos como los no técnicos.
- 2. Amplio espectro:** El enfoque de la gobernanza para los SdS trasciende los estrechos límites de las soluciones tecnológicas. De hecho, el alcance de la gobernanza comprende consideraciones tecnológicas, sociales, humanas, organizativas, de gestión, de recursos, de contexto, comerciales, normativas y políticas.
- 3. Visión Pluralista:** La gobernanza no asume una visión unitaria (única perspectiva) del diseño e implementación de los SdS. En cambio, adopta una visión pluralista (de perspectivas múltiples) para entender que la gobernanza de los SdS debe lidiar con numerosas motivaciones, puntos de vista y objetivos diferentes y potencialmente contradictorios.

- 4. Continuidad:** El desarrollo de la gobernanza es continuo, teniendo en cuenta que la gobernanza de un SdS nunca se encuentra completa y debe perseguirse continuamente.
- 5. Viabilidad:** El desarrollo de la gobernanza debe considerar que existen problemas de diseño y de implementación que escapan a la capacidad de un SdS para ser abordados. Todo SdS tiene limitaciones en cuanto a lo que se puede lograr con una alta probabilidad de éxito.
- 6. Existencia:** Independientemente de su reconocimiento, las funciones de gobernanza de los SdS se realizan en cualquier SdS viable (que siga existiendo). Sin embargo, el estado y el rendimiento de la gobernanza de los SdS podrían no ser los deseados y potencialmente encontrarse en dirección al fracaso.
- 7. Contexto:** El contexto incluye las circunstancias, los factores y las condiciones que influyen en el SdS, así como los que se ven influenciados por él. El contexto en el que se lleva a cabo la gobernanza del SdS es fundamental para facilitarla o limitarla.
- 8. Incorporación:** El diseño y la implementación de la gobernanza de los SdS deben incorporar el conjunto de métodos, herramientas y técnicas disponibles y accesibles para apoyar su desarrollo. Cada SdS es único y dispondrá de diferentes métodos, herramientas y técnicas.



Figura 1. Ocho características distintivas de la gobernanza de un sistema de sistemas.

2. GOBERNANZA EN EL CONTEXTO DE UN SISTEMA DE SISTEMAS

Para un SdS, la gobernanza desempeña un papel fundamental para garantizar la coherencia en la dirección, la supervisión y la definición de responsabilidades a medida que el SdS cumple su misión. En este apartado, se distingue entre gestión y gobernanza y se explora el papel de esta última en el diseño e implementación de SdS.

2.1. Naturaleza y funciones de la gobernanza

El concepto de gobernanza es relativamente novedoso en lo que respecta a los SdS. A grandes rasgos, se entiende como la dirección activa de un sistema mediante el diseño, la ejecución, el mantenimiento y la evolución del SdS [2] de forma ingeniosa e integrada.

La gobernanza presenta similitudes, pero también diferencias, con la gestión. La tabla 1 identifica las principales diferencias entre gestión y gobernanza, junto con las implicaciones para el diseño y la implementación del SdS [2,3].

Característica	Gestión	Gobernanza	Implicaciones para el diseño y la implementación de un SdS
Énfasis	Salidas (tangibles, objetivas, a corto plazo)	Resultados (menos tangibles, subjetivos, a largo plazo)	La determinación de la 'bondad' de la gobernanza no es simple ni directa.
Cuestiones principales de interés	¿Qué? y ¿Cómo?	¿Por qué?	La gobernanza existe en un nivel lógico más alto de desempeño, enfatizando el propósito del sistema.
Foco	Resultados demostrables a corto plazo	Trayectoria futura a largo plazo	El foco de la gobernanza es expansivo y considera cuestiones a largo plazo de importancia más estratégica que operativa.
Factores de éxito	Fácil de definir, medir y seguir	Difícil de definir y medir	Si bien se pueden desarrollar medidas de gobernanza, carecen de precisión.
Horizonte temporal	Corto plazo	Largo plazo	La naturaleza de la gobernanza implica un horizonte temporal mucho más largo.
Proximidad de la acción y la respuesta	Duración limitada entre la acción y la respuesta del sistema	Gran separación entre la acción y la ejecución de la respuesta del sistema	La naturaleza evolutiva del SdS incrementa el tiempo y la proximidad entre las acciones y la ejecución de los resultados.
Incertidumbre	Preocupaciones por la incertidumbre local	Preocupaciones por la incertidumbre global	La gobernanza tiene un nivel de incertidumbre más global.
Estabilidad y emergencia	Estabilidad y comportamientos emergentes a nivel local	Estabilidad y comportamientos emergentes a nivel global	El enfoque global de la gobernanza se centra en los comportamientos emergentes y la estabilidad a un nivel superior.

Tabla 1. Diferencias entre gestión y gobernanza.

Etimológicamente, “gobernanza” (gobernar) tiene sus raíces en el latín “*gubernare*”, que significa dirigir, así como en el griego “*kybernētēs*”, que significa pilotar, y en gobernador (del irlandés guiar, dirigir). Por tanto, a nivel fundamental, la gobernanza consiste en dirigir.

Existen múltiples perspectivas sobre la gobernanza. Sin embargo, para los SdS, se identifican tres esenciales [3] respecto al rol de la gobernanza.

- 1. Centrada en procesos:** La gobernanza se centra en los procesos colectivos de toma de decisiones. Estos procesos se basan en una toma de decisiones formal, consensuada y deliberativa. El principal objetivo de esta perspectiva de la gobernanza es proporcionar procesos eficaces. Estos procesos permiten que se lleven a cabo las acciones de gobernanza.
- 2. Centrada en la estructura:** La gobernanza enfatiza la definición y la ejecución de estructuras. Estas estructuras preservan el orden y la continuidad, a la vez que guían al sistema en la dirección deseada. El objetivo principal es establecer una estructura suficiente para que se mantenga la trayectoria del sistema hacia los fines deseados.
- 3. Centrada en políticas:** La gobernanza se centra en la formulación de políticas. Estas políticas tienen como objetivo inculcar las reglas, normas, principios y comportamientos que sustentan un rendimiento uniforme. El objetivo principal es implementar políticas que apoyen la dirección y el control esenciales para lograr y mantener el rendimiento del sistema.

2.2. El papel único de la gobernanza en el diseño e implementación de Sistemas de Sistemas

En base a este análisis de la gobernanza, se pueden identificar varios elementos importantes para el diseño e implementación de un SdS. Para un SdS, la gobernanza debe garantizar el éxito continuo de una tríada (Figura 2):

- 1) Dirección:** mantenimiento de una identidad coherente (la esencia fundamental de un SdS) capaz de sustentar la coherencia en las decisiones, acciones e interpretaciones, en apoyo de la trayectoria estratégica a largo plazo del SdS.
- 2) Supervisión:** proporcionar los controles, la regulación y la supervisión del rendimiento necesarios para integrar el SdS en el cumplimiento de la misión/propósito.
- 3) Responsabilidad:** garantizar que se establezcan responsabilidades y que los recursos se utilicen de forma eficiente para apoyar la consecución de los objetivos del SdS.

Con estos tres pilares de la gobernanza puede conseguirse el cumplimiento de los imperativos de *Proceso-Estructura-Política* para el diseño e implementación de la gobernanza del SdS.

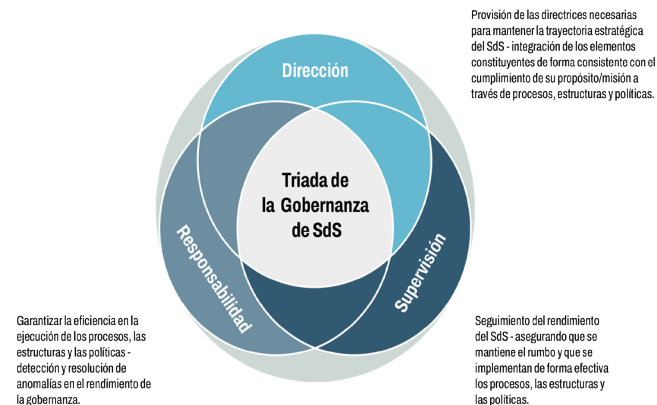


Figura 2. La tríada de gobernanza de un Sistema de Sistemas.

3. GOBERNANZA DE SISTEMAS DE SISTEMAS

Los SdS actuales están cada vez más interconectados y son más complejos, pero ofrecen posibilidades que superan con creces cualquier capacidad anterior. Estas capacidades mejoradas trascienden a los sistemas constituyentes del SdS y eran inimaginables hace una década. Es evidente que existen dificultades para gobernar los SdS, ya que parece imposible seguir el ritmo de aceleración de la información, de la interconexión y de la tecnología que caracterizan a la situación actual. A pesar de todas las “bondades” y capacidades que aportan los SdS, también han generado problemas que parecen irresolubles con los métodos y marcos actuales [1-3]. Es el momento oportuno para diseñar e implementar nuevos marcos que mejoren la capacidad para gobernar con mayor eficacia la evolución presente y futura de los SdS.

La GSC es una evolución emergente de los SdS que pretende mejorar las capacidades de su diseño, ejecución y evolución. Este campo emergente ofrece un nuevo e innovador camino para gobernar unos SdS cada vez más complejos, así como sus problemas. Desde su creación en 2014, la GSC ha madurado significativamente, evolucionando desde sus inicios hasta convertirse en un campo rico en conocimiento, aportando tanto puntos de vista teóricos como aplicaciones prácticas para los SdS en servicio [4].

La GSC evolucionó a partir del trabajo en Ingeniería de Sistemas de Sistemas (ISdS o SoSE, por sus siglas en inglés, *Systems of Systems Engineering*), tal como se describe en [5, 6]. El desarrollo de la GSC se debió en parte a las limitaciones de la SoSE para considerar tanto los aspectos técnicos como los no técnicos de los sistemas complejos, superando el estricto dominio de los enfoques y soluciones basados en “primero la tecnología, sólo la tecnología”. Por tanto, al incorporar la esencia de la teoría de sistemas, la GSC se orientó para abordar todo el espectro de los SdS, abarcando toda la gama de dimensiones sociotécnicas, económicas y políticas. Además, la incorporación de la gobernanza llevó a la GSC a un mayor énfasis, centrado en la dirección, la supervisión y la definición de responsabilidades en toda la gama de *drivers* sociotécnicos, económicos y políticos de los SdS. El campo de la GSC es muy joven, y si bien aún queda mucho por conseguir, a medida que la GSC continúa desarrollándose, se afronta un momento adecuado para reconocer y ampliar las contribuciones de la GSC a la gobernanza de los SdS.

En este apartado, se proporciona un marco para el diseño e implementación de la gobernanza de un SdS (GSC). Se comienza explicando los dos campos — Cibernética de la Gestión y Teoría de Sistemas— que, junto con la Gobernanza (de Sistemas), proporcionan las bases conceptuales para el establecimiento de la GSC. A continuación, se desarrolla el paradigma de la GSC para establecer sus aspectos fundamentales para la gobernanza de un SdS. Tras la introducción del paradigma, se aporta el marco detallado para la GSC. Este marco, basado en los campos y el paradigma subyacentes, proporciona la esencia de la GSC mediante un conjunto de funciones y canales de comunicación asociados. En conjunto, estos facilitan el diseño e implementación de la gobernanza de un SdS.

3.1. La gobernanza de Sistemas de Sistemas en la intersección de tres campos

Tres campos sirven como base conceptual para la GSC [3, 30]. En términos generales, estos campos comprenden:

- 1) La Teoría General de Sistemas**, que proporciona el conjunto de proposiciones (leyes, principios, conceptos) que sirven para definir el comportamiento, la estructura y el rendimiento de los sistemas complejos.
- 2) La Cibernética de la Gestión**, que se define como la ciencia de la organización eficaz de la estructura de los sistemas, orientada a asegurar su viabilidad (existencia continua).

- 3) La Gobernanza de Sistemas**, que se centra en establecer la dirección, la supervisión y la definición de responsabilidades en los sistemas complejos. La gobernanza de sistemas se ha abordado anteriormente. A continuación, se abordan los dos campos restantes, que se analizarán brevemente por su contribución única a la GSC como base para el diseño e implementación de la gobernanza de SdS.

3.1.1. Contribuciones de la teoría de sistemas

La contribución de la teoría de sistemas a la gobernanza de los SdS es triple. En primer lugar, proporciona una base teórica sólida y rigurosa. En segundo lugar, se ha articulado como un conjunto de axiomas (suposiciones que se dan por sentadas) y propuestas asociadas (principios, leyes y conceptos) que pretenden describir, explicar y predecir el comportamiento, la estructura o el rendimiento de los sistemas, ya sean naturales o artificiales [7-9]. Si bien un desarrollo detallado de la teoría de sistemas queda fuera del alcance de este capítulo, basta con decir que los axiomas y las propuestas proporcionan la base intelectual para la gobernanza de los SdS. En tercer lugar, la teoría de sistemas proporciona un lenguaje para comprender las deficiencias sistémicas que impiden el desempeño de las funciones de gobernanza de los SdS. Estas deficiencias, conocidas como patologías, se observan como violaciones de las proposiciones subyacentes de la teoría de sistemas.

En efecto, la teoría de sistemas proporciona una base teórica para el diseño y la implementación de la gobernanza de los SdS, de modo que se pueda mantener la integración y la coordinación necesarias para garantizar la viabilidad de los mismos.

3.1.2. Contribuciones de la Cibernética de Gestión

La Cibernética de Gestión se define en términos generales como “la ciencia de la organización eficaz de la estructura del sistema” [10-12]. El concepto de “metasistema” es fundamental para la Cibernética de Gestión, considerándolo un conjunto de funciones interrelacionadas que debe realizar todo sistema viable (que siga existiendo). El metasistema proporciona integración (permitiendo que un sistema actúe como una unidad) y coordinación (facilitando la interacción fluida entre sus componentes). Por tanto, un SdS se estructura de forma que pueda alcanzar los niveles de rendimiento necesarios para seguir siendo viable (existiendo).

La Cibernética de Gestión aporta tres importantes contribuciones al diseño e implementación de la gobernanza del SdS. En primer lugar, sus fundamentos ofrecen una sólida base teórica y conceptual. En su nivel más básico, se ocupa de la comunicación y el control. Esto se alinea con la gobernanza para proporcionar dirección y supervisión a medida que un

sistema continúa en una trayectoria deseable. En cuanto al control, el punto de vista cibernético sugiere que las restricciones basadas en el control impuestas a un sistema proporcionan una capacidad regulatoria esencial para asegurar el rendimiento del sistema y su viabilidad continua. En segundo lugar, la Cibernética de Gestión sienta las bases para la estructura de gobernanza, que incluye funciones y canales de comunicación consistentes con el éxito de la gobernanza para un sistema [10, 12, 13]. La Cibernética de Gestión es un punto de partida para el Modelo de Referencia GSC [3]. En tercer lugar, la Cibernética de Gestión se ha aplicado con éxito durante más de cinco décadas. A pesar de los avances tecnológicos y del rápido ritmo de cambio en los SdS, la Cibernética de Gestión ha mantenido su relevancia y una presencia sólida y sostenida.

3.2. Gobernanza de Sistemas de Sistemas – Un paradigma de última generación

La GSC es el «diseño, ejecución y evolución de las funciones del metasisistema necesarias para proporcionar control, comunicación, coordinación e integración de un sistema complejo» [4, p. 264]. Dentro de esta definición, se encuentra el paradigma subyacente que la materializa [2, 14], el cual se muestra en la Figura 3.

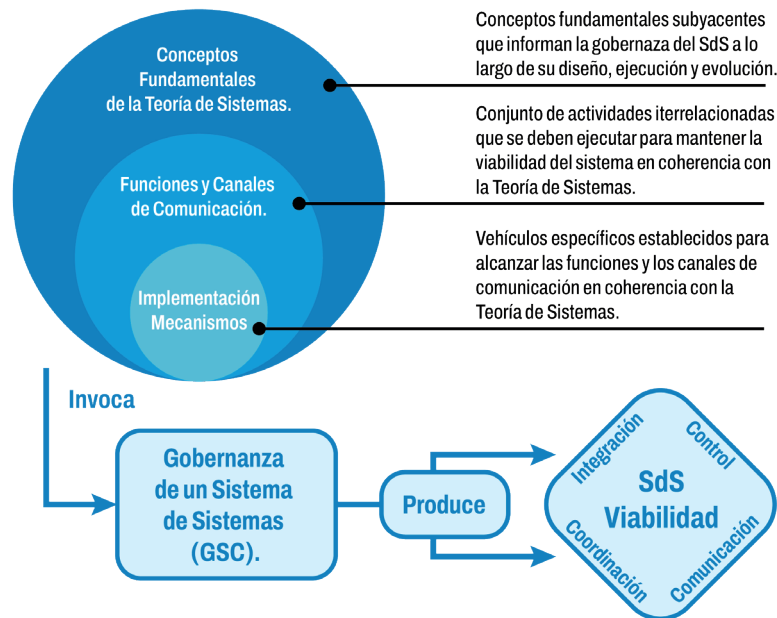


Figura 3. Relaciones del paradigma de gobernanza de un Sistema de Sistemas.

En primer lugar, el *diseño* acentúa la necesidad de perseguir deliberadamente la creación de la estructura de gobernanza. Si bien el diseño para la gobernanza de un SdS representa el caso normativo, la *ejecución* lo modera en base a la implementación en el entorno operativo. Cuando el diseño se une a la ejecución, el resultado es inevitablemente un diseño que requiere *evolución (desarrollo)* para realizar las modificaciones necesarias y adaptarse a la incertidumbre, los comportamientos emergentes y las deficiencias de diseño en un contexto de aplicación determinado.

En segundo lugar, los cuatro aspectos de la GSC incluyen el *control* (las restricciones regulatorias que garantizan el rendimiento y la trayectoria del SdS), la *comunicación* (el flujo, el procesamiento y la interpretación de la información a través de canales), la *coordinación* (centrada en la interacción entre las entidades constituyentes del sistema y con las externas al sistema, para evitar fluctuaciones innecesarias) y la *integración* (mantenimiento de la unidad del sistema mediante objetivos comunes, las responsabilidades y el equilibrio entre la autonomía individual de los constituyentes y los intereses a nivel de sistema).

En tercer lugar, la viabilidad (existencia continua) se garantiza mediante el desempeño de las *funciones* (imperativos del sistema que deben cumplirse para mantener la viabilidad) y los *canales de comunicación* asociados (los conductos que facilitan el flujo de información y la interpretación dentro y fuera del sistema). Las funciones y los canales de comunicación conforman el “*metasisistema*” que rige la gobernanza del SdS (Figura 4).

En cuarto lugar, las funciones y los canales de comunicación son ejecutados por *mecanismos* (vehículos que sirven para la implementación) exclusivos de cada sistema gobernado.

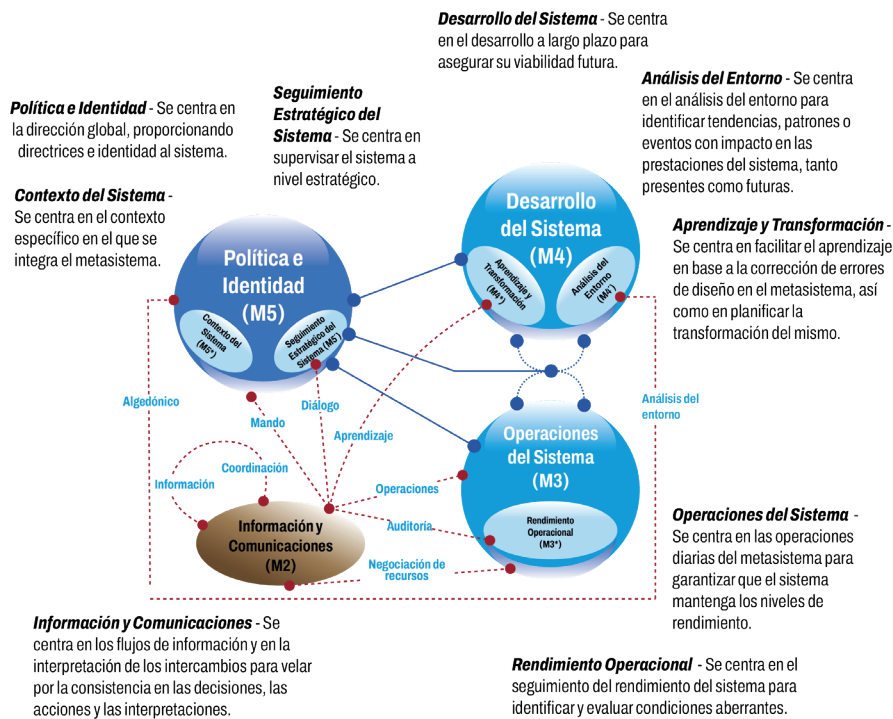


Figura 4. Funciones del metasisistema y canales de comunicación asociados.

El metasisistema es fundamental para la gobernanza de un SdS. Este metasisistema es el conjunto de funciones y canales de comunicación que van más allá de los sistemas individuales y que permiten la gobernanza de un SdS. Se presentan nueve funciones interrelacionadas y diez canales de comunicación que capturan la esencia de la gobernanza de un SdS [3, 4, 15]. Estas funciones son una extensión de la Cibernética de Gestión [10-12]. Las funciones del metasisistema y sus correspondientes canales de comunicación se muestran en la Figura 4. Como se recoge en dicha figura, existen cuatro funciones principales y cinco funciones subordinadas relacionadas, junto con diez canales de comunicación [2-4, 16].

La GSC es el conjunto de las nueve funciones interrelacionadas que proporcionan gobernanza a un sistema complejo. Estas funciones incluyen:

- **Metasisistema Cinco Estrella (M5*) – Contexto del Sistema:** responsable de elaborar y gestionar el contexto específico (factores que facilitan y limitan el rendimiento del SdS, p. ej., infraestructura de soporte, cultura, partes interesadas) en el que se integra el metasisistema. Supervisa y facilita la comunicación de los factores contextuales, tanto internos como externos al SdS.
- **Metasisistema Cinco Prima (M5') – Seguimiento Estratégico del Sistema:** proporciona supervisión del rendimiento del sistema a un nivel estratégico y determina el grado en el cual el SdS es efectivo en la búsqueda de directrices a largo plazo y el mantenimiento de la trayectoria futura.
- **Metasisistema Cuatro (M4) – Desarrollo del Sistema:** Se centra en la comprensión y las implicaciones para perseguir y lograr el desarrollo a largo plazo del SdS y garantizar su viabilidad futura. Analiza el entorno de los procesos para determinar los impactos en las operaciones actuales y el desarrollo futuro.
- **Metasisistema Cuatro Estrella (M4*) – Aprendizaje y Transformación:** se concentra en facilitar el aprendizaje basado en la corrección de errores de diseño en el metasisistema y la planificación para la transformación responsiva del SdS.
- **Metasisistema Cinco (M5) – Política e Identidad:** proporciona una dirección general (por ejemplo, visión, propósito, misión) para el SdS, dando dirección para asegurar que se mantenga la trayectoria del mismo, provee el mantenimiento de la identidad (la esencia de la exclusividad del SdS) responsable de generar consistencia en la decisión, acción e interpretación, representa al SdS en el “exterior”, mantiene las condiciones límite y equilibra el enfoque entre los intereses del SdS a corto y largo plazo.

- **Metasistema Cuatro Prima (M4') – Análisis del Entorno:** diseña, implementa y monitoriza la detección del entorno para identificar tendencias, patrones, condiciones, circunstancias o eventos emergentes con implicaciones para la viabilidad presente y futura del sistema. Mantiene un modelo activo del entorno del SdS.
- **Metasistema Tres (M3) – Operaciones del Sistema:** se centra en las operaciones diarias del metasistema para garantizar que el SdS mantenga los niveles de rendimiento deseados, consistentes con las distribuciones de recursos para producir valor por parte del SdS.
- **Metasistema Tres Estrella (M3*) – Rendimiento Operacional:** se ocupa del desarrollo y seguimiento de medidas de rendimiento operativo del sistema para monitorizar la productividad e identificar y evaluar condiciones aberrantes.
- **Metasistema Dos (M2) – Información y Comunicaciones:** se centra en el diseño del flujo y la interpretación de la información dentro del metasistema del SdS y desde este hacia los sistemas que lo componen. Facilita la interpretación consistente de los intercambios a través de los canales de comunicación para promover la coherencia en las decisiones, acciones e interpretaciones, tanto dentro como fuera del SdS.

Una forma en que las nueve funciones interrelacionadas de la GSC facilitan la gobernanza del SdS es a través de los *canales de comunicación del metasistema*. Estos canales facilitan el flujo de información para la toma de decisiones y la acción, además de generar coherencia en la interpretación de los intercambios dentro del metasistema y entre este y las entidades externas. La Tabla 1 enumera los canales de comunicación, su función principal asociada en el metasistema y su rol específico en la gobernanza del SdS [2, 10-12, 17].

Canal de comunicaciones y función metasistema asociadas	Función del canal para la gobernanza del SdS
Mando (Metasistema 5)	<ul style="list-style-type: none"> • Proporciona una orientación no negociable al metasistema y a los sistemas gobernados. • Fluye principalmente del Metasistema 5 y se difunde por todo el sistema.
Negociación de recursos/ Responsabilidad (Metasistema 3)	<ul style="list-style-type: none"> • Determina y asigna los recursos (mano de obra, material, dinero, métodos, tiempo, información, apoyo) a los sistemas gobernados. • Define los niveles de rendimiento (productividad), las responsabilidades y la rendición de cuentas para los sistemas gobernados. • Principalmente una interfaz entre el Metasistema 3 y los sistemas gobernados.
Operaciones (Metasistema 3)	<ul style="list-style-type: none"> • Proporciona la interfaz de rutina pertinente con un enfoque operacional a corto plazo. • Concentrado en dirigir al sistema en la generación de valor (productos, servicios, procesos, información) que se consume fuera del sistema. • Principalmente una interfaz entre el Metasistema 3 y los sistemas gobernados.
Coordinación (Metasistema 2)	<ul style="list-style-type: none"> • Garantiza el equilibrio y la estabilidad del metasistema y los sistemas gobernados. • Garantiza el diseño y la consecución (mediante la ejecución) del diseño: (1) el intercambio de información, dentro del sistema, necesaria para coordinar actividades, y (2) garantiza que las decisiones y acciones necesarias para prevenir perturbaciones se compartan dentro del metasistema y los sistemas gobernados. • Principalmente un canal diseñado y ejecutado por el Metasistema 2.
Auditoría (Metasistema 3*)	<ul style="list-style-type: none"> • Proporciona información rutinaria y esporádica sobre el rendimiento operativo. • Investigación e informes sobre problemas de rendimiento dentro del sistema. • Principalmente es un canal de Metasistema 3* para la comunicación entre Metasistema 3, los sistemas gobernados y el metasistema en relación con problemas de rendimiento.

Canal de comunicaciones y función metasisistema asociadas	Función del canal para la gobernanza del SdS
Algedónico (Metasisistema 5)	<ul style="list-style-type: none"> • Proporciona una derivación de todos los canales cuando la integridad del sistema se ve amenazada. • Genera alertas instantáneas ante crisis o situaciones potencialmente catastróficas para el sistema. • Se dirige al Metasisistema 5 desde cualquier parte del metasisistema o de los sistemas gobernados.
Análisis del entorno (Metasisistema 4')	<ul style="list-style-type: none"> • Proporciona el diseño para la detección y el seguimiento de aspectos críticos del entorno externo. • Identifica patrones, actividades o eventos del entorno con implicaciones para el sistema. • Proporciona acceso a través del metasisistema, así como a los sistemas gobernados por Metasisistema 4.
Diálogo (Metasisistema 5')	<ul style="list-style-type: none"> • Permite examinar las decisiones, acciones e interpretaciones del sistema para asegurar su coherencia con el propósito y la identidad del sistema. • Acceso al Metasisistema 5 desde cualquier parte del metasisistema o de los sistemas gobernados.
Aprendizaje (Metasisistema 4')	<ul style="list-style-type: none"> • Proporciona detección y corrección de errores dentro del metasisistema, así como en los sistemas gobernados, centrándose en problemas de diseño del sistema en lugar de problemas de ejecución. • Acceso a Metasisistema 4' desde cualquier parte del metasisistema o de los sistemas gobernados.
Información (Metasisistema 2)	<ul style="list-style-type: none"> • Facilita el flujo y el acceso a la información rutinaria dentro del metasisistema o entre este y los sistemas gobernados. • Acceso a todo el metasisistema y los sistemas gobernados. • Diseñado principalmente por Metasisistema 2 para su uso por todas las funciones del metasisistema, así como por los sistemas gobernados.

Tabla 2. Canales de comunicación para apoyar la gobernanza de un Sistema de Sistemas.

A primera vista, el marco de gobernanza de los SdS puede parecer algo detallado y quizás abrumador. Sin embargo, las funciones y los canales de comunicación ya se llevan a cabo, en cierta medida, en cada SdS viable (que sigue existiendo). Desafortunadamente, es probable que se realicen de forma fragmentada (*ad hoc*), sin las ventajas de un diseño, una supervisión y una definición de responsabilidades bien definidos e integrados. En definitiva, un SdS puede gobernarse sin un reconocimiento explícito de las funciones y los canales de comunicación. No obstante, se sugiere que, para que el rendimiento de la gobernanza de los SdS alcance niveles superiores, las funciones y los canales de comunicación deben ofrecer un marco explícito para un riguroso estudio y desarrollo.

4. APLICACIÓN DE LA GOBERNANZA PARA EL DISEÑO E IMPLEMENTACIÓN DE SISTEMAS DE SISTEMAS

Dado el marco de gobernanza del SdS proporcionado, la pregunta ahora es: ¿Cómo puede este marco ser implementado para apoyar el desarrollo de las capacidades de la gobernanza de un SdS?. Para responder a esta pregunta, el presente apartado se centra en la identificación de las deficiencias de la gobernanza de un SdS. Las deficiencias de gobernanza se identifican como “patologías”, que son alteraciones en el rendimiento normal o saludable de las funciones de gobernanza. Este apartado examina la implementación de la gobernanza de un SdS en base a tres temas centrales. Primero, se explora la identificación de patologías en las funciones de gobernanza del metasisistema. Estas patologías indican áreas donde las funciones de gobernanza no cumplen con las expectativas de rendimiento deseables. En segundo lugar, se exploran varios escenarios donde la gobernanza del SdS podría ofrecer utilidad para mejorar el rendimiento actual y futuro del SdS. En tercer lugar, se desarrolla un conjunto de guías para profesionales para la implementación de la gobernanza de los SdS. Esta guía ofrece “lecciones” de la aplicación de la GSC para mejorar la gobernanza de un SdS.

4.1. Identificación de patologías y problemas en la gobernanza de Sistemas de Sistemas

Básicamente, una patología se asocia generalmente con la salud, donde indica una desviación de lo que se espera como funcionamiento normal o saludable del sistema (p. ej., el cuerpo humano). Con respecto a la gobernanza de un SdS, una patología es “una circunstancia, condición, factor o patrón que actúa para limitar el rendimiento del sistema o disminuir su viabilidad, de modo que se reduce la probabilidad de que un sistema alcance las expectativas de sus prestaciones”. [18, p. 253]. En esencia, una patología es la degradación de una función del sistema, lo que afecta su capacidad para producir las prestaciones deseadas. La patología se observa como síntoma de una condición subyacente. Por tanto, una patología no es necesariamente algo obvio. En cambio, requiere una exploración a un nivel sistémico más profundo, más allá de su manifestación sintomática superficial. En la gobernanza de un SdS, una patología indica “violaciones” de las proposiciones de la teoría de sistemas (leyes, principios y conceptos).

Las funciones de la GSC y los canales de comunicación que regulan la gobernanza de un SdS ofrecen un conjunto de “coordenadas” para localizar la existencia de una patología. Esta ubicación está vinculada a nueve funciones metasistémicas diferentes, esenciales para la viabilidad continua de un SdS, que se articulan en un conjunto de 53 patologías específicas relacionadas con las funciones metasistémicas, que se organizan en torno ellas y sirven para identificar anomalías en el funcionamiento normal (saludable) de dichas funciones (Tabla 3) [18, 19].

Funciones del metasistema	Conjunto correspondiente de patologías
Metasistema cinco (M5): Política e identidad	M5.1. La identidad del sistema es ambigua y no genera consistencia efectiva en las decisiones, acciones e interpretaciones del sistema.
	M5.2. La visión, el propósito, la misión o los valores del sistema permanecen sin articular, o bien, articulados, pero no integrados en la ejecución del sistema.
	M5.3. El equilibrio entre el enfoque operativo a corto plazo y el enfoque estratégico a largo plazo no se ha explorado.
	M5.4. El enfoque estratégico carece de claridad suficiente para orientar el desarrollo coherente del sistema.
	M5.5. La identidad del sistema no se evalúa, mantiene ni cuestiona de forma rutinaria para garantizar su capacidad continua de guiar la coherencia en las decisiones y acciones del sistema.
	M5.6. La proyección del sistema externo no se realiza de manera efectiva.
Metasistema Cinco Estrella (M5*): Contexto del sistema	M5*.1. Contexto del metasistema incompatible que limita el rendimiento del sistema.
	M5*.2. Falta de articulación y representación del contexto del metasistema.
	M5*.3. Falta de consideración del contexto en las decisiones y acciones del metasistema.
Metasistema Cinco Prima (M5'): Monitorización estratégica del sistema	M5'.1. Falta de seguimiento estratégico del sistema.
	M5'.2. Procesamiento inadecuado de los resultados del seguimiento estratégico.
	M5'.3. Falta de indicadores de desempeño del sistema estratégico.
Metasistema Cuatro (M4): Desarrollo del sistema	M4.1. Falta de foros para impulsar el desarrollo y la transformación del sistema.
	M4.2. Interpretación y procesamiento inadecuados de los resultados del análisis del entorno: inexistentes, esporádicos y limitados.
	M4.3. Procesamiento y difusión ineficaces de los resultados del análisis del entorno.
	M4.4. Se sacrifica el desarrollo estratégico a largo plazo en aras de la gestión de las operaciones diarias: se dedica poco tiempo al análisis estratégico.
	M4.5. La planificación/reflexión estratégica se centra en la planificación y la mejora a nivel operativo.

Funciones del metasistema	Conjunto correspondiente de patologías
Metasistema Cuatro Estrella (M4*): Aprendizaje y transformación	M4*.1. Aprendizaje limitado en relación con los cambios del entorno.
	M4*.2. Transformación estratégica integrada no implementada, limitada o ineficaz.
	M4*.3. Falta de diseño para el aprendizaje sistémico: informal, inexistente o ineficaz.
	M4*.4. Ausencia de modelos representativos del sistema, presentes y futuros.
Metasistema Cuatro Prima (M4'): Análisis del entorno	M4'.1. Falta de mecanismos de monitorización efectivos.
	M4'.2. Objetivo inadecuado/monitorización del entorno no dirigido.
	M4'.3. Frecuencia de monitorización inadecuada para la velocidad de los cambios del entorno.
	M4'.4. El sistema carece de suficiente control sobre la variedad generada por el entorno.
	M4'.5. Falta de un modelo actual del entorno del sistema.
Metasistema Tres (M3): Operaciones del sistema	M3.1. Desequilibrio entre la autonomía de los elementos productivos y la integración de todo el sistema.
	M3.2. Cambios en los recursos sin los correspondientes cambios en la rendición de cuentas/cambios sin los correspondientes cambios en los recursos.
	M3.3. Desajuste entre las expectativas de recursos y productividad.
	M3.4. Falta de claridad en cuanto a responsabilidades, expectativas y rendición de cuentas por el rendimiento.
	M3.5. Planificación operativa frecuentemente preterida por crisis emergentes.
	M3.6. Equilibrio inadecuado entre el enfoque operativo a corto plazo y el enfoque estratégico a largo plazo.
	M3.7. Falta de claridad en la dirección operativa de las entidades productivas (es decir, sistemas constituyentes).
	M3.8. Dificultad para gestionar la integración de las entidades productivas (es decir, sistemas constituyentes).
	M3.9. Lentitud para anticipar, identificar y responder a los cambios del entorno.
Metasistema Tres Estrella (M3*): Rendimiento operativo	M3*.1. Accesibilidad limitada a los datos necesarios para supervisar el rendimiento.
	M3*.2. Los indicadores de rendimiento operativo a nivel de sistema son inexistentes, limitados o ineficaces.
	M3*.3. Ausencia de supervisión del rendimiento a nivel de sistema y subsistema.
	M3*.4. Falta de análisis de la variabilidad del rendimiento o desviaciones emergentes respecto a los niveles de rendimiento esperados: significado de las desviaciones.
	M3*.5. La auditoría de rendimiento es inexistente, limitada o se limita principalmente a la resolución de problemas emergentes.
	M3*.6. La evaluación periódica del rendimiento del sistema es en gran medida desorganizada e informal.
	M3*.7. Aprendizaje limitado del sistema basado en evaluaciones de rendimiento.
Metasistema Dos (M2): Información y comunicaciones	M2.1. Problemas de coordinación no resueltos dentro del sistema.
	M2.2. El exceso de redundancias en el sistema genera inconsistencias y un uso ineficiente de los recursos, incluida la información.
	M2.3. Problemas de integración del sistema derivados del aislamiento o la fragmentación excesivos de las entidades.
	M2.4. Conflictos en el sistema derivados de decisiones y acciones unilaterales.
	M2.5. Exceso de crisis emergentes asociadas con la transmisión, comunicación y coordinación de información dentro del sistema.
	M2.6. Sistemas de comunicación deficientes o ineficaces entre las entidades del sistema (es decir, subsistemas).
	M2.7. Falta de métodos estandarizados (es decir, procedimientos, herramientas y técnicas) para las actividades rutinarias a nivel del sistema.
	M2.8. Uso excesivo de métodos estandarizados (es decir, procedimientos, herramientas y técnicas) cuando deberían personalizarse.
	M2.9. Coordinación del sistema excesivamente <i>ad hoc</i> frente a un diseño específico.
	M2.10. Dificultad para realizar funciones intersistémicas que requieren integración o estandarización.
	M2.11. Introducción de cambios descoordinados en el sistema que resultan en una oscilación excesiva.

Tabla 3. Patologías correspondientes a funciones del metasistema.

Aunque exceden el alcance de este capítulo, las patologías del metasisistema también están vinculadas a violaciones de las proposiciones subyacentes de la teoría de sistemas [20, 21]. La identificación de patologías para la gobernanza del SdS incluye la evaluación en tres dimensiones de existencia, impacto y viabilidad. La **existencia** se refiere al grado en que se determina que la patología está presente en un metasisistema del SdS. El **impacto** se refiere al grado de gravedad que la existencia de la patología sugiere sobre el rendimiento del metasisistema del SdS. La **viabilidad** aborda la probabilidad de que, dado el estado actual del metasisistema y su contexto, la patología pueda abordarse con una probabilidad razonable de ser resuelta con éxito. Cada patología puede evaluarse a lo largo de las tres dimensiones (Figura 5).

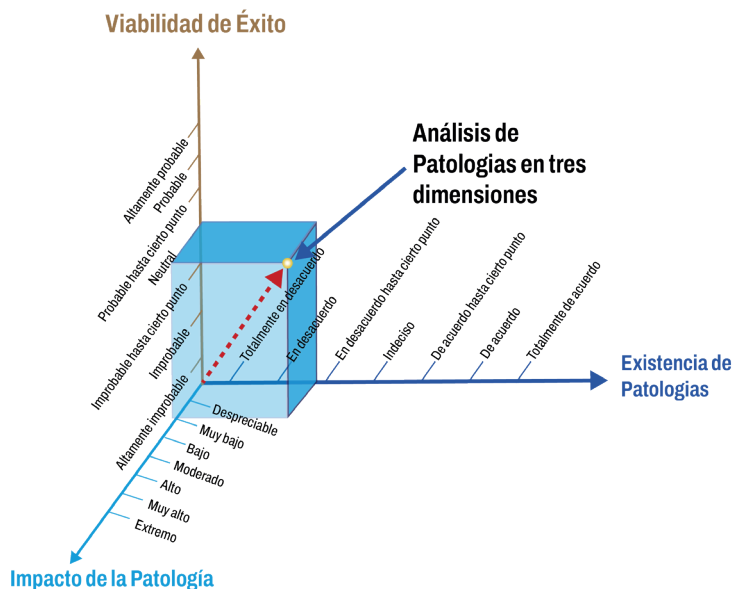


Figura 5. Evaluación tridimensional de patologías.

4.2. Escenarios para la implementación de la gobernanza de un SdS

En esta sección, se examinan tres escenarios donde la gobernanza de un SdS aporta perspectivas y se ofrecen alternativas para su desarrollo. El primer escenario examina la priorización de patologías para el desarrollo del SdS. Este escenario examina la posibilidad de dirigir, o redirigir, los recursos limitados según las necesidades prioritarias de desarrollo para abordar dichas patologías. El segundo escenario se basa en el desarrollo de capacidades sistémicas para la gobernanza de un SdS. El tercer escenario examina la claridad en la gobernanza del SdS.

4.2.1. Priorización de la inversión en recursos escasos para el desarrollo de un SdS

Todos los SdS cuentan con recursos que se invierten para su desarrollo y mejora. Un ejemplo de ello es la introducción de una nueva iniciativa (p. ej., Lean Six Sigma) como algo que se reconoce como una buena medida para mejorar un SdS. Sin embargo, no reconocer los problemas que generaría la iniciativa no significa que sea correcta ni que sea una inversión inteligente de unos recursos escasos. La gobernanza de un SdS sugiere que los recursos limitados deben destinarse a las áreas de gobernanza que presentan deficiencias (p. ej., patologías mayoritarias, con mayor impacto y de corrección viable).

No se deben malgastar los recursos limitados en actividades de desarrollo sin establecer su necesidad específica y su contribución prioritaria para impulsar el estado de la gobernanza del SdS. Otras dos contribuciones de la gobernanza del SdS podrían materializarse en este escenario. En primer lugar, existe la posibilidad de redirigir los recursos limitados, ya comprometidos, a patologías de mayor prioridad que requieren desarrollo. En segundo lugar, comprender el estado de la gobernanza del SdS brinda la oportunidad de ir más allá del desarrollo fragmentado para buscar un desarrollo de gobernanza del SdS más orquestado, holístico e integrado. En lugar de considerar las iniciativas de desarrollo de forma aislada, cada iniciativa de desarrollo de gobernanza del SdS puede priorizarse en función de sus necesidades. Se deben evitar aquellas iniciativas que se consideren ineficaces o inviables.

4.2.2. Maduración de las capacidades basadas en sistemas para la gobernanza de Sistemas de Sistemas

Todos los SdS tienen un nivel de madurez sistémica. Este nivel depende de la experiencia, actividades, diseño y ejecución del SdS. Si se deja que esta madurez se desarrolle por sí sola, no hay garantía de que alcance el estado deseado ni de que tenga la rapidez de desarrollo esperada. La participación en el diseño e implementación de la gobernanza del SdS contribuye a la maduración de las capacidades basadas en los sistemas. El desarrollo de la gobernanza puede seguir un ciclo de (1) descubrimiento de la existencia, impacto y viabilidad

de las patologías de la gobernanza para su resolución; (2) priorización y clasificación de las patologías que deben abordarse con recursos limitados; y (3) el compromiso deliberado con los objetivos de desarrollo de alta prioridad. El impacto de este ciclo es la maduración continua de las capacidades basadas en sistemas para la gobernanza del SdS.

4.2.3. Claridad y transparencia en la gobernanza de un Sistema de Sistemas

El desarrollo de la gobernanza del SdS puede mejorar el rendimiento de diversas formas. En primer lugar, la gobernanza del SdS hace que éste y su contexto sean claros, explícitos y transparentes. El desarrollo de la gobernanza del SdS exige la construcción de una representación que implica hacer explícito el diseño de la misma. Si se deja en un nivel tácito, la gobernanza del SdS puede resultar ambigua y potencialmente generar confusión en la ejecución de las funciones de gobernanza y los canales de comunicación. Por el contrario, al hacer explícito el diseño de la gobernanza del SdS, la transparencia puede generar claridad tanto en el diseño como en la implementación. En segundo lugar, a partir de un estado de referencia de la gobernanza del SdS, se puede evaluar la trayectoria de la gobernanza y las contribuciones de iniciativas de desarrollo específicas. Esto facilita la continuación, modificación o eliminación de iniciativas de desarrollo de la gobernanza basadas en la evaluación del rendimiento. En tercer lugar, una gobernanza del SdS clara y transparente proporciona coherencia en la trayectoria actual y futura del mismo. Esto facilita la capacidad de cuestionar decisiones, acciones e interpretaciones que mantienen un *statu quo* sin suficiente explicación, lógica o justificación.

La claridad y la transparencia son características distintivas de una gobernanza eficaz del SdS. Cuanto mejor se comprenda la gobernanza, mayor será la probabilidad de que las personas comprendan sus roles, responsabilidades, contribuciones y rendición de cuentas en las funciones de gobernanza.

5. PRÁCTICAS DE REFERENCIA

Esta sección pretende resumir el diseño y la implementación de la gobernanza del SdS. En primer lugar, se resume la gobernanza del mismo en una presentación concisa de sus puntos clave. Se ofrecen cinco puntos clave para definir la gobernanza del SdS desde la perspectiva de la GSC. En segundo lugar, se exploran una serie de desafíos y precauciones para la implementación de la gobernanza del SdS. La esencia de estos desafíos se basa en nuestra experiencia en la implementación de la gobernanza de SdS.

5.1. Una explicación concisa de la gobernanza de Sistemas de Sistemas para profesionales

La gobernanza de los SdS ofrece a los profesionales un enfoque novedoso para lograr una gobernanza más sofisticada. La GSC se definió para la gobernanza del SdS como el “diseño, ejecución y evolución de las funciones del metasisistema necesarias para proporcionar control, comunicación, coordinación e integración de un sistema complejo” [4, p. 264]. La gobernanza del SdS puede resumirse en cinco elementos fundamentales [2, 15, 22, 23]:

1. Todos los sistemas están sujetos a las proposiciones (leyes, principios y conceptos) de los sistemas. Así como las leyes rigen la materia y la energía (p. ej., la ley de la gravedad), existen proposiciones que explican y predicen el comportamiento y el rendimiento de los sistemas. Estas proposiciones de los sistemas se derivan de la Teoría General de Sistemas y son innegociables, imparciales y ofrecen explicaciones sobre el rendimiento del sistema. Los profesionales deben preguntarse: ¿Se entienden las proposiciones fundamentales de los sistemas y cómo impactan en el diseño y el rendimiento de la gobernanza del SdS?
2. Todos los sistemas desempeñan funciones esenciales de gobernanza que determinan su rendimiento. Se han presentado nueve funciones de gobernanza y diez canales de comunicación. Estas funciones y canales de comunicación se ejecutan en todos los sistemas, independientemente de su sector, tamaño o propósito. Las funciones definen qué se debe lograr para la gobernanza de un sistema. Las funciones se invocan mediante un conjunto de mecanismos de implementación exclusivos de cada SdS (medios para lograr las funciones de gobernanza). Por ejemplo, una reunión semanal de personal y conferencias semestrales son ejemplos de estos mecanismos. Éstos determinan cómo se llevan a cabo las funciones de gobernanza y los canales de comunicación. Cada uno de ellos puede ser tácito/explicito, formal/informal, limitado/exhaustivo o rutinario/espórádico en su aplicación. Los profesionales deben preguntarse: ¿Se entienden los mecanismos que desempeñan las funciones esenciales de gobernanza y los canales de comunicación para lograr el rendimiento del SdS?
3. Las funciones de gobernanza pueden experimentar patologías (desviaciones de las condiciones “saludables” del sistema) en el desempeño de las funciones y los canales de comunicación. Ningún diseño de sistema de gobernanza de un SdS logra la perfección en la ejecución. Independientemente de la “excelente” calidad del diseño de un sistema, la ejecución depende

de demasiadas variabilidades para “asegurar” la completa realización de las intenciones de diseño. La eficacia de la gobernanza depende de la eficacia en la identificación, la evaluación, la respuesta y el análisis de patologías inevitables. La gobernanza apoya la consecución de resiliencia y robustez para resistir y perseverar en medio de las turbulencias externas y los cambios internos del sistema. Los sistemas bien gobernados abordan las patologías a medida que ocurren, mientras que los sistemas excelentes eliminan continuamente las patologías de gobernanza antes de que se conviertan en crisis. Los profesionales deben preguntarse: ¿Se diseñan y rediseñan continuamente los SdS para abordar y prevenir las patologías?

4. Las violaciones de las proposiciones sistémicas en la ejecución de las funciones de gobernanza conllevan consecuencias. Estas proposiciones no pueden ignorarse. Independientemente de las buenas intenciones, la ignorancia inconsciente o la indiferencia deliberada, violar las proposiciones sistémicas conlleva consecuencias reales para el desempeño del sistema. En el mejor de los casos, las violaciones de las proposiciones sistémicas sólo degradarán el desempeño. En el peor, se intensificarán hasta causar consecuencias catastróficas o, peor aún, el colapso total del sistema. Los profesionales deben preguntarse: ¿Se entiende el desempeño deficiente del SdS como violaciones de las proposiciones sistémicas subyacentes que afectan a sus funciones?
5. El rendimiento del sistema puede mejorarse mediante el desarrollo intencionado de las funciones de gobernanza. Cuando los SdS no cumplen las expectativas de rendimiento, la evaluación de las deficiencias (patologías) que contribuyen a las funciones de gobernanza ofrece nuevas perspectivas sobre las causas profundas del fallo. Los problemas de rendimiento de los SdS pueden atribuirse a problemas en las funciones de gobernanza y, eventualmente, a infracciones de las proposiciones subyacentes del sistema. Mediante un desarrollo con objetivos, la gobernanza puede proceder desde una posición más informada. Los profesionales deben preguntarse: ¿Cómo podrían explorarse nuestras funciones de gobierno y canales de comunicación para determinar las violaciones de las proposiciones del sistema?

Si bien no es un conjunto completo de temas para la gobernanza de los SdS, sí captura la esencia del nuevo campo la GSC para guiar el diseño y la implementación de la gobernanza de los SdS.

5.2. Desafíos y precauciones para la implementación de la gobernanza de un Sistema de Sistemas

El diseño y la implementación de la gobernanza de un SdS no es una tarea sencilla, ni rutinaria, ni que requiera pocos recursos. Dada la complejidad de un proyecto de desarrollo de la gobernanza de un SdS, existen varios desafíos que los profesionales deben considerar. Nuestra experiencia en la aplicación de la GSC ha proporcionado las siguientes perspectivas para quienes contemplan este tipo de proyecto para el desarrollo de la gobernanza de un SdS [2]:

1. El desarrollo de la gobernanza de un SdS no es el punto de partida: Por muy prometedora que sea la gobernanza de un SdS para mejorar la comprensión y el rendimiento del sistema, no debe ser la máxima prioridad para quienes consideran su aplicación. En cambio, la prioridad para los profesionales se centra en los problemas del SdS. Por tanto, centrarse en el papel de la gobernanza del SdS para abordar los problemas relevantes del mismo probablemente generará una mayor implicación.
2. La participación en la gobernanza de un SdS no es una tarea de todo o nada: A partir de los resultados de las exploraciones iniciales sobre la gobernanza de un SdS y sus implicaciones, se pueden explorar numerosas vías potenciales de desarrollo. Considerar la gobernanza de un SdS como una tarea de todo o nada es erróneo. En cambio, la vía de desarrollo podría abarcar un espectro de actividades (educación, formación, evaluación limitada, modelado, etc.) y niveles de desarrollo (profesional, sistema, proyecto, entidad, empresa, problema) para mejorar la gobernanza del SdS.
3. La gobernanza de un SdS no es una iniciativa complementaria: A diferencia de las actividades de desarrollo de sistemas más tradicionales que buscan abordar un nuevo reto mediante la introducción de una iniciativa totalmente nueva (p. ej., Lean, Six Sigma, TQM, CRM, adopción de IA, Internet de las Cosas, etc.), las funciones y los canales de comunicación de la gobernanza de un SdS ya se realizan en un SdS viable (existente). Puede que las funciones no estén desarrolladas ni produzcan el rendimiento deseado, pero, sin embargo, se están ejecutando en cierta medida. La gobernanza de un SdS se centra en comprender y, potencialmente, mejorar lo que ya realiza un SdS. Por tanto, el lenguaje, el pensamiento y las exploraciones de la gobernanza de un SdS se aplican a uno existente donde ya se realizan tácitamente.
4. El tiempo y el riesgo del desarrollo de la gobernanza de un SdS deben recaer inicialmente en un guía: No es realista esperar que los participantes de un SdS se involucren plenamente en una iniciativa de desarrollo de la gobernanza de un SdS en términos de inversión de tiempo y aceptación del riesgo de fracaso. En cambio, un facilitador de la

gobernanza del SdS, en lugar de implementar el SdS, debería asumir la carga inicial de la inversión de tiempo y la mitigación de riesgos. Esta división debe continuar hasta que el valor de la inversión (tiempo) y la utilidad del desarrollo de la gobernanza del SdS se combinen para producir un equilibrio riesgo-valor-coste que los profesionales consideren razonable. El desarrollo de la gobernanza de un SdS debe llevarse a cabo en un modo “seguro ante el fracaso”, donde existan posibilidades de asumir riesgos sin temor a represalias por no cumplir con los requisitos.

6. CONCLUSIONES

El diseño y la implementación de la gobernanza de un SdS no resultan triviales. Sin embargo, para aquellos profesionales y entidades dispuestos a involucrarse con audacia en el desarrollo de la gobernanza de un SdS, el potencial para mejorar su rendimiento es sustancial. Existen cuatro puntos principales de énfasis para esta exploración de la gobernanza y la implementación de un SdS. En primer lugar, la gobernanza del SdS proporciona dirección, supervisión y una definición de responsabilidades presentándose como una opción claramente diferente a la gestión tradicional. En segundo lugar, la gobernanza de los SdS se ha presentado a través de un conjunto de nueve funciones del metasisistema y diez canales de comunicación que generan la gobernanza y apoyan la viabilidad continua del sistema. En tercer lugar, las funciones de gobernanza del SdS están sujetas a experimentar patologías como desviaciones respecto a las condiciones normales, o robustas. Las patologías degradan el desempeño del SdS y pueden eventualmente conducir a un fallo catastrófico. Éstas pueden evaluarse por su existencia, su impacto y la viabilidad para ser resueltas. En cuarto lugar, las contribuciones de la gobernanza del SdS incluyen permitir una asignación eficiente de los recursos limitados para su desarrollo, madurar las capacidades basadas en sistemas y proporcionar transparencia en el diseño, ejecución y desarrollo de la gobernanza del SdS. Participar en el desarrollo de la gobernanza del SdS no es una tarea fácil. Los recursos, la voluntad y el compromiso necesarios para su desarrollo son elevados. Sin embargo, el potencial de mejora en el rendimiento de los SdS es considerable.

REFERENCIAS

1. Katina, P. F., Keating, C. B., Bobo, J. A., & Toland, T. S. (2019). A governance perspective for system-of-systems. *Systems*, 7(4), 54.
2. Keating, C. B. (2022). *Complex System Governance*. In *Complex System Governance: Theory and Practice* (pp. 151-186). Cham: Springer International Publishing.
3. Keating, C. B. (2014, June). Governance implications for meeting challenges in the system of systems engineering field. In 2014 9th International Conference on System of Systems Engineering (SOSE) (pp. 154-159). IEEE.
4. Keating, C. B., Katina, P. F., & Bradley, J. M. (2014). Complex system governance: concept, challenges, and emerging research. *International Journal of System of Systems Engineering*, 5(3), 263-288.
5. Gorod, A., Sauser, B., & Boardman, J. (2008). System-of-systems engineering management: A review of modern history and a path forward. *IEEE Systems Journal*, 2(4), 484-499.
6. Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., and Rabadi, G. (2003). System of systems engineering. *Engineering Management Journal*, 15(3), 36-45.
7. Adams, K. M., Hester, P. T., Bradley, J. M., Meyers, T. J., & Keating, C. B. (2014). Systems theory as the foundation for understanding systems. *Systems Engineering*, 17(1), 112-123.
8. Whitney, K., Bradley, J. M., Baugh, D. E., & Jr, C. W. C. (2015). Systems theory as a foundation for governance of complex systems. *International Journal of System of Systems Engineering*, 6(1-2), 15-32.
9. Castelle, K., Bradley, J. M., & Chesterman Jr, C. W. (2022). Systems Theory for Complex System Governance. In *Complex System Governance: Theory and Practice* (pp. 97-118). Cham: Springer International Publishing.
10. Beer, S. (1979). *The heart of Enterprise*. Wiley, Cheshire.
11. Beer, S. (1995). *Brain of the Firm*. John Wiley & Sons.
12. Beer, S. (1985). *Diagnosing the system for organizations*. John Wiley and Sons.
13. Warfield, J.N. (1976). *Societal systems: Planning, policy and complexity*. New York, NY: Wiley-Interscience.
14. Keating, C. B. (2022). *Complex System Governance*. In *Complex System Governance: Theory and Practice* (pp. 151-186). Cham: Springer International Publishing.
15. Keating, C. B., Katina, P. F., & Bradley, J. M. (2015). Challenges for developing complex system governance
16. Keating C.B. (2014). Governance implications for meeting challenges in the system of systems engineering field. In: 2014 9th international conference on system of systems engineering (SOSE), IEEE, Adelaide, pp. 154–159.
17. Keating, C. B., & Morin, M. (2001). An approach for systems analysis of patient care operations. *JONA: The Journal of Nursing Administration*, 31(7/8), 355-363.
18. Keating, C. B., & Katina, P. F. (2012). Prevalence of pathologies in systems of systems. *International Journal of System of Systems Engineering*, 3(3-4), 243-267.
19. Katina, P. (2022). "Metasystem Pathologies in Complex System Governance" in eds. Keating, C., Katina, P., Chesterman, C., and Pyne, J. *Complex System Governance*, Springer.
20. Keating, C. B. (2015, May). Complex system governance: Theory to practice challenges for system of systems engineering. In 2015 10th System of Systems Engineering Conference (SoSE) (pp. 226-231). IEEE.
21. Katina, P.F. (2016). Systems theory as a foundation for discovery of pathologies for complex system problem formulation. In A. J. Masys (Ed.), *Applications of Systems Thinking and Soft Operations Research in Managing Complexity* (pp. 227–267). Geneva, Switzerland: Springer International Publishing.
22. Keating, C. B., & Katina, P. F. (2019). Complex system governance: Concept, utility, and challenges. *Systems Research and Behavioral Science*, 36(5), 687-705.
23. Keating, C. B. (2022). *Complex System Governance*. In *Complex System Governance: Theory and Practice* (pp. 151-186). Cham: Springer International Publishing.

BIOGRAFÍAS

DR. POLINPAPILINHO F. KATINA

El Dr. Katina es Profesor titular del Departamento de Informática y Sistemas de Ingeniería de la Universidad de Carolina del Sur Upstate (Spartanburg, Carolina del Sur, EE.UU.). Ha desempeñado diversos cargos en el sector privado y en el público, incluyendo la empresa Design Electric (Charlottesville, Virginia, EE.UU.), el Politécnico de Milán (Milán, Italia), los Centros Nacionales de Ingeniería de Sistemas (Norfolk, Virginia, EE.UU.), la Universidad Old Dominion (Norfolk, Virginia, EE.UU.), la Universidad de Siracusa (Siracusa, Nueva York, EE.UU.) y la Universidad de Alabama en Huntsville (Huntsville, Alabama, EE.UU.).



El Dr. Katina es licenciado en Tecnología de Ingeniería, máster en Ingeniería de Sistemas y doctor en Gestión de Ingeniería e Ingeniería de Sistemas (Universidad Old Dominion, Norfolk, Virginia, EE.UU.). Realizó formación adicional en el Politécnico de Milán (Milán, Italia).

Sus intereses de docencia e investigación giran en torno a temas de gobernanza de sistemas complejos, infraestructuras críticas, tecnologías emergentes (p.e., IoT, redes inteligentes), gestión de ingeniería, Infranomics, sistemas de fabricación, sistemas de sistemas, ingeniería de sistemas, patología de sistemas, teoría de sistemas y pensamiento sistémico.

Su perfil incluye más de 200 artículos en revistas revisados por pares, actas de congresos y capítulos de libros. También es coautor de más de una docena de libros, incluyendo "Gamification for Resilience" (Wiley, 2023). Ha publicado en más de 21 revistas de investigación de gran impacto y visibilidad. Ha recibido varios premios, incluyendo el Premio al Libro de Ciencias Sociales de la IAA 2020 (IAA: Academia Internacional de Astronáutica).

El Dr. Katina es miembro senior del IEEE y de Epsilon Mu Eta (Sociedad de Honor en Gestión de Ingeniería), evaluador del programa ABET y editor de varias revistas, entre las que se incluyen Advanced Manufacturing (ELSP), Control and Engineering of Complex Systems Frontiers in Complex Systems (Frontiers Media SA), Cureus Journal of Engineering (Springer/Nature), Discrete Dynamics in Nature and Society (John Wiley), International Journal of Critical Infrastructures (Inderscience) e International Journal of System of Systems Engineering (Inderscience).

DR. CHARLES B. KEATING

El Dr. Keating es Profesor Emérito de Gestión de la Ingeniería e Ingeniería de Sistemas en la Universidad Old Dominion. Su trabajo de investigación se centra en la Gobernanza de Sistemas Complejos, la Ingeniería de Sistemas, la Intervención Sistémica y la Cibernética de Gestión. Es miembro y expresidente de la Sociedad



Americana de Gestión de la Ingeniería y ganador en 2015 del Premio Sarchet de dicha sociedad por sus iniciativas pioneras en este campo. Ha publicado más de 170 artículos revisados por pares y ha graduado a 32 doctores. Su investigación abarca las industrias de defensa, seguridad, aeroespacial, salud, I+D y automoción. Es licenciado en Ingeniería por la Academia Militar de los Estados Unidos (West Point), Máster en Gestión por la Universidad Central de Michigan y doctor en Gestión de la Ingeniería por la Universidad Old Dominion. Es miembro de la Sociedad Americana de Gestión para la Ingeniería, la Sociedad Internacional de Ciencias de Sistemas y el Consejo Internacional de Ingeniería de Sistemas.

CÉSAR HERAS MENOR DE GASPAR

Ingeniero de sistemas. Durante sus casi 20 años en Isdefe su trayectoria profesional se ha orientado a actividades de ingeniería dentro de la Dirección general de Armamento y Material de la Secretaría de Estado de Defensa, en los ámbitos de planeamiento de obtención de recursos de armamento y material, desarrollo de sistemas RPAS (Remotely Piloted Aircraft System) y la gestión de grandes programas de defensa.



Actualmente presta asistencia técnica en la Oficina de Programa Española del Sistema de Armas de Última Generación (NGWS, Next Generation Weapon System) dentro del entorno de un Futuro Sistema de Combate Aéreo (FCAS, Future Combat Air System), especialmente en los ámbitos del caza de sexta generación, sistemas remotos y sistemas de sistemas.

Antes de unirse a Isdefe, César trabajó como ingeniero de sistemas en HoneyWell, Amper Programas en los programas de sistemas de aviónica y comunicaciones.

En 2021 le fue concedida la Cruz del Mérito Aeronáutico con distintivo blanco por su trabajo apoyando a la Dirección General de Armamento y Material.

VÍCTOR RAMOS DEL POZO

Víctor Ramos del Pozo es ingeniero de sistemas en Ingeniería de Sistemas para la Defensa de España (Isdefe) que actualmente presta asistencia técnica en la Oficina de Programa Española del Sistema de Armas de Última Generación (NGWS, Next Generation Weapon System) dentro del entorno de un Futuro Sistema de Combate Aéreo



(FCAS, Future Combat Air System), especialmente en los ámbitos de la nube de combate, sensores colaborativos, simulación y sistemas de sistemas. Anteriormente prestó asistencia técnica en la Dirección General de Armamento y Material de la Secretaría de Estado de Defensa en los ámbitos del análisis de la base industrial de defensa, el planeamiento de la obtención de armamento y material, y en la gestión de grandes programas de defensa.

Antes de unirse a Isdefe, Víctor trabajó como ingeniero de sistemas en EADS-CASA (actual Airbus Defence and Space) en los programas de aviones de reabastecimiento en vuelo MRTT y FSTA, así como en Indra en el departamento de sistemas de identificación amigo-enemigo (IFF).

Es miembro del Consejo Internacional de Ingeniería de Sistemas (INCOSE) y es formador interno en Isdefe en los ámbitos de la ingeniería de sistemas y de la gestión de programas.

En 2015 le fue concedida la Cruz del Mérito Aeronáutico con distintivo blanco por su trabajo apoyando a la Dirección General de Armamento y Material.



Planificación e integración de Sistemas de Sistemas

Dr. Michael Sievers, *University of Southern California* (michael.sievers@usc.edu)
Pablo Marticorena San José, *Isdefe* (pmarticorena@isdefe.es)

Resumen

La heterogeneidad de los Sistemas de Sistemas y su organización dinámica crean oportunidades y complejidades que requieren un proceso cuidadoso de planificación e integración para satisfacer las necesidades de los primeros usuarios, sin restringir las necesidades futuras anticipadas o no identificadas. Este proceso requiere la negociación y el equilibrio que involucra considerar hasta dos docenas de factores, llevando a cabo hasta nueve actividades de planificación. Este capítulo examina los factores que afectan la planificación de SdS y los objetivos de cada actividad de planificación.

Palabras clave

Planificación de SdS, integración de SdS, ciclo de vida de SdS, integración de SdS.



1. INTRODUCCIÓN

Este capítulo examina los factores y métodos empleados en la planificación e integración de los aspectos técnicos de un Sistema de Sistemas (SdS). En particular, la vida útil de un SdS puede extenderse durante muchos años, comenzando con capacidades iniciales modestas que evolucionan continuamente para satisfacer las necesidades de los nuevos usuarios y usos. Un ejercicio de planificación bien concebido tiene como objetivo crear una base de SdS que se adapte a la conectividad y a los servicios a corto plazo para los primeros usuarios, sin restringir las necesidades futuras anticipadas o no identificadas.

Mientras que la planificación e integración de un SdS comparte muchas actividades similares a las utilizadas en el desarrollo de un sistema tradicional, un SdS tiene características únicas que contribuyen a su utilidad y complican su planificación e integración [1-5], como se discute en el Capítulo 1, que ocurre en múltiples actividades interdependientes.

En la siguiente sección se analizan las consideraciones de planificación. La Sección 3 examina la planificación de la interoperabilidad, la Sección 4 analiza la integración de SdS y, por último, la Sección 5 proporciona una conclusión.

2. CONSIDERACIONES DE PLANIFICACIÓN DE SISTEMAS DE SISTEMAS

2.1. Características de rendimiento y comportamiento de SdS

Antes de describir las actividades de planificación, es necesario discutir las características clave que impactan en la planificación. Estas características establecen los requisitos y las restricciones para la arquitectura, la funcionalidad, la seguridad (safety y security), las operaciones y la utilidad de SdS. La Tabla 1 enumera los factores clave de SdS y sus influencias de planificación.

Factor	Consideración	Función de la planificación
Accesibilidad	Los servicios y productos proporcionados por el SdS son fácilmente accesibles para los usuarios del SdS.	Determinar qué servicios y recursos necesitan los clientes, así como los medios y limitaciones para proporcionarlos.
Adaptabilidad	El SdS cambia interactivamente sus comportamientos y conexiones para adaptarse a los usuarios individuales.	Determinar qué adaptabilidad se necesita y cómo proporcionarla.
Asequibilidad	El grado en que el coste del tiempo de vida del SdS está dentro de las limitaciones presupuestarias previstas.	Establecer presupuestos para desarrollo, operaciones y mantenimiento.
Calidad del servicio	Medida de la eficacia y el rendimiento de un servicio o el SdS para realizar tareas y satisfacer las expectativas de los usuarios.	Evaluar la calidad de servicio esperada por los clientes y los planes de medición y realización de reparaciones o modificaciones para mantenerla.
Confianza	Confianza en la capacidad, la honestidad y la fiabilidad.	Determinar el alcance y los medios necesarios para garantizar que los clientes confíen en las interacciones con otros clientes y en los recursos del SdS.
Confidencialidad	Los datos y servicios sensibles sólo se ponen a disposición de los usuarios designados.	Determinar qué contenido, servicios y aplicaciones son sensibles y el enfoque para garantizar la confidencialidad.
Componibilidad	Los componentes de SdS pueden estar combinados de muchas maneras, según lo definido por los objetivos y requisitos del usuario.	Planificación de la arquitectura que asegura la capacidad de combinar servicios y aplicaciones.
Disponibilidad	Los servicios y recursos de SdS se pueden suponer razonablemente disponibles siempre que sea necesario.	Determinar la disponibilidad necesaria de servicios y recursos que cumplan con las expectativas del cliente.
Elasticidad	La arquitectura del SdS permite el aprovisionamiento automático y la liberación de recursos que cumplen con cargas de trabajo cambiantes.	Planificación de la arquitectura SdS para acomodar la asignación flexible de recursos y que así coincidan con las cargas de trabajo.

Factor	Consideración	Función de la planificación
Escalabilidad	Este atributo está relacionado con la elasticidad. La escalabilidad mide qué tanto el SdS puede adaptarse al aumento de las cargas de trabajo.	Evaluar y planificar la asignación temporal de recursos necesarios para la carga de trabajo de un cliente.
Extensibilidad	El grado en que un SdS puede proporcionar recursos, aplicaciones, servicios y comportamientos futuros.	Planificar una arquitectura del SdS que pueda adaptarse sin problemas a los recursos y servicios futuros.
Gestión de desastres	Preparar y ejecutar procesos y procedimientos para recuperar operaciones en caso de que una parte o la totalidad de un SdS sea destruida por un desastre a gran escala, como un incendio, una inundación o un terremoto.	Un desastre catastrófico puede dañar los datos y los recursos computacionales. Determinar lo que debe ser protegido y un plan para proporcionar esa protección.
Gestión del espectro	Gestión eficiente del espectro electromagnético en redes inalámbricas para evitar interferencias y garantizar la comunicación.	Planificar el uso del espectro electromagnético de manera coherente con los requisitos de comunicación del SdS y las disposiciones reglamentarias.
Interoperabilidad	La capacidad de los sistemas constituyentes para intercambiar y utilizar información y servicios.	Determinar cómo se logrará la interoperabilidad en componentes nuevos y heredados.
Mantenibilidad	El grado en que los recursos en el SdS se pueden mantener para corregir, prevenir y eliminar fallos.	Establecer los procesos, roles, responsabilidades y auditorías necesarios para evaluar y restaurar el rendimiento del SdS
Rendimiento	Asegurar que el SdS cumpla con los requisitos de rendimiento en las condiciones de uso y ambientales esperadas.	Evaluar los requisitos de rendimiento del cliente y planifique la arquitectura y los recursos del SdS necesarios.
Reparabilidad	El SdS tiene características que admiten el diagnóstico y la reparación de recursos fallidos o que funcionan mal.	Relacionados con la capacidad de mantenimiento, los planes que hacen que el SdS sea diagnosticable y reparable, preservando al mismo tiempo la disponibilidad requerida.
Resiliencia	La capacidad del SdS para continuar brindando servicios útiles durante la interrupción y reparar cualquier daño posterior a la interrupción.	Planes para incorporar mecanismos que preserven la funcionalidad en presencia de fallos de recursos y ciberataques.
Seguridad (Safety)	Un SdS que implementa funciones críticas para la seguridad requiere una atención especial para garantizar que esas funciones permanezcan controladas y seguras.	Planear agregar características que protejan las funciones críticas para la seguridad.
Seguridad (Security)	Proteger el SdS del acceso no autorizado y garantizar la integridad y confidencialidad de los datos.	Los planes de ciberseguridad incluyen auditorías, métodos de detección, capacitación de usuarios y pruebas continuas. La planificación de la ciber-resiliencia implica medios para aprender y adaptarse a nuevos ataques.
Tiempo real	Algunas aplicaciones de SdS pueden requerir limitaciones de tiempo para completar servicios o entregar productos.	Planificar la implementación de computación perimetral (edge) o computación en la niebla (fog) para satisfacer las necesidades de aplicaciones de tiempo real estricto.
Usabilidad	El SdS es fácil de usar y fácil de operar	Planificar interfaces de cliente intuitivas y fáciles de usar, como servicios web, aplicaciones REST e interfaces de programación de aplicaciones.
Usuarios y usos	Las necesidades y expectativas de los clientes del SdS impulsan la necesidad de recursos. Un número creciente de usuarios e hilos afectan directamente a la cantidad y el tipo de recursos y su interoperabilidad.	Planificar los cambios y el crecimiento en los servicios y recursos del cliente.
Verificabilidad	Asegurar que el SdS se construyó correctamente y proporciona las capacidades esperadas.	Desarrollar planes para realizar pruebas de extremo a extremo de misiones, capacidades e hilos del SdS.

Tabla 1: Consideraciones de planificación.

2.2. Actividades de planificación de SdS

La Figura 2 muestra la interacción entre las acciones de planificación que se definen en la Tabla 3. La planificación estratégica, táctica y operativa se descompone en sus sub-planes. Los planes de nivel inferior tienden a tener un solo propósito, cuyos detalles se encuentran en Tabla 3. La planificación a largo plazo extiende los objetivos de la planificación estratégica. La planificación de la retirada se desarrolla a partir de la planificación estratégica y táctica. La planificación del ciclo de vida se basa en la planificación táctica. Cada plan puede abordar uno o más de los factores de la Tabla 2.

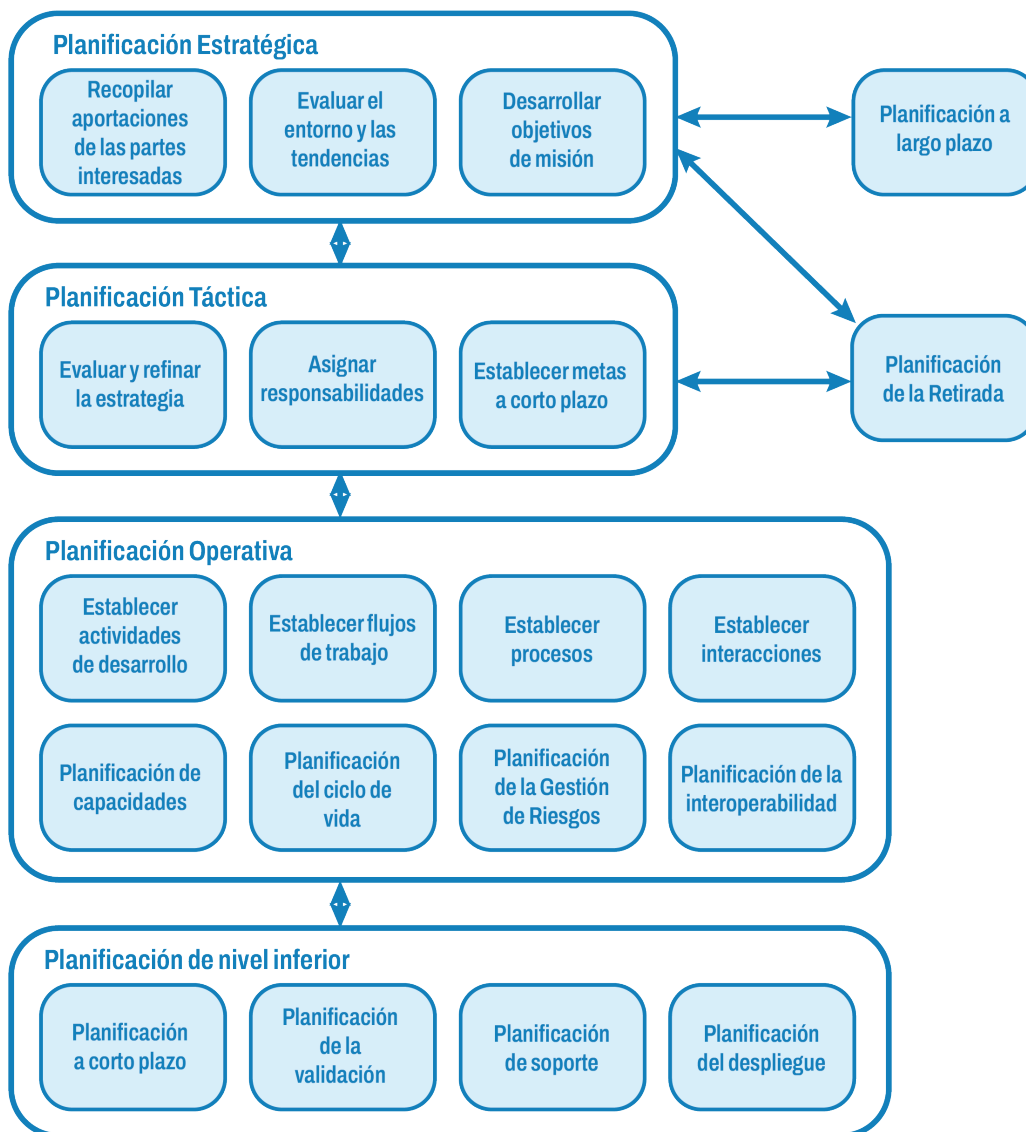


Figura 2: Estructura ontológica que muestra conceptos y relaciones de SdS.

Actividad de planificación	Finalidad	
Estratégica	<p>La planificación estratégica establece el proceso para definir la dirección, las necesidades y los objetivos futuros de una organización. Estos planes se alinean, motivan e involucran a las partes interesadas internas y externas con prioridades estratégicas que reflejan los objetivos y prioridades de la misión. Un plan estratégico es la base para comprender y acomodar la extensibilidad, adaptabilidad, asequibilidad, disponibilidad, resiliencia, escalabilidad, seguridad, confidencialidad y protección. Los planes estratégicos son a menudo estáticos o mínimamente cambiados durante la vida de un SdS. A medida que los planes estratégicos definen el caso de negocio y las misiones en general, rara vez se cambian, aunque las capacidades y los hilos pueden cambiarse.</p> <p>Esta actividad analiza y evalúa la propuesta de valor de SdS, las misiones, las tendencias, las mejores prácticas, los desarrollos actuales y las brechas; desarrolla una estrategia que alinee a las principales partes interesadas con las misiones y los recursos necesarios para alcanzarlas; establece los procesos necesarios para llevar a cabo la estrategia; y supervisa y evalúa el progreso del plan y vuelve a planificar cuando es necesario.</p> <p>El plan incluye una declaración del propósito; análisis de fortalezas, debilidades, oportunidades y amenazas.</p>	
Táctica	<p>Los planes tácticos mapean los planes estratégicos en los objetivos a corto plazo, las estrategias y las microestrategias de organizaciones específicas para lograr esos objetivos. Estos planes incluyen consideraciones de los recursos de procesamiento y almacenamiento del SdS, comunicación, elasticidad, mantenibilidad e interoperabilidad que se esperan del SdS para llevar a cabo servicios y generación de productos críticos en tiempo real y diferido, evaluar la calidad del producto y monitorear el rendimiento y la resiliencia del SdS. Los planes tácticos se adaptan a las nuevas necesidades y realidades durante la vida útil del SdS y son más volátiles que los planes estratégicos.</p> <p>La planificación táctica incluye la creación de objetivos, la asignación de responsabilidades, el establecimiento de plazos, la determinación de recursos y la asignación de tareas.</p>	
Operacional	<p>Los planes operativos establecen el flujo de actividades que logran los planes tácticos. Estos planes a menudo cubren unos meses y proporcionan la orientación que necesitan la gestión empresarial para operar el SdS. La planificación de la operación comprende la planificación detallada de cómo funcionará el SdS en escenarios del mundo real. Esto incluye definir flujos de trabajo, procesos e interacciones del sistema para garantizar operaciones fluidas. Esta actividad inicia la planificación basada en la capacidad, el ciclo de vida, la gestión de riesgos y la interoperabilidad.</p>	
	Basado en capacidades	Se centra en identificar y desarrollar las capacidades requeridas por el SdS. Esto implica definir los resultados deseados y garantizar que los sistemas dentro del SdS puedan lograr estas capacidades.
	Ciclo de vida	Se centra en la gestión de todo el ciclo de vida del SdS, desde el concepto inicial hasta el desarrollo, despliegue, operación y eventual desmantelamiento. Esto garantiza que el SdS siga siendo efectivo y relevante a lo largo de su vida útil. La planificación del ciclo de vida se basa principalmente en la planificación estratégica y táctica.
	Gestión de riesgos	Implica identificar, evaluar y mitigar los riesgos asociados con el SdS. Esto garantiza que los problemas potenciales se aborden de manera proactiva para minimizar su impacto en el SdS. Los riesgos a menudo se expresan como la probabilidad de un evento no deseado y el impacto del evento si sucede.
Interoperabilidad	Asegura que los sistemas dentro del SdS puedan trabajar juntos sin problemas. Los planes implican definir estándares, protocolos e interfaces que faciliten la comunicación y la integración del sistema.	
A largo plazo	<p>Estos planes exploran opciones del SdS y usos potenciales muchos años más allá de los planes estratégicos. La principal distinción entre planes a largo plazo y estratégicos es que los planes estratégicos son los medios para lograr las necesidades futuras esperadas de una organización. Por el contrario, los planes a largo plazo son una forma de lluvia de ideas sin restricciones. Los planes estratégicos están en el camino que desarrolla los requisitos del SdS y los casos de uso, mientras que los planes a largo plazo son candidatos sin fondos para objetivos futuros.</p>	
A corto plazo	<p>Los planes a corto plazo deciden las capacidades disponibles para los clientes del SdS en un momento dado. Estos planes influyen directamente en los requisitos del SdS, las asignaciones de personal y las tareas a corto plazo. Un ejemplo podría ser la planificación de la entrega de un servicio específico utilizado en todo el SdS.</p>	
Validación	<p>Describe las instalaciones, la configuración, los recursos, el equipo de prueba, el personal, los escenarios de prueba y los criterios de éxito asociados con la validación de la funcionalidad y los requisitos del SdS esperados.</p>	

Actividad de planificación	Finalidad
Despliegue	Detalla los planes sobre cómo se implementará el SdS. Las opciones de implementación incluyen la ejecución del nuevo sistema junto con un sistema antiguo, la configuración del nuevo sistema para realizar lentamente las funciones de un sistema antiguo, el cierre de un sistema antiguo y la asunción del nuevo sistema, la modificación gradual de un sistema antiguo al nuevo sistema, el hecho de que el nuevo sistema siga un sistema antiguo pero que esté oculto para los usuarios, o la habilitación y prueba progresivas de componentes del nuevo sistema disponibles para los usuarios cuando no exista un sistema antiguo. La implementación también puede ocurrir con la integración.
Apoyo	Planifica los medios y la dotación de personal para apoyar el funcionamiento del SdS después de su despliegue. El soporte puede incluir funciones de servicio que monitorean el rendimiento, las tasas de error, el uso y las interrupciones. La planificación del apoyo puede consistir en planes para la formación de operadores y usuarios.
Planificación de la retirada	Describe los planes de retiro y disposición del SdS. Retirar un SdS no implica necesariamente retirar los sistemas constituyentes, pero podría eliminar el marco y las utilidades asociadas con la interoperabilidad y el soporte de SdS. Es importante destacar que el plan debe abordar la continuación de los servicios al cliente en otro SdS. La planificación para la retirada comienza con la planificación estratégica y se detalla en la planificación táctica.

Tabla 3: Detalles de las actividades de planificación.

3. PLANIFICACIÓN DE LA INTEROPERABILIDAD

Un aspecto esencial de la planificación de SdS implica comprender la forma y el grado en que los servicios y recursos deben interoperar [6]. La interoperabilidad puede ocurrir a diferentes escalas y tiempos y puede aplicarse de manera diferente a clientes y servicios individuales. Es importante destacar que la interoperabilidad debe existir con sistemas heredados que pueden o no proporcionar los servicios necesarios.

Comúnmente, se pueden implementar cuatro niveles de interoperabilidad en un SdS:

- **La interoperabilidad fundacional** es la forma más simple en la que los datos se comparten entre sistemas en un SdS, pero los datos no se interpretan. Los usuarios interpretan los datos intercambiados para su uso por los clientes de datos. Implica establecer enlaces de comunicación y protocolos que permitan la transmisión de datos entre sistemas.
- **La interoperabilidad estructural o sintáctica** comprende formatos de datos consistentes que todos los sistemas dentro del SdS entienden. Esta forma de interoperabilidad permite a todos los sistemas recuperar, interpretar y procesar información. Asegura que la sintaxis de datos, como los formatos de datos y los protocolos de comunicación, estén estandarizados para que los sistemas entiendan y procesen los datos correctamente.

- **La interoperabilidad semántica** asegura que los sistemas dentro de un SdS comparten una comprensión conceptual común de datos y mensajes. También elimina la posibilidad de malinterpretar la información compartida. Implica definir modelos de datos, vocabularios y ontologías comunes para garantizar que los datos intercambiados tengan el mismo significado para todos los sistemas participantes.
- **La interoperabilidad organizacional** permite el intercambio de datos consistentes a través de un SdS alineado con objetivos, necesidades, expectativas, uso, flujos de trabajo y gobierno empresarial comunes.

Las necesidades de ejecución de hilos y los plazos influyen fuertemente en la interoperabilidad de SdS, lo que afecta las decisiones arquitectónicas de SdS, las opciones de comunicación, la resiliencia, la capacidad de mantenimiento, la seguridad y la disponibilidad.

Las consideraciones y adaptaciones para la interoperabilidad se capturan en los planes tácticos como declaraciones de capacidad que describen fuentes de información, usos y destinos. Por ejemplo, un objetivo de misión para reducir los accidentes de aeronaves se asocia con capacidades para recopilar datos heterogéneos de sensores de aeronaves, recopilar datos meteorológicos, distribuir datos a procesadores de conciencia situacional, toma de decisiones y control de gestión del tráfico aéreo en tiempo real.

La operación y los planes a corto plazo describen los medios técnicos para lograr la interoperabilidad como dictan las

capacidades tácticas. Siguiendo el ejemplo de control de tráfico aéreo anterior, la interoperabilidad puede involucrar intermediarios de servicios y mensajes, middleware, estándares y protocolos de intercambio de datos, servicios web y servidores proxy, entre muchas otras opciones. El control de tráfico en tiempo real necesita intercambios de información garantizados para estar libre de retrasos y cortes prolongados.

La interoperabilidad afecta al rendimiento general de los SdS, mejorando su adaptabilidad, lo que le permite incorporar nuevos sistemas y tecnologías. La planificación debe ser flexible para adaptarse a los cambios y actualizaciones. Además, la interoperabilidad garantiza una integración perfecta de los sistemas constituyentes, lo cual es crucial para un funcionamiento eficaz del SdS. La planificación debe considerar cómo optimizar el rendimiento a través de su interoperabilidad efectiva mitigando o eliminando los desafíos resultantes de:

- Los sistemas heredados que utilizan tecnologías obsoletas y protocolos de comunicación. Este sistema (Legacy) puede ser difícil de interoperar con sistemas que son más nuevos, a menos que se utilicen cambios significativos o middleware especializado para traducir entre la sintaxis y la semántica del sistema heredado y el más nuevo.
- La propia interoperabilidad que requiere protocolos de comunicación estandarizados (por ejemplo, HTTP, FTP, SOAP, REST, TCP/IP, UDP, ICMP, etc.) para facilitar el intercambio de información entre sistemas. La planificación debe definir estos protocolos para garantizar interacciones fluidas. La gestión de datos a escala implica la coordinación y consolidación de datos de múltiples fuentes. Un desafío particular es proporcionar interoperabilidad con datos heredados almacenados en bases de datos aisladas con formatos obsoletos que dependen de un software de gestión de bases de datos también obsoleto.
- Los lenguajes de programación heterogéneos, los sistemas operativos y el hardware utilizados por diferentes sistemas pueden no interoperar fácilmente.
- Los sistemas que pueden utilizar diferentes formatos de datos (por ejemplo, JSON, XML, texto sin formato, CSV, formatos propietarios, etc.). Traducir esos formatos para la interoperabilidad puede ser difícil y puede ralentizar los intercambios.
- Los cuellos de botella de comunicación y procesamiento que pueden ralentizar los intercambios, especialmente a medida que el SdS aumenta. Los cuellos de botella pueden empeorar cuando las capas de middleware agregan sobrecargas y demoras de comunicación adicionales.

- Los malentendidos semánticos, que son quizás el desafío más importante de la interoperabilidad porque impiden una interpretación adecuada de la información intercambiada. Los malentendidos son especialmente problemáticos cuando se intercambian datos de diferentes dominios.

4. ARQUITECTURA Y PLANIFICACIÓN DE LA INTEGRACIÓN

Una arquitectura SdS comprende recursos empresariales y sus conexiones, según sea necesario para proporcionar hilos que alcancen las capacidades de la misión. Acomodar los atributos señalados en el Capítulo 1 impone exigencias adicionales a la planificación arquitectónica. En términos simples, planificar la arquitectura física de un SdS implica elegir opciones que equilibren eficientemente los atributos opuestos. Desafortunadamente, varias realidades complican la planificación de la arquitectura.

Considere una arquitectura federada para ilustrar las complejidades en la estructura y la interoperabilidad de SdS. Una federación es una forma de SdS colaborativa en la que múltiples sistemas autónomos trabajan juntos para lograr un objetivo común. Cada sistema dentro de la federación opera independientemente de los otros sistemas utilizando sus datos, procesos y control. La colaboración se produce debido a interfaces de intercambio e invocación de aplicaciones bien definidas. Los sistemas en federaciones sueltas operan casi completamente independientemente unos de otros y se coordinan a través de intercambios de datos y autenticación. Las federaciones estrechas constituyen una autoridad central para coordinar las actividades de los sistemas autónomos. Las federaciones jerárquicas están estructuradas como federaciones estrechas, pero la autoridad central también determina las políticas empresariales. Las federaciones “peer-to-peer” no tienen autoridad central, y la coordinación se gestiona por consenso y protocolos predefinidos. Por último, las federaciones híbridas son una mezcla de sistemas estrechamente federados y poco federados que una autoridad central puede coordinar.

La planificación de la resiliencia, la disponibilidad, la ciberseguridad, el mantenimiento, la recuperación ante desastres y las adaptaciones de seguridad implica conocer y predecir posibles interrupciones cuando sea posible y contar con procedimientos para gestionar lo inesperado. Otras complicaciones surgen cuando el SdS ejecuta hilos que deben completarse en escalas de tiempo que no pueden tolerar interrupciones prolongadas. El proceso de planificación debe anticipar la necesidad de arquitecturas robustas y desarrollar, probar y revisar planes de contingencia a medida que ocurren condiciones de fallos inesperados y nuevos ciberataques.

Los proveedores comerciales de nube ofrecen herramientas, servicios y recomendaciones de arquitectura para construir SdS robustos y receptivos. Un plan de arquitectura podría incluir un análisis de costo-beneficio para involucrar a un proveedor de nube comercial en lugar de construir un SdS interno.

La planificación de la arquitectura también debe tener en cuenta los usos fuera de las aplicaciones de la misión, como el desarrollo y la implementación de aplicaciones y servicios, la detección de servicios, el mantenimiento, la supervisión del rendimiento y la seguridad, y la auditoría. Los proveedores de nube comerciales también ofrecen estas capacidades. Un plan podría incluir la evaluación de la relación costo-beneficio de utilizar un proveedor de servicios comerciales o la creación de las capacidades necesarias internamente.

La planificación de un SdS configurado como una nube necesita un proceso para determinar el tipo de nube, por ejemplo, pública, privada, híbrida o comunitaria:

- Los entornos de nube pública utilizan la infraestructura y los servicios de tecnología de la información proporcionados por proveedores externos a través de Internet. Los recursos de la nube pública se comparten entre múltiples organizaciones e individuos. Las características clave incluyen:

Escalabilidad	Escalabilidad prácticamente ilimitada para satisfacer diferentes demandas
Eficiencia de costes	Funciona con un modelo de pago por uso, lo que reduce los costes iniciales
Accesibilidad	Accesible desde cualquier lugar con conexión a Internet
Gestión	El proveedor de la nube gestiona y mantiene la infraestructura

- Las nubes privadas están dedicadas a una sola organización, proporcionando acceso exclusivo a recursos informáticos. Pueden ser alojadas en las instalaciones o por un proveedor externo. Las características clave incluyen:

Seguridad	Mayor seguridad y control sobre los datos y las aplicaciones
Personalización	Adaptado para satisfacer las necesidades específicas de la organización
Cumplimiento	Más fácil de cumplir con los requisitos reglamentarios y de cumplimiento
Gestión	Puede gestionarse internamente o subcontratarse

- Las nubes híbridas combinan nubes públicas y privadas, lo que permite compartir datos y aplicaciones entre ellas. Este enfoque ofrece flexibilidad y optimización de la infraestructura existente. Las características clave incluyen:

Agilidad	Capacidad para adaptarse rápidamente a las necesidades cambiantes del negocio
Escalabilidad	Aprovecha la escalabilidad de las nubes públicas al tiempo que mantiene el control sobre los datos críticos en las nubes privadas
Optimización de costes	Equilibra el coste y el rendimiento utilizando el entorno más adecuado para cada carga de trabajo
Continuidad del negocio	Mejora la resiliencia mediante la distribución de cargas de trabajo en múltiples entornos

- Las nubes comunitarias se comparten entre múltiples organizaciones con requisitos similares, como el cumplimiento normativo o las necesidades de seguridad. Las organizaciones participantes o un tercero las gestionan. Las características clave incluyen:

Recursos compartidos	Distribución rentable de los recursos entre los miembros de la comunidad
Cumplimiento	Adaptado para satisfacer las necesidades regulatorias y de cumplimiento específicas de la comunidad
Colaboración	Facilita la colaboración y el intercambio de datos entre organizaciones con objetivos comunes
Seguridad	Medidas de seguridad mejoradas adaptadas a las necesidades de la comunidad

Un SdS que ejecuta hilos en tiempo real o críticos para el tiempo puede necesitar considerar arquitecturas de alto rendimiento y baja latencia, como arquitecturas de borde o perimetral (edge) o niebla (fog). La computación perimetral reduce la latencia al mover físicamente la computación y el almacenamiento de datos cerca de la fuente de datos o el usuario. La computación de niebla reduce la latencia mediante el uso de computación distribuida geográficamente que extiende la computación en la nube a los bordes de la red. Un plan debe incluir un proceso para evaluar las opciones arquitectónicas relacionadas con el desempeño de los hilos de misión a corto plazo y futuros.

Por último, los recursos, aplicaciones y servicios de SdS están conectados de manera coherente con los atributos de la Sección 1. Otros factores [2] que afectan a la integración de SdS se muestran en la Tabla 4.

Factor	Propósito
Partes interesadas	Las partes interesadas tienen diversas necesidades y objetivos que deben estar alineados para una integración exitosa del SdS. Sus requisitos impulsan el diseño y la funcionalidad del SdS, asegurando que el sistema integrado cumpla con los objetivos colectivos. Además, incluyen a los clientes de productos de datos de misión, proveedores de información necesaria para producir productos de datos de misión, monitores de control de calidad, desarrolladores, equipos de verificación y validación, especialistas en seguridad, contables, personal de mantenimiento de sistemas, auditores y gerentes que necesitan acceso a datos, estado y recursos físicos específicos del SdS.
Desarrollo de arquitectura	Además de definir la composición física y de software, un plan debe identificar el proceso mediante el cual se desarrolla y revisa una arquitectura de SdS, así como las operaciones necesarias para asignar funciones y datos. La arquitectura define la estructura y la interacción de los sistemas constituyentes dentro del SdS. Una arquitectura bien diseñada facilita una integración perfecta, asegurando que los sistemas puedan trabajar juntos de manera efectiva.
Recursos de integración	La integración de SdS a veces requiere equipos especiales, instalaciones y personal para la instalación y el pago. Los recursos adecuados, incluidos los fondos, el personal y la tecnología, son cruciales para la integración. Los recursos limitados pueden dificultar la integración, lo que lleva a retrasos y a un rendimiento subóptimo.
Procesos de integración	Los procesos de integración comprenden los pasos para realizar la integración utilizando recursos de integración. Las descripciones iniciales del proceso suelen ser lo suficientemente detalladas como para determinar los recursos de integración. Los detalles del proceso se desarrollan durante la operación y la planificación a corto plazo.
Requisitos de integración	Las funciones de la misión tendrán requisitos relacionados con la precisión, el rendimiento, la capacidad, la puntualidad, la seguridad, etc. La planificación debe establecer cómo se obtienen, validan, rastrean y asignan los requisitos.
Influencias externas	Factores externos como los cambios regulatorios, los avances tecnológicos y la dinámica del mercado pueden afectar la integración de SdS. Estas influencias deben ser consideradas y gestionadas para asegurar una integración exitosa.
Gestión de riesgos	Debe elaborarse un plan de gestión de riesgos para todos los aspectos de la declaración de fiabilidad. En cuanto a la integración, los riesgos pueden estar asociados con entregas tardías, fallos y defectos, inconsistencias y errores de montaje. Un plan de gestión de riesgos proporciona el proceso para la recopilación, la contabilidad y la retirada de los riesgos.
Verificación y Validación (V&V)	El tipo de SdS determinará en qué medida es posible la V&V. Como mínimo, los sistemas deben realizar V&V en los protocolos de interfaz para garantizar el cumplimiento de los acuerdos y estándares del SdS. Un plan V&V explica lo que se comprueba y los criterios de éxito. Además, la planificación debe indicar los roles y responsabilidades de V&V y el proceso para documentar y adjudicar los fallos de V&V.
Adaptación y Reutilización	La adaptación implica personalizar los procesos de integración para satisfacer necesidades específicas, mientras que la reutilización aprovecha los componentes y procesos existentes para ahorrar tiempo y recursos. Ambas prácticas mejoran la eficiencia y reducen los costos de integración.
Certificación y Acreditación	La certificación y la acreditación garantizan que el SdS cumple con las normas y regulaciones pertinentes. Este proceso valida la seguridad, la protección y el rendimiento del SdS, asegurando a las partes interesadas.

Tabla 4: Factores que afectan la integración de SdS.

Un plan debe definir el proceso para seleccionar la integración de SdS más adecuado para la ejecución de hilos, como se muestra en la Tabla 5.

Opción	Propuesta
Integración de datos	La integración de datos se centró en mantener la coherencia de los datos en todo el SdS. Idealmente, los datos deben estandarizarse en todos los sistemas. Sin embargo, pueden emplearse middleware o servicios similares cuando la normalización a nivel de sistema no es factible.
Punto a punto	La integración punto a punto comprende conexiones directas entre recursos. Este método es sencillo, pero inflexible, lo que lleva a complejidades a medida que el SdS crece.
Estrella	La integración de estrellas se asocia con la creación de conexiones directas entre todos los recursos. Proporciona un alto grado de conectividad, pero puede ser difícil de manejar.
Modelo Radial (Hub-and-Spoke)	La integración de un modelo radial (Hub-and-spoke) tiene un nodo central (hub) que conecta recursos. Simplifica la gestión y reduce el número de conexiones necesarias.
Vertical	La integración vertical conecta sistemas dentro de una organización. Este enfoque es adecuado dentro de los mismos límites organizacionales.
Horizontal	La integración horizontal conecta sistemas entre organizaciones. Es ideal para integrar sistemas que necesitan colaborar más allá de los límites de la organización.
Middleware	Middleware es un software de pegamento que suaviza la conectividad entre los componentes de software que pueden no compartir la semántica de datos o la sintaxis comunes. Es útil para integrar sistemas y datos heterogéneos.
SOA (Arquitectura orientada al Servicio)	La integración orientada al servicio conecta los servicios con los clientes a través de un intermediario de servicios, ocultando los detalles de la transacción y la ubicación del cliente y del servicio. Descripción Universal Descubrimiento e Integración (UDDI) es un estándar para especificar, publicar y descubrir servicios web.
SOAP (Protocolo simple de acceso a objetos)	Es un protocolo de mensajería de servicio web independiente del transporte que es adecuado para escenarios que requieren seguridad robusta y cumplimiento de transacciones.
REST (Transferencia de estado representacional)	Es un protocolo de comunicaciones cliente-servidor simple, escalable, flexible, sin estado. REST es ideal para aplicaciones basadas en la web que requieren una comunicación ligera y eficiente.

Tabla 5: Opciones de ejecución de hilos

Idealmente, el enfoque de arquitectura e integración elegido debería proporcionar capacidades para los primeros usuarios sin restringir el crecimiento futuro. Existe una compensación entre los costes iniciales de diseño e implementación frente al aumento de los costes del ciclo de

vida. Crear arquitecturas flexibles implica mecanismos de integración más complicados con mayores costes y riesgos iniciales. Sin embargo, las arquitecturas restrictivas que son menos costosas y arriesgadas al principio serán más caras de actualizar en el futuro.

5. CONCLUSIÓN

Este breve capítulo presentó una visión general de los factores y consideraciones de compromiso necesarios para planificar el desarrollo de un Sistema de Sistemas (SdS). Se abordaron múltiples fases de planificación, coherentes con la ingeniería de sistemas tradicional. Cada fase de planificación define procesos para extraer y evaluar niveles cada vez más detallados de diseño y operación.

REFERENCIAS

1. Boardman, J. and Sauser, B., "System of Systems – The Meaning of," Proc. IEEE/SMC Int. Conf. Systems Engineering, Los Angeles, CA, 2006, pp. 118-123
2. Madni, Azad and Sievers, M., "System of Systems Integration: Key Considerations and Challenges," INCOSE Systems Engineering, 17 (3), 2014, pp. 330-347
3. INCOSE Systems of Systems Primer, INCOSE-TP-2018-003-01.0
4. Maier, M.W. "Architecting Principles for System of Systems, Systems Engineering, 1 (4) 1998, pp. 267-284
5. Sage, A.P., and C.D. Cuppan. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems," Information, Knowledge, Systems Management, Vol. 2, No. 4, 2001, pp. 325-345
6. Diallo, Saikou & Herencia, Heber & Padilla, José & Tolk, Andreas. (2011). Understanding interoperability," EAIA '11: Proceedings of the 2011 Emerging M&S Applications in Industry and Academia Symposium, Boston, MA, April 3-7, pp. 84-91

BIOGRAFÍAS

DR. MICHAEL SIEVERS

El Dr. Michael Sievers obtuvo un doctorado en Ciencias de la Computación y títulos de maestría y licenciatura en Ingeniería Eléctrica, todos en la Universidad de California, Los Ángeles (UCLA). Actualmente es ingeniero de sistemas sénior en el Laboratorio de Propulsión a Chorro (JPL) del Instituto de Tecnología de California y forma parte del cuerpo docente de ingeniería de sistemas de la Universidad del Sur de California. Está especializado en diseño de misiones para sistemas espaciales y terrestres, tolerancia a fallos y resiliencia, sistemas de información de extremo a extremo, software de sistemas, computación de alto rendimiento, control y manejo de datos, ingeniería de sistemas basada en modelos (MBSE), y confianza y reputación en sistemas. Actualmente lidera el desarrollo de métodos de resiliencia para un gran sistema de sistemas del gobierno de los Estados Unidos y enseña cursos sobre integración de sistemas, resiliencia de sistemas críticos y aprendizaje automático. El Dr. Sievers es fellow de INCOSE, associate fellow de AIAA y miembro sénior del IEEE.



PABLO MARTICORENA

Pablo Marticorena es Ingeniero Superior en Informática por la Universidad Autónoma de Madrid e Ingeniero Técnico en Informática de Gestión por la Universidad Politécnica de Madrid. A lo largo de su carrera, ha desempeñado múltiples funciones técnicas y de gestión, tanto fuera como dentro de Isdefe, incluyendo administración de Data Warehouse, desarrollos SQL/e-CODEX, digitalización de las administraciones públicas, gestión de proveedores tecnológicos, especificación de requisitos, verificación/validación y análisis RAMS. En su trabajo principal, se ha consolidado como un experto en Seguridad y Garantía del Software, acumulando más de 22 años de experiencia en el sector de Gestión de Tráfico Aéreo. Ha brindado apoyo en múltiples áreas de automatización, asegurando el cumplimiento normativo de los cambios realizados en distintos sistemas críticos de control, además de colaborar en la integración de metodologías dentro de los ciclos de vida de estos sistemas. Actualmente, aporta su experiencia en la ingeniería de requisitos al programa iTEC SkyNex, donde Isdefe colabora con ENAIRE e INDRA para la integración de los avances tecnológicos en el Control del Tráfico Aéreo, creando un entorno ATM unificado, eficiente y sostenible. Finalmente, cabe destacar que sigue apasionado por su profesión y por contribuir al desarrollo de España y Europa mediante las competencias técnicas adquiridas y sus habilidades interpersonales. También participa como formador interno en Isdefe y mantiene la certificación CSEP por INCOSE.





Pruebas y evaluación de Sistemas de Sistemas

Mark Phillips, *Universidad de Nueva Gales del Sur* (mark.phillips@unsw.edu.au)

Dr. Keith Joiner, *Universidad de Nueva Gales del Sur* (k.joiner@unsw.edu.au)

Aurelio Fernández Sáez, *Isdefe* (afernandez@isdefe.es)

Manuel Fernández Astaburuaga, *Isdefe* (mfastaburuaga@isdefe.es)

Resumen

Las actividades de prueba y evaluación (T&E, por sus siglas en inglés) de un Sistema de Sistemas (SdS) presentan desafíos únicos que difieren significativamente de los encontrados en las pruebas tradicionales a nivel de sistema. Este capítulo presenta un enfoque integral para la T&E de SdS, haciendo hincapié en la necesidad de marcos adaptativos, técnicas sólidas de modelado y simulación y la integración del contexto operativo a lo largo de todo el ciclo de vida de las pruebas. Entre las consideraciones clave se incluyen la interoperabilidad, la configuración dinámica y la evaluación del rendimiento en escenarios realistas orientados a validar la misión. El capítulo también aborda métricas de éxito, estrategias de validación entre sistemas heterogéneos y la importancia de la colaboración entre las partes interesadas. El objetivo es garantizar que el SdS cumpla con los requisitos generales de la misión, manteniendo al mismo tiempo la fiabilidad, escalabilidad y resiliencia en entornos complejos.

Palabras clave

Sistema, Prueba, Evaluación, Sistema de Sistemas (SdS).



1. INTRODUCCIÓN

Este capítulo proporciona orientación sobre cómo realizar pruebas y evaluaciones (T&E, por sus siglas en inglés) de Sistemas de Sistemas (SdS). La T&E de un SdS es un proceso complejo y dinámico que involucra a múltiples partes interesadas, integrando sistemas multipropietario que, aunque independientes, son interdependientes para alcanzar un objetivo común. A diferencia de las pruebas tradicionales de sistemas, la T&E de SdS debe abordar la aparición de comportamientos emergentes, desafíos de interoperabilidad y requisitos operativos en continua evolución [1, 2].

Desde la perspectiva del cliente, la T&E garantiza que el SdS cumpla con los objetivos de la misión, la efectividad operativa (éxito de la misión) y la idoneidad dentro de las restricciones del mundo real. Los clientes, típicamente agencias gubernamentales o grandes empresas, se centran en validar el rendimiento, mitigar riesgos y asegurar el cumplimiento de normas regulatorias y de seguridad [3]. Por otra parte, desde la perspectiva del contratista, la T&E de SdS implica equilibrar la integración de sistemas, los costes, los cronogramas y la viabilidad técnica. Los contratistas son responsables de cumplir con los requisitos contractuales mientras aseguran que los sistemas constituyentes funcionen de manera cohesionada dentro de la arquitectura general del SdS. Deben navegar entre las necesidades cambiantes del cliente, las incertidumbres tecnológicas y las limitaciones de interoperabilidad entre distintos proveedores y sistemas heredados.

Una T&E efectiva de un SdS requiere, en la mayor medida posible, la delimitación del rendimiento predecible frente a los fenómenos emergentes que pueden aparecer y aparecerán a lo largo del ciclo de vida. Esto exige una metodología de prueba rigurosa, que incluya modelado y simulación, pruebas en vivo y evaluaciones iterativas durante toda la vida útil del SdS y de cada uno de sus sistemas constituyentes [4]. Tanto clientes como contratistas deben participar en una planificación colaborativa, aprovechando estrategias de T&E ágiles y adaptativas para abordar las complejidades de los entornos SdS.

2. CONCEPTOS QUE AFECTAN A LAS PRUEBAS DE UN SDS

La diferencia en el tratamiento entre los sistemas tradicionales y los SdS en cuanto a T&E puede ser considerable. Por lo tanto, se debe tener en cuenta esta diferencia al desarrollar los planes de prueba y evaluación. Generalmente, los sistemas constituyentes que se integran presentan distintos niveles de madurez en términos de tecnología, fabricación y capacidad para interoperar. Esto supondrá un reto a la hora de la sincronización de los distintos ciclos de vida para poder garantizar que el SdS pueda ser probado, con unas líneas de referencia asignadas estables de configuración y funcionales. También requerirá una estrecha coordinación con el (los) cliente(s) como parte del proceso de gobernanza del SdS.

Como se muestra en la Tabla 1, es posible identificar áreas en las que se debe prestar especial atención debido a los problemas de integración que surgen al construir un SdS. Cabe señalar que algunas de estas consideraciones también pueden aplicarse a sistemas tradicionales.

Es crucial gestionar la línea de referencia general del SdS en coordinación con las líneas de referencia de los sistemas constituyentes. No hacerlo puede llevar a sobrecostes y a no cumplir con los objetivos de la misión.

	Ingeniería de Sistemas Tradicional	Ingeniería de Sistemas de Sistemas	Consideraciones de T&E
Propósito	Desarrollo de un solo sistema para cumplir con las necesidades de las partes interesadas y con el rendimiento definido.	Desarrollar nuevas capacidades de SdS aprovechando las sinergias de sistemas heredados.	Considerar el modelo de gobernanza y los requisitos de recursos para el/los cliente/s.
Arquitecturas de Sistemas	La arquitectura del sistema se establece al principio del ciclo de vida y permanece relativamente estable.	Reconfiguración dinámica de la arquitectura a medida que cambian las necesidades; uso del enfoque de arquitectura orientada a servicios como habilitador.	Definir los límites del SdS y proporcionar un Plan Maestro de T&E que preste mucha atención a la interoperabilidad entre los sistemas constituyentes y la satisfacción de la misión del SdS.
Interoperabilidad de Sistemas	Define e implementa requisitos específicos de interfaz para integrar componentes en el sistema.	Los sistemas constituyentes pueden operar de manera independiente del SdS de manera útil. Los protocolos y estándares son esenciales para habilitar sistemas intercambiables.	Desarrollar representaciones tempranas de modelos de arquitectura que identifiquen las métricas de interoperabilidad necesarias para evaluar su idoneidad para el propósito. Integrar un SdS puede disminuir el rendimiento general. Vincular sistemas a capacidades de misión e identificar medidas de umbral y objetivo.
Adquisición y Gestión	Adquisición y gestión centralizadas del sistema.	Los sistemas constituyentes se adquieren por separado y continúan siendo gestionados como sistemas independientes.	Asegurar que el modelo de gobernanza para el SdS esté dotado de recursos para desarrollar y ejecutar eficazmente un plan de T&E. Considerar el impacto en los sistemas constituyentes y la necesidad imprevista de tener en cuenta los cambios de línea de referencia. Esto implica que puede no ser posible realizar pruebas intrusivas en los sistemas constituyentes o detener los servicios para realizar pruebas en el límite del sistema, ya que el comportamiento del SdS cambiará.

Tabla 1: Consideraciones de T&E para SdS (Adaptado a partir de [5]).

3. GESTIÓN DE RIESGOS EN PRUEBAS DE SDS

La T&E de un SdS es un proceso inherentemente complejo y dinámico que requiere un enfoque estructurado pero flexible. Desde la perspectiva del cliente, el proceso de la T&E del SdS garantiza que se cumplan los objetivos de la misión, la seguridad y la efectividad operativa. Desde el punto de vista del contratista, la gestión de riesgos técnicos, de coste y de plazo es un desafío crítico. Empleando una combinación de modelado y simulación, pruebas ágiles, evaluaciones de interoperabilidad y estrategias adaptativas de gestión de riesgos, las partes interesadas pueden mejorar la confiabilidad y el rendimiento de los SdS. A medida que la tecnología avanza, es probable que las futuras metodologías de prueba de SdS incorporen inteligencia artificial, ingeniería digital y marcos de prueba autónomos para mejorar aún más la eficiencia y la efectividad.

La gestión de riesgos es integral para el T&E de un SdS, ya que las interdependencias del sistema pueden introducir errores no intencionados o riesgos críticos para la misión. Tanto los clientes como los contratistas deben colaborar en la identificación de riesgos, su mitigación y la planificación de contingencias a lo largo de todo el proceso de T&E.

3.1. Identificación de riesgos emergentes

El comportamiento emergente es un rasgo distintivo de un SdS, donde los componentes individuales pueden funcionar correctamente de forma aislada, pero causar consecuencias no deseadas cuando se integran. Normalmente, los sistemas constituyentes ya están operativos y la diferencia entre integrar sistemas nuevos y aquellos que ya están en operación significa que sus comportamientos cambiarán; ya sea de forma positiva o negativa, el comportamiento emergente será la base de dicho descubrimiento. Identificar estos riesgos de manera temprana a través del análisis de amenazas del sistema de sistemas (SoSHA, por sus siglas en inglés) y el análisis de modos y efectos de fallo (FMEA, por sus siglas en inglés) es esencial [3].

3.2. Coordinación y comunicación con los interesados

La gestión de riesgos del SdS requiere una fuerte coordinación entre agencias gubernamentales, contratistas y proveedores externos. La existencia de canales de

comunicación claros y registros de riesgos compartidos facilitan que las partes interesadas alineen las prioridades en el tratamiento de riesgos y sus estrategias de mitigación [4]. Este reto es más desafiante para los SdS que para los sistemas tradicionales, ya que el poder de cada parte interesada variará, al igual que la etapa del ciclo de vida de cada sistema constituyente y los recursos programados para la remediación e integración. Herramientas como el análisis de partes interesadas pueden ayudar a mitigar problemas desde el principio mediante la identificación del poder y la influencia, conduciendo a un modelo de gobernanza más coherente.

3.3. Estrategias adaptativas de mitigación de riesgos

A diferencia de las pruebas de sistemas tradicionales, las estrategias de mitigación de riesgos de los SdS deben ser adaptativas e iterativas, ya que los sistemas constituyentes rara vez están alineados en la etapa de desarrollo, actualización, obsolescencia y capacidad. La implicación de esto es que el TEMP (Plan Maestro de T&E, por sus siglas en inglés) debe ser de naturaleza dinámica. La monitorización continua, los análisis predictivos y los marcos de pruebas de respuesta rápida ayudan a detectar y abordar los riesgos a medida que surgen [2].

3.4. Equilibrar costes, programación y prestaciones de los riesgos

Los contratistas a menudo enfrentan desafíos para equilibrar las restricciones de costes, la planificación y los requisitos de prestaciones técnicas. Los marcos de toma de decisión en base a los riesgos ayudan a priorizar los esfuerzos de pruebas, asegurando que se aborden los riesgos críticos sin exceder el presupuesto ni los plazos. Los SdS tienen el problema adicional de la competencia mutua entre los cronogramas de cada sistema constituyente, según sus propias gobernanzas.

3.5. Riesgos regulatorios y de cumplimiento normativo

El cumplimiento de las regulaciones gubernamentales, los estándares de seguridad y los requisitos de ciberseguridad añaden otra capa de complejidad a las pruebas y evaluaciones del SdS. Adherirse a las directivas del Departamento de Defensa (DoD, por sus siglas en inglés), la regulación de la UE, la regulación nacional como el Esquema Nacional de Seguridad (ENS), estándares ISO como ISO 27001 e ISO 29119, u otras directrices regulatorias es esencial para mitigar riesgos legales y operativos [3].

4. CONSIDERACIONES PARA EL APOYO A LAS PRUEBAS Y EVALUACIÓN DE SDS

Las consideraciones para apoyar el T&E de un SdS son variadas. Aunque constantemente se abordan con nuevas tecnologías, los principios de estas consideraciones se conocen desde hace más de una década. Por ejemplo, la siguiente lista ha sido adaptada de la referencia [6], incluyendo varias citas directas en el recuadro gris.

4.1. Traducir objetivos de capacidad

Para una misión, los objetivos de capacidad del SdS se declararán a un alto nivel; estas capacidades impulsan los objetivos del SdS, realizados a través del cumplimiento de medidas de rendimiento y efectividad operacionales (MOP, MOE, por sus siglas en inglés, respectivamente). Como ya se mencionó, mientras que los sistemas tradicionales tendrán medidas de rendimiento técnico (TPM, por sus siglas en inglés) y parámetros clave de rendimiento (KPP, por sus siglas en inglés) alineados con requisitos funcionales bien definidos, generalmente, un SdS no los tendrá, debido a la integración de sistemas constituyentes, muchos de los cuales ya habrán sido desplegados e, incluso, estarán en fase de mantenimiento de su ciclo de vida. Podría ser útil implicar a los clientes en el proceso de desarrollo de capacidades y la comprensión de alternativas requeridas.

4.2. Monitorizar y evaluar cambios

Los SdS se rigen de manera diferente a los sistemas tradicionales. La integración de sistemas constituyentes, que ya cuentan con modelos de gobernanza establecidos, hará que su integración y modificación sea un desafío. Esto requiere una cuidadosa sincronización de los cambios, ya que cada sistema constituyente tiene su propio ciclo de vida.

4.3. Comprender los sistemas y sus relaciones

Los sistemas constituyentes están conectados con distintos grados de complejidad y comprender sus interrelaciones es esencial. Es necesario tener una comprensión clara de los flujos de control e información dentro del SdS. Aunque estos desafíos pertenecen principalmente al ámbito de la ingeniería y el desarrollo, los costes asociados a la integración y acciones correctivas son relevantes para los clientes al poder afectar significativamente a los plazos del proyecto y la asignación de recursos.

4.4. Desarrollar y hacer evolucionar las arquitecturas del sistema

Identificar y comprender los límites del SdS es esencial, junto con la identificación y representación de todas las interconexiones y la descripción precisa de la arquitectura global. Es probable que el SdS evolucione tanto en diseño como en arquitectura con el tiempo y esta posibilidad de cambio debe anticiparse y abordarse durante la fase de diseño. La responsabilidad de estos esfuerzos debe ser compartida equitativamente entre el equipo de desarrollo y la organización de gobernanza del cliente.

4.5. Evaluar las prestaciones de un SdS

Las MOP y MOE del SdS deben estar claramente definidas y ser bien comprendidas. Dado que los entornos SdS suelen ser multipropietario y abarcan múltiples versiones, debe establecerse claramente la responsabilidad sobre el control del diseño global, especialmente en lo que respecta a las Medidas de Sostenibilidad —MoS (por sus siglas en inglés)—, como la formación, el mantenimiento y los repuestos. Es esencial determinar si el SdS puede alcanzar sus objetivos de misión mediante la integración de las capacidades deseadas. Todas las “prestaciones” relevantes (incluyendo interoperabilidad, escalabilidad, flexibilidad, resiliencia, adaptabilidad, sostenibilidad y seguridad) deben estar descritas explícitamente y revisarse con cada actualización. Estas expectativas deben ser establecidas por el cliente.

4.6. Orquestrar mejoras para un SdS

Debe garantizarse de forma coherente la sincronización del rendimiento y la mantenibilidad de todos los sistemas constituyentes, en continua evolución. Es necesario identificar las futuras actualizaciones tanto de los sistemas constituyentes como del SdS en su conjunto, junto con su posible impacto en la capacidad operativa. Una eficaz planificación y asignación de recursos para estas actualizaciones es esencial. Para lograrlo, se requiere un modelo de gobernanza sólido con una alineación entre todos los clientes, que permita al contratista implementar los cambios necesarios de manera eficiente.

4.7. Abordar las opciones de capacidad y solución

El SdS debe estar alineado para soportar una misión definida, lo que influye directamente en los MOE y MOP requeridos. Es probable que sean necesarias modificaciones en los sistemas constituyentes para garantizar una integración exitosa dentro del SdS. Estas modificaciones pueden evaluarse en función de las “prestaciones” relevantes. Esto sigue siendo una preocupación clave para el cliente.

Esto también implica la necesidad de generar métricas que definan las capacidades de SdS de extremo a extremo y que proporcionen un “punto de referencia” para el desarrollo del SdS. Desarrollar estas métricas y recopilar datos (evidencias) para evaluar el estado del SdS se logra como parte del elemento central del SdS desde un punto de vista sistémico, evaluando en qué medida el rendimiento del SdS cumple con los objetivos de capacidad y, por tanto, con la misión a lo largo del tiempo.

4.8. Comportamiento emergente

El comportamiento emergente surge de la combinación del comportamiento y las propiedades de los elementos del sistema constituyente y de la estructura, a través de las interacciones permitidas entre los sistemas constituyentes, y puede ser desencadenado o influenciado por un estímulo proveniente del entorno del sistema. Por lo tanto, los ingenieros de sistemas deben evaluar y comprender el entorno externo a las fronteras del SdS y considerar estas interacciones lo máximo posible. La aparición de un comportamiento emergente puede no predecirse y puede tener un impacto negativo en el cronograma y en el presupuesto.

Perspectiva de T&E para SdS:

Los objetivos de capacidad no son “requisitos específicos” evaluados a través de KPP. Los objetivos de capacidad son un punto de partida para desarrollar una declaración de expectativas a nivel de SdS y requieren una mayor especificación y elaboración para llevar a cabo pruebas y evaluaciones (T&E).

La comunicación es clave, al igual que comprender cómo se gobiernan los sistemas constituyentes y quién tiene influencia y recursos. Los cambios en los sistemas constituyentes suelen ser asíncronos respecto al SdS. Es fundamental asegurar que tanto los aspectos de capacidad como los aspectos técnicos del SdS se aborden en igual medida.

El SdS no puede descomponerse fácilmente para ser probado y, de hecho, no debería descomponerse al poder afectar al rendimiento integrado. El SdS debe probarse en su forma integrada como SdS; de lo contrario, su comportamiento puede, muy probablemente, cambiar.

Delimitar el SdS y realizar pruebas considerando el SdS como un todo, independientemente del comportamiento de los sistemas constituyentes. Mantener la configuración y ajustarla conforme las necesidades de capacidad impulsan cambios en la arquitectura del SdS.

Asegurar que las MOP y MOE del SdS sean plenamente comprendidas y acordadas tanto por el propietario de la capacidad como por los responsables de los sistemas constituyentes. Evaluar si el SdS puede cumplir su misión según lo diseñado. ¡La medida puede ser tanto subjetiva como objetiva!

Los cambios en los sistemas constituyentes pueden impactar el rendimiento del SdS. Planificar y orquestar actualizaciones o mejoras al SdS será esencial para las pruebas y modificarán el rendimiento y afectarán el comportamiento.

Puede haber necesidades en conflicto entre los sistemas constituyentes y el SdS. Los objetivos de capacidad del SdS proporcionan una base para identificar los sistemas que soportan al SdS, desarrollar una arquitectura del SdS y recomendar cambios o añadidos a los sistemas para satisfacer dichas capacidades. También constituyen la base para definir y medir el rendimiento y la efectividad del SdS a alto nivel.

Dado que el comportamiento emergente podría ser impredecible, la mejor estrategia es asignar presupuesto de forma consecuyente. Obtener conocimiento a partir de integraciones similares puede ayudar a estimar la magnitud del impacto, aunque no necesariamente a identificar su origen o mecanismo.

5. METODOLOGÍAS DE PRUEBAS

Las pruebas y la evaluación de un SdS requieren un enfoque distinto al de los métodos tradicionales de verificación de sistemas, debido a la naturaleza compleja de los entornos SdS. A diferencia de los sistemas individuales, que pueden probarse en condiciones aisladas y controladas, un SdS se caracteriza por comportamientos emergentes, control distribuido y gestión del ciclo de vida independiente de sus sistemas constituyentes [1, 2]. Estos aspectos de un SdS socavan especialmente la confianza en los siguientes enfoques:

- Desarrollar una vez y mantener durante todo el ciclo de vida.
- Asegurar exhaustivamente una vez y mantener durante todo el ciclo de vida.
- Pruebas de aceptación únicas.
- Externalizar el desarrollo y realizar pruebas de integración solo una vez al momento de la entrega.
- Prueba operativa única.

Además [7]:

“Tradicionalmente, la teoría de sistemas ha enfatizado la distinción entre la estructura interna de un sistema y su comportamiento externo, siendo este último derivado del primero. Este enfoque ha llevado a centrarse en representaciones estructurales de los sistemas en los métodos de protección en ciberseguridad. Sin embargo, los CPS [sistemas ciberfísicos, por sus siglas en inglés] son de naturaleza heterárquica y están compuestos por múltiples elementos diversos que interactúan tanto de forma independiente como interdependiente. Como resultado, los métodos tradicionales de descomposición y predicción son inadecuados para capturar la complejidad de los CPS. En los sistemas complejos, la estructura y la función están intrínsecamente vinculadas y las características estructurales de un sistema determinan sus procesos y comportamientos.”

Una ilustración del aseguramiento en una organización heterárquica de sistemas frente a una organización jerárquica se muestra en la Figura 1, para ilustrar los diferentes enfoques de aseguramiento requeridos.

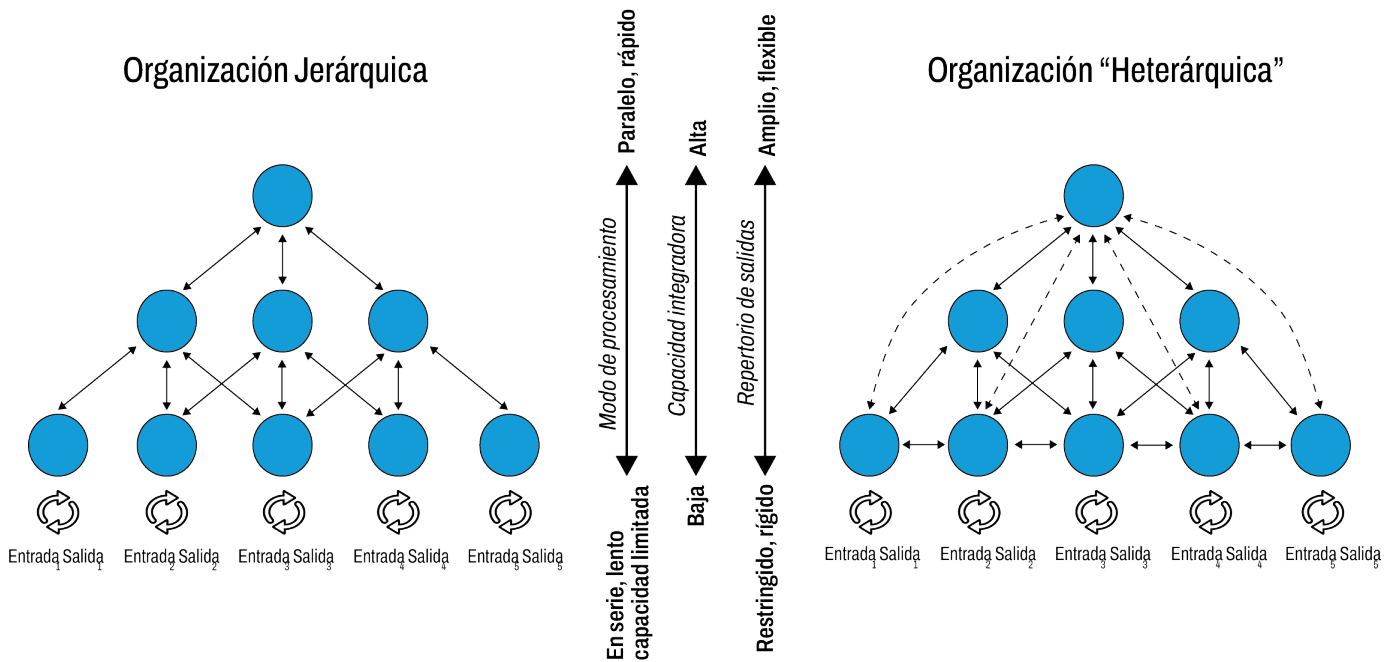


Figura 1: Ilustración de diferentes estructuras organizativas para un SdS (Adaptada de [8]).

Las pruebas efectivas de un SdS dependen de una combinación de metodologías adaptadas para abordar esta complejidad [9, 10, 11].

5.1. Modelado y simulación

La modelización y la simulación constituyen un enfoque fundamental en pruebas de SdS, ya que permiten a las partes interesadas evaluar las interacciones, el rendimiento y los comportamientos emergentes antes de realizar pruebas físicas. Las simulaciones ofrecen una forma rentable de ensayar diversos escenarios, condiciones de estrés y posibles fallos del sistema. A partir de un análisis exhaustivo de investigación, los gemelos digitales son definidos como "una representación virtual de un sistema físico (y su entorno y procesos asociados), que se actualiza mediante el intercambio de información entre los sistemas físico y virtual" (véase la Figura 2) [12]. Los gemelos digitales y las emulaciones de sistemas se utilizan cada vez más para analizar el rendimiento y predecir el comportamiento del SdS bajo distintas condiciones operativas.

5.2. Pruebas incrementales y agile

Dada la naturaleza evolutiva de los SdS, se emplean con frecuencia metodologías de prueba incrementales y *agile*, cuya persistencia a lo largo del ciclo de vida es crítica para mantener un estado "perenne" y, por tanto, resiliente frente a comportamientos emergentes [10,11]. Estas metodologías se centran en evaluaciones iterativas que permiten ajustes dinámicos basados en retroalimentación en tiempo real [3], con sincronización de los cambios generacionales dentro de un "ritmo de batalla" orientado a la resiliencia. Las estrategias de prueba *agile* garantizan que las modificaciones en sistemas individuales no comprometan la funcionalidad global del SdS, especialmente en entornos donde los sistemas constituyentes son desarrollados por contratistas o agencias diferentes, que introducen cambios en distintos momentos y para distintos perfiles de cliente.

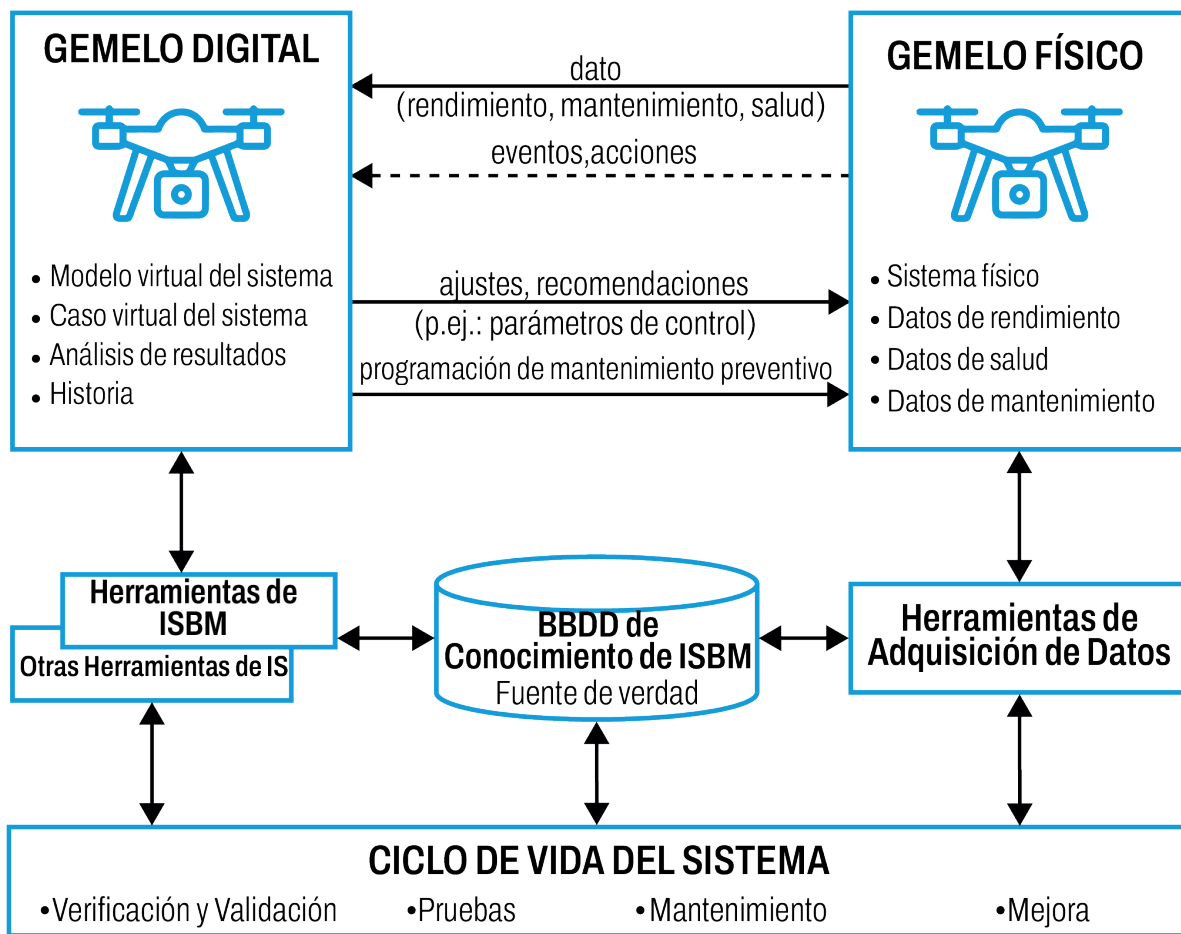


Figura 2: Ilustración de un gemelo digital (Adaptada de [13]).

5.3. Pruebas de campo y operacionales

Las pruebas de campo en un entorno operativo son fundamentales para validar el rendimiento real de un SdS. Las pruebas de campo, los ejercicios conjuntos y las simulaciones tipo “juego de guerra” proporcionan información crítica sobre la interoperabilidad, la resiliencia y la efectividad de la misión. No obstante, este tipo de pruebas implica un alto consumo de recursos y suele estar limitado por consideraciones logísticas y de seguridad [4].

5.4. Pruebas extremo a extremo y de interoperabilidad

Garantizar la interoperabilidad fluida entre los sistemas constituyentes representa uno de los principales desafíos en T&E de SdS. Es necesario realizar pruebas extremo a extremo en el nivel más alto de interoperabilidad (véase Fig. 1) para evaluar la eficacia de la comunicación, el intercambio de datos y el logro de los objetivos de misión entre sistemas heterogéneos, así como para gestionar los comportamientos emergentes a lo largo del ciclo de vida. Estas pruebas suelen implicar la monitorización en tiempo real de interfaces, estándares de intercambio de datos y compatibilidad entre plataformas [2]. Este aspecto del SdS resulta especialmente complejo para el gestor de capacidad, quien opera sistemas interconectados de múltiples proveedores con el objetivo de generar efectos integrados, pero que aún no ha invertido en una representación soberana de banco de pruebas de los sistemas conectados, es decir, carece de un centro de integración de redes (NIC, por sus siglas en inglés) o de una simulación En Vivo-Virtual-Constructiva (LVC, por sus siglas en inglés) en el nivel más alto de efectividad (véase Fig. 2) [14, 9].

5.5. Pruebas de ciberseguridad y resiliencia

Dada la naturaleza interconectada de los SdS, las vulnerabilidades de ciberseguridad pueden generar efectos en cascada a lo largo de todo el sistema. Las pruebas de ciber-resiliencia implican técnicas como las pruebas de penetración, el modelado de amenazas y las evaluaciones de resiliencia, con el objetivo de mitigar posibles ciberamenazas. Esto requiere un análisis exhaustivo e independiente de las fases en cadena de un ciberataque (*Cyber Kill Chain*) para identificar los vectores de ataque (misión, ataque, variante) a través de toda la superficie de ataque del SdS.

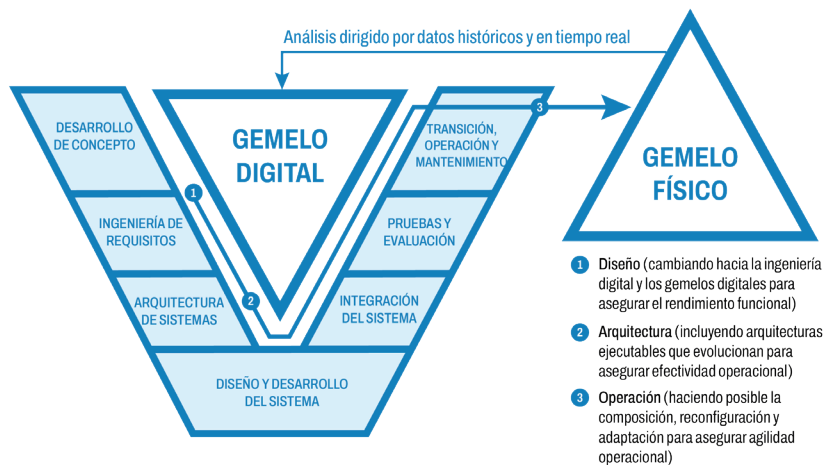


Figura 3: Ilustración del importante uso a lo largo del ciclo vida de un gemelo digital como base para T&E de SdS perenne (Adaptado de [16]).

5.6. Gemelos digitales

Los gemelos digitales son particularmente eficaces para T&E de SdS cuando se desarrollan progresivamente junto con la capacidad y se utilizan a lo largo del ciclo de vida como base para la T&E del SdS, como se muestra en la Figura 3 [16]. Sin embargo, el modelo en diamante Modelo-Virtual-Diseño-Físico (Figura 4) [17] podría ser más apropiado, aunque se reutilice o revise a lo largo del ciclo de vida (p.ej., de forma helicoidal).

En resumen, T&E de SdS requerirá un cambio de paradigma en aquellos entornos donde el gestor de capacidad y los contratistas de apoyo aún no hayan adoptado:

- Supervisión, desarrollo, pruebas e integración continua (que pueden habilitarse mediante la Ingeniería de Sistemas Basada en Modelos (ISBM) [11], gemelo digital [12] e ingeniería digital [18]).
- Pruebas altamente automatizadas.
- Contratación de soporte basada en la tasa de cambio [19, 20, 21, 22, 23].
- Centro de Integración de Redes en el nivel más alto de integración [24].
- Simulación en vivo, virtual y constructiva (LVC) para la interoperabilidad entre sistemas de diferentes generaciones.

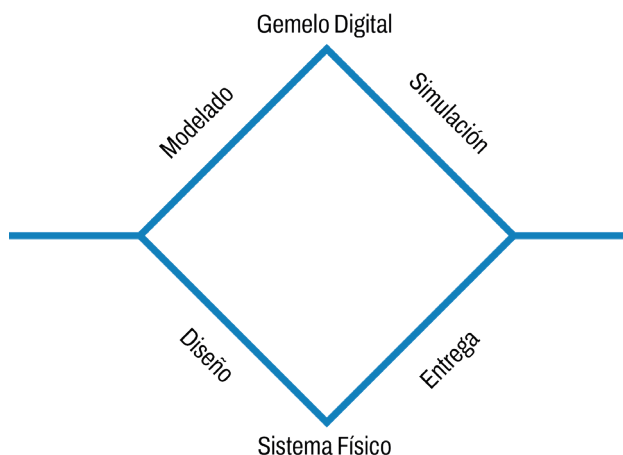


Figura 4: Desarrollo en paralelo de sistemas virtuales y físicos (Adaptado de [17]).

En términos sencillos, los SdS son sistemas complejos cuya gestión adecuada requiere establecer los medios y la cultura necesarios para explorar, detectar y responder a lo largo del ciclo de vida (p.ej., marco Cynefin [25]).


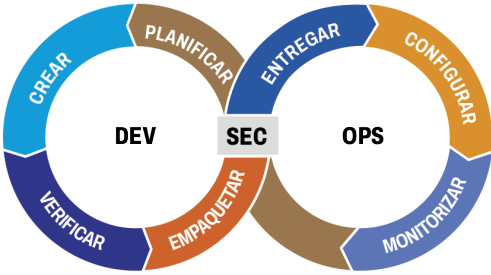
Práctica observada (“No-hagas”)	Estado deseado (“Haz”)	Obstáculo
 <p data-bbox="150 491 625 521">Defense Acquisition University, junio 2010</p>	 <p data-bbox="675 471 1153 516"> https://commons.wikimedia.org/wiki/File:Devops-toolchain.svg (modifications licensed CC-BY-SA) </p>	<p data-bbox="1230 154 1445 369"> 10 U.S.C. §2334 10 U.S.C. §2399 10 U.S.C. §2430 10 U.S.C. §2433a 10 U.S.C. §2460 10 U.S.C. §2464 </p> <p data-bbox="1254 413 1422 521"> DODI 5000.02 par 5.c.(2) y 5.c.(3)(c)-(d) </p>

Figura 5: Ilustración de la necesidad de integrar la complejidad en los enfoques de gestión [26].

6. PRINCIPIOS RECTORES DE T&E EN SDS

6.1. Adopción de desarrollo, seguridad y aseguramiento continuos

Uno de los conceptos fundamentales para abordar T&E en SdS consiste en integrar aquellos elementos caracterizados por una alta complejidad (véase Figura 5).

Los conceptos de DevOps and DevSecOps (“Desarrollo y Operaciones” y “Desarrollo, Seguridad y Operaciones”, respectivamente, por sus siglas en inglés) [27] son fundamentales para T&E en SdS:

“En el ámbito empresarial, la demanda de agilidad y velocidad continúa en aumento. Los avances tecnológicos, como la Ingeniería Continua —especialmente DevOps—, han permitido que algunas organizaciones obtengan ventajas competitivas. Sin embargo, las preocupaciones en materia de seguridad han crecido debido a incidentes, como filtraciones masivas de datos, lo que ha obligado a las organizaciones de todo el mundo a prestar una atención significativa a las amenazas de seguridad. Esto es particularmente relevante en el contexto de sistemas críticos para la seguridad, dada la posible gravedad de los incidentes, como la pérdida de vidas humanas, la exposición o uso indebido de información sensible y pérdidas financieras considerables. En este escenario, se requiere una integración sólida de la seguridad dentro de los enfoques DevOps. Así, la necesidad de incorporar la seguridad en DevOps —como DevSecOps— fue mencionada por primera vez en 2012.”

DevSecOps puede, en teoría, conducir a una Autorización Continua para Operar (CATO, por sus siglas en inglés) [28, 10, 29], siempre que exista un nivel suficiente de pruebas automatizadas que garanticen la confianza en las siguientes categorías:

- Funcionalidad/Usó (con expectativa de cambio).
- Abuso (p.ej., seguridad).
- Uso indebido/Malicioso (p.ej., ciberamenazas en constante evolución) [30].
- Todos los aspectos no funcionales (en la medida de lo posible, mediante pruebas de tipo *fuzzing*, esto es, pruebas con introducción de datos aleatorios o maliciosos) [31, 24].

A medida que se añaden casos de prueba desde la primera hasta la última categoría, aumentan las permutaciones posibles y, con ello, la necesidad de mayor capacidad computacional u otros recursos de prueba. Las pruebas combinatorias contribuyen mediante la cobertura de factores y funciones críticas (multifactorial) [32, 33]. Por ello, algunas herramientas de prueba automatizadas ya incorporan pruebas combinatorias, aunque para aplicarlas correctamente se requieren competencias adicionales por parte de los profesionales en el diseño de pruebas combinatorias. El proceso de diseño de pruebas combinatorias se ilustra en la Figura 6.

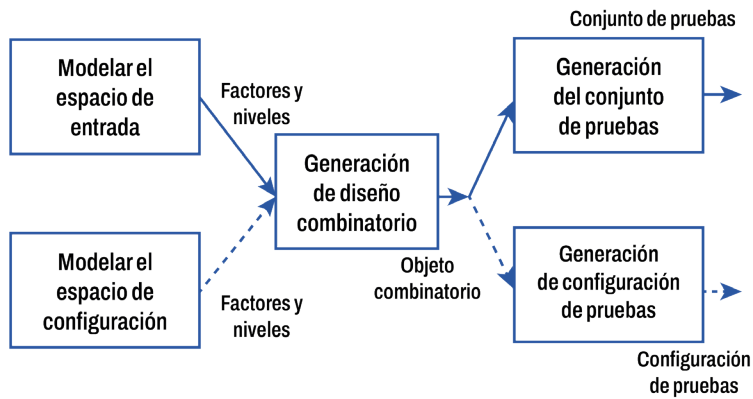


Figura 6: Ilustración del proceso de diseño de pruebas combinatorias.

Desarrollar la infraestructura de pruebas necesaria para implementar una Autorización Continua para Operar (CATO) representa un desafío. Un ejemplo de ello es la metodología de 10 pasos para un enfoque arquitectónico en capas basado en Simulación, Experimentación, Analítica y Pruebas (SEAT, por sus siglas en inglés) [34]. Este tipo de enfoques de prueba resulta prometedor, al igual que las mejoras en las redes de prueba, tanto del ámbito corporativo como de defensa, que permiten realizar pruebas de tipo *fuzzing* [24], así como metodologías de prueba orientadas a la evaluación de APIs [35].

necesidad justifica el desarrollo de un entorno virtual que proporcione una verdad de referencia (ground truth) en un escenario simulado, que se conecte con datos del mundo real y que utilice diversos sistemas de simulación tanto específicos de dominio como agnósticos, para el desarrollo, prueba y evaluación. ... Esta arquitectura virtual y constructiva de SdS debe ser independiente de la infraestructura computacional subyacente, pero debe estar habilitada para la nube a fin de facilitar una integración más amplia de componentes de software con capacidades de IA."

"Los sistemas autónomos y habilitados con IA presentan un reto para su integración dentro del paradigma de Sistemas de Sistemas (SdS). Es necesario contar con un banco de pruebas completo de SdS para verificar la integridad de un sistema dado y preservar la modularidad y la trazabilidad de sus sistemas constituyentes. Este sistema integrado debe permitir pruebas y desarrollo iterativos y continuos. Esta

La prueba de tipo *fuzzing* sobre todas las permutaciones, combinando enfoques aleatorios y sistemáticos, parece ser la más eficaz, especialmente para funciones críticas intensivas en hardware y software, como un bus de datos aeronáutico, una red LAN vehicular, un sistema IPMS naval o la señalización ferroviaria. La Figura 7 muestra una ilustración del uso de pruebas de tipo *fuzzing* mediante un algoritmo de Bandido Multibrazo (*Multi-Armed Bandit*).

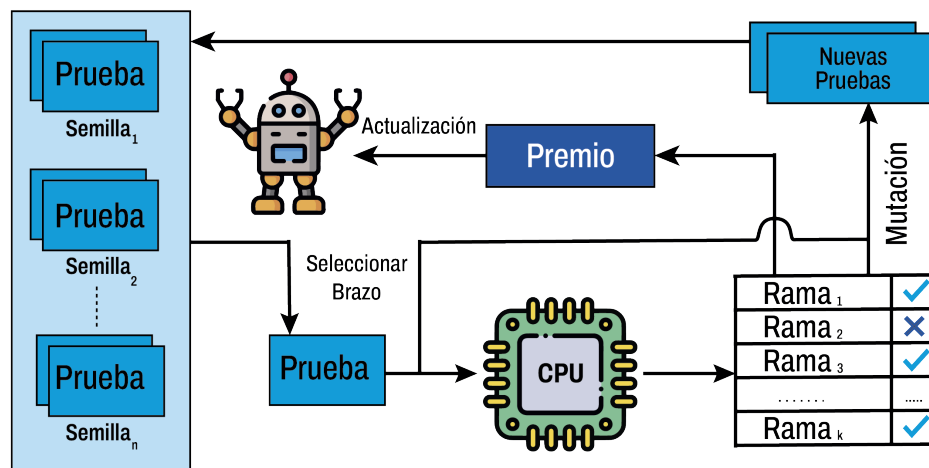


Figura 7: Ilustración de un algoritmo de Bandido Multibrazo (*Multi-Armed Bandit*) para un enfoque de pruebas tipo *fuzzing* [37].

6.2. Integración de la digitalización y la transformación digital en pruebas

El segundo enfoque principal de T&E para SdS proviene también del ámbito de la digitalización: la Ingeniería de Sistemas Basada en Modelos (MBSE, por sus siglas en inglés). MBSE captura el contexto de diseño, los requisitos, las métricas de prueba, el diseño de sistemas y los casos de prueba a medida que se desarrolla un SdS, introduciendo la información una sola vez para reutilizarla múltiples veces. Esto permite una documentación sustancialmente reducida, aprobaciones más generalizadas y actualizadas y la agilidad necesaria para mantener el sistema siempre vigente (es decir, como fuente autorizada de la verdad) [13, 37]. Cuando se implementa correctamente, incluyendo la capa de análisis operacional, MBSE permite vincular la ingeniería de escenarios o misiones de extremo a extremo con los casos de prueba y facilita la gestión de cambios derivados de comportamientos emergentes, mediante pruebas de regresión más focalizadas.

El uso de MBSE a lo largo del ciclo de vida contribuye de forma fundamental a mantener los procesos T&E siempre actualizados, y facilita el diseño de pruebas impulsado por modelos [38]. Construir un modelo MBSE que cubra todos los elementos relevantes del SdS representa un desafío [39], como se muestra en la Figura 8. En esencia, un MBSE para SdS debe lograr que cada sistema propietario se integre en un entorno común de referencia de modelado, que sea capaz de ejecutar pruebas [40], o que permita transformaciones entre distintos sistemas. Dado que algunos sistemas fueron diseñados antes de la era digital, probablemente será necesario realizar trabajo retrospectivo en MBSE para la automatización de las pruebas.

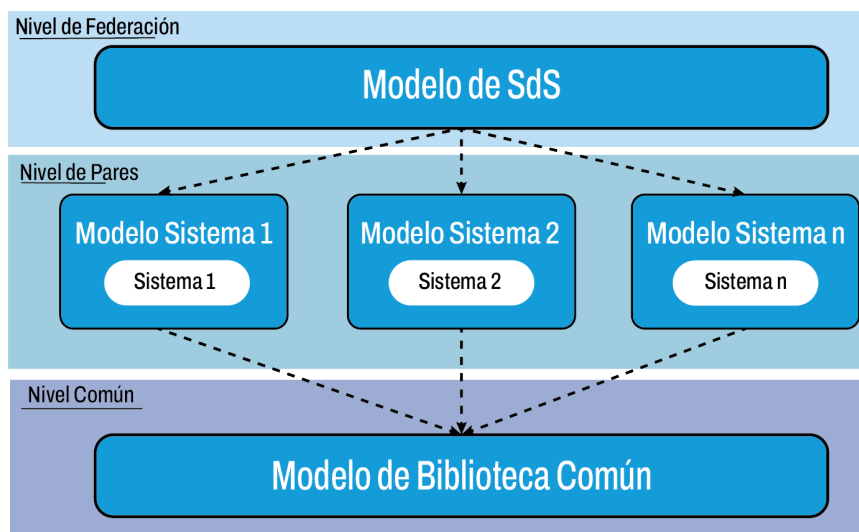
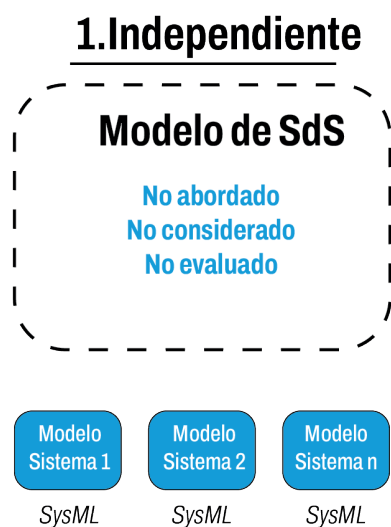


Figura 8: Ilustración de enfoques MBS para SdS (Adaptada de [41]).

7. CONCLUSIONES

Los T&E de SdS requieren un enfoque diferente al de las pruebas de sistemas debido a las diferencias entre ambos y a las propiedades de los SdS. Estas propiedades requieren que el enfoque adoptado para T&E considere los impactos de las líneas de referencia de los sistemas constituyentes, la gobernanza y los recursos para la integración y prueba de SdS y las dificultades de probar un programa geográficamente distribuido que probablemente exhiba comportamientos emergentes. Esto requiere prestar mucha atención al modelo de gobernanza empleado y anclar las métricas en la efectividad y rendimiento de la misión.

Una consideración clave para T&E de SdS que debe tenerse en cuenta es que la idea de que el SdS pueda ser completamente probado antes del despliegue no es realista.

Puede ser más apropiado aproximarse a T&E de SdS con un enfoque basado en la capacidad para abordar el riesgo. El equipo de ingeniería de sistemas de SdS identifica los problemas críticos para el éxito de cada época del desarrollo de SdS, así como los lugares donde los cambios en la evolución podrían afectar negativamente las misiones de los usuarios y luego se enfoca en ellos durante las pruebas previas al despliegue.

Retrasar las actualizaciones del sistema hasta que todos los sistemas constituyentes en una época estén listos para probar con éxito es poco práctico e inviable en la mayoría de los casos. Dado que la mayoría de los SdS están compuestos por sistemas ya desplegados, puede que no haya una decisión de despliegue única.

Las pruebas a nivel de SdS pueden ser costosas y puede ser muy difícil crear entornos de prueba que representen de manera realista los resultados esperados en un entorno operativo debido al tamaño y la complejidad de muchos entornos de SdS.

REFERENCIAS

1. Dahmann, J. and D. DeLaurenits (2023). Unique Challenges in System of Systems Analysis, Architecting, and Engineering. *Systems Engineering for the Digital Age: Practitioner Perspectives*. D. Verma, John Wiley & Sons, Inc.
2. Dahmann, J., Lane, J., & Baldwin, K. (2011). An integrated approach to system of systems effectiveness analysis. *The Journal of Defense Modeling and Simulation*, 8(1), 3-16.
3. Department of Defense (DoD). (2020). *Department of Defense Test and Evaluation Management Guide*. Office of the Director, Operational Test and Evaluation.
4. Rebovich, G., & DeLaurentis, D. (2011). *Systems of Systems Engineering: Principles and Applications*. CRC Press.
5. Vaneman, W. K. (2016). The system of Systems Engineering and Integration “vee” model. 2016 Annual IEEE Systems Conference (SysCon). <https://doi.org/10.1109/syscon.2016.7490599>
6. Dahmann, J., Lane, J. A., Rebovich, G., & Lowry, R. (2010). Systems of systems test and evaluation challenges. 2010 5th International Conference on System of Systems Engineering, 1–6. <https://doi.org/10.1109/sysose.2010.5543979>
7. McDermott, T., et al. (2023). Concepts of Trust and Resilience in Cyber-Physical Systems. *Systems Engineering for the Digital Age: Practitioner Perspectives*. D. Verma, John Wiley & Sons: 473-488.
8. Norman, G. J., et al. (2011). “Current Emotion Research in Psychophysiology: The Neurobiology of Evaluative Bivalence.” *Emotion Review* 3(3): 349-359.
9. Joiner, K. and M. Tutty (2018). “A tale of two allied defence departments: new assurance initiatives for managing increasing system complexity, interconnectedness and vulnerability.” *Australian Journal of Multi-Disciplinary Engineering* 14(1): 4-25.
10. Weiss, J. and D. Patt (2022). *Software Defines Tactics: Structuring Military Software Acquisitions for Adaptability and Advantage in a Competitive Era*. Online, Hudson Institute.
11. Call, D. R. and D. R. Herber (2023). *A Case for Model-Based Systems Engineering in an Agile World & Principles for Growth*. INCOSE International Symposium, INCOSE.
12. VanDerHorn, E. and S. Mahadevan (2021). “Digital Twin: Generalization, characterization and implementation.” *Decision Support Systems* 145: 113524.
13. Madni, A. M., et al. (2023). Exploiting Digital Twins in MBSE to Enhance System Modeling and Life Cycle Coverage. *Handbook of Model-Based Systems Engineering*. C. C. Madni. Cham Springer International Publishing: 527-548.
14. Joiner, K., et al. (2018). Modelling the Efficacy of Assurance Strategies for Better Integration, Interoperability and Information Assurance in Family-of-System-of-Systems Portfolios. *International Conference on Complex Systems Design & Management*, Springer.
15. Joiner, K. F., et al. (2018). “Four testing types core to informed ICT governance for cyber-resilient systems.” *International Journal of Advances in Security* 11.
16. Tolk, A., Barry, P. and Doskey, S.C. (2022). Using modeling and simulation and artificial intelligence to improve complex adaptive systems engineering. *International Journal of Modeling, Simulation, and Scientific Computing*, 13 (2) 2241004: 1-19.
17. Richardson (2020). *Model Based Systems Engineering: A stepping stone on the path to Digital Engineering*. GSFC Systems Engineering Seminar
18. Mukhopadhyay, A., et al. (2023). *A Perspective on the Adoption of Digital Engineering Within an Enterprise*. Online, Ansys.
19. Cordero, S., et al. (2020). Addressing Obsolescence from day one in the conceptual phase of complex systems as a design constraint. *Product Lifecycle Management Enabling Smart X: 17th IFIP WG 5.1 International Conference*, Rapperswil, Switzerland, Springer International Publishing.
20. Morgan, M., et al. (2021). “Synergizing model-based systems engineering, modularity, and software container concepts to manage obsolescence.” *Systems Engineering* 24(5): 369-380.
21. Oliver, E., et al. (2022). “A resilience systemic model for assessing critical supply chain disruptions.” *Systems Engineering*(5): 510-533.
22. Donelli, G., et al. (2023). “Concurrent Value-Driven Decision-Making Process for the Aircraft, Supply Chain and Manufacturing Systems Design.” *Systems* 11(12): 578.
23. Herburger, M., et al. (2024). “Building supply chain resilience to cyber risks: a dynamic capabilities perspective.” *Supply Chain Management: An International Journal* 29(7): 28-50.
24. Joiner, K. F. (2024). “Review of Fuzz Testing to find System Vulnerabilities.” *ITEA Journal of Test and Evaluation* 45(4).

25. Daniel, P. A. and C. Daniel (2018). "Complexity, uncertainty and mental models: from a paradigm of regulation to a paradigm of emergence in project management." *International Journal of Project Management* 36: 184–197.
26. McQuade, M., et al. (2018). Defense Innovation Board Do's and Don'ts for Software (draft). D. I. Board. Online, U.S. DoD.
27. Sánchez-Gordón, M. and R. Colomo-Palacios (2020). Security as Culture: A Systematic Literature Review of DevSecOps. *IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW, ACM*.
28. Sanders, G., et al. (2021). Integrating ZeroTrust and DEVSECOPS. Online, Carnegie Mellon University.
29. Olena, J. (2023). "Software Standardization and Infrastructure Development Efforts in Support of Unmanned Maritime Vehicle Autonomy." *Naval Engineers Journal* 135(1): 41-45.
30. Fowler, S., et al. (2025: submitted). "Assessing Cyberworthiness of Complex System Capabilities using the Cyber Evaluation and Management Toolkit (CEMT)." *Computers and Security*.
31. Brumley, A. (2022). Introduction to Fuzzing Software. ITEA Online (<https://itea.memberclicks.net/member-landing-page>), International T&E Association: 29.
32. Kuhn, D. R., et al. (2016). Combinatorial Testing for Cybersecurity and Reliability. *Information Technology Bulletin, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce*.
33. Lanus, E., et al. (2021). Combinatorial testing metrics for machine learning. *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, Online, IEEE.
34. Mittal, S., et al. (2023). "Providing a User Extensible Service-Enabled Multi-Fidelity Hybrid Cloud-Deployable SoS Test and Evaluation (T&E) Infrastructure: Application of Modeling and Simulation (M&S) as a Service (MSaaS)." *Information* 14(10): 528.
35. Gomez, A. and A. Vesey (2024). On the Design, Development, and Testing of Modern APIs. Online, Software Engineering Institute.
36. Gohil, V., et al. (2024). MABFuzz: Multi-armed bandit algorithms for fuzzing processors. *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE.
37. Verma, D., Ed. (2023). *Systems Engineering for the Digital Age: Practitioner Perspectives.*, John Wiley & Sons.
38. Alvarado, J. L. J. and T. H. Bradley (2024). "A Case Study-based Assessment of a Model-driven Testing Methodology for Applicability and Metrics of Model Reuse." *ITEA Journal* 45(4).
39. DeLaurentis, D., et al. MBSE for System-of-Systems. *Handbook of Model-Based Systems Engineering*. C. C. Madni. Cham, Springer International Publishing: 987–1015.
40. Martell, J. A., et al. "Development of a Digital Engineering Testing Framework for CUBESAT Applications." *ITEA Journal* 45(3).
41. Swickline, C., et al. (2024). "A methodology for developing SoS architectures using SysML model federation." *Systems Engineering* 27(2): 368-385.
42. Gorod, A., et al. (2019). *Evolving Toolbox for Complex Project Management*. Boca Raton, Florida, Auerbach Publications.
43. Dahmann, J., & Baldwin, K. (2008). Understanding the current state of US defense systems of systems and the implications for systems engineering. *18th Annual International Symposium of the International Council on Systems Engineering (INCOSE)*.
44. Ferreira, F. H., et al. (2021). Reliability in Software-intensive Systems: Challenges, Solutions, and Future Perspectives. *47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE.
45. Nachbagauer, A. (2021). "Managing complexity in projects: Extending the Cynefin framework." *Project Leadership and Society* 2.

BIOGRAFÍAS

MARK PHILLIPS

Mark Phillips es Director de Estrategia de Programas de Adquisiciones de Defensa en Intel Corporation Government Technologies. Anteriormente, fue Fellow Técnico y Jefe de Ingeniería en Raytheon Technologies, centrado en tecnología de misiles y Arquitecto Principal para Lockheed Martin, dedicando los últimos dos años al National Cyber Range.



Mark tiene más de 45 años de experiencia en tecnologías de defensa y gubernamentales, abarcando múltiples países y especializándose en interoperabilidad entre sistemas. Mark sirvió 20 años en el Ejército Australiano en una variedad de roles y luego pasó a la Reserva de la Real Fuerza Aérea Australiana para colaborar en la coordinación entre Estados Unidos y Australia en áreas de entrenamiento y prueba. Mark está graduado por el Royal Military College de Australia. Mark tiene una Licenciatura en Ingeniería Eléctrica (cum laude) de la University of New South Wales, un Máster en Ingeniería en Modelado y Simulación de la Old Dominion University y es candidato a doctorado en Ingeniería de Sistemas en la Old Dominion University.

DR. KEITH JOINER

El Dr. Keith Joiner, CSC, fue ingeniero aeronáutico de la Fuerza Aérea, gerente de proyectos y profesor durante 30 años antes de unirse a la Universidad de Nueva Gales del Sur como docente e investigador en pruebas y evaluación. Actualmente es el Editor Jefe de la revista Journal of Testing de la International Test and Evaluation Association (ITEA) y



también enseña diseño de aeronaves. Como Director General de Pruebas y Evaluación de Defensa en todos los dominios de servicio, recibió la Conspicuous Service Cross de Australia, y por realizar planes de reducción para la Fuerza Multinacional en Irak, recibió la Meritorious Service Medal de EE. UU. Es Ingeniero Profesional Certificado y Director de Proyectos Certificado, con más de 100 artículos de investigación publicados sobre la fiabilidad de sistemas de ingeniería.

Su diversa investigación incluye:

- 1) Asegurar la ciberseguridad, sistemas habilitados por IA y sistemas autónomos robóticos.
- 2) Utilizar diseño de pruebas de alto rendimiento y gobernanza de sistemas complejos.
- 3) Desarrollar vehículos híbridos aire-mar y la electrificación de aeronaves.

AURELIO FERNÁNDEZ SÁEZ

Aurelio Fernández Sáez es actualmente el Gerente de Proyectos Especiales de Modernización Tecnológica en Isdefe, donde lidera iniciativas TIC para la Administración General del Estado, particularmente en las áreas de Justicia, Empleo, Tráfico y Seguridad Social. Anteriormente, fue Jefe del Área de Calidad y Pruebas



de Sistemas, coordinando proyectos estratégicos para el Ministerio de Justicia y el Servicio Público de Empleo Estatal. Con más de 30 años de experiencia en Ingeniería de Sistemas y Transformación Digital, su perfil combina visión estratégica, liderazgo técnico y compromiso institucional. Es ingeniero aeronáutico por la Universidad Politécnica de Madrid y posee certificaciones en PMP, Lean Six Sigma y TOGAF. Es miembro de INCOSE. Ha contribuido a la estandarización de procesos y la gestión de equipos multidisciplinares. Complementa su actividad profesional con la docencia como profesor invitado en instituciones como la Universidad Politécnica de Madrid, la Universidad Autónoma de Barcelona, la Universidad de Málaga, la Universidad de Las Palmas de Gran Canaria y los programas de formación interna de Isdefe.

MANUEL FERNÁNDEZ ASTABURUAGA

Manuel Fernández Astaburuaga es ingeniero aeronáutico por la Universidad Politécnica de Madrid. Está certificado en Prince2, ISTQB, COBIT y CSEP por INCOSE. Tiene 25 años de experiencia en Ingeniería de Sistemas, 3 de ellos en Investigación y Desarrollo en diseño de aviones, en la Universidad de Delft, y los siguientes como consultor de



procesos y pruebas en Isdefe; a este respecto, dirigió el laboratorio de pruebas de software de Aena durante 14 años. Como especialista en pruebas de software, fue miembro del Consejo Español de Certificación de Pruebas Software y del Grupo de Trabajo Español WP 26 dentro del comité SC7 de AENOR AEN / CTN 71 “Tecnologías de la Información” para la definición de estándares de pruebas de software ISO-IEC 29119. Actualmente lidera una PMO para la Dirección de TI en la Dirección General de Tráfico (DGT) y es formador interno en Isdefe sobre Ingeniería de Sistemas.



AUTO DRIVING



100m

60 km/h



Del ciclo de vida a la evolución de los Sistemas de Sistemas

Tom McDermott, *Stevens Institute of Technology* (tmcdermo@stevens.edu)
Miguel Ángel Coll, *Isdefe* (macoll@isdefe.es)

Resumen

Los *Sistemas de Sistemas* (SdS) no tienen ciclos de vida rígidos, sino que evolucionan constantemente a medida que algunos elementos constituyentes se desvanecen y otros nuevos se incorporan. Este capítulo analiza cómo diseñar la arquitectura y cómo gestionar la evolución, tanto de los SdS como de sus elementos constituyentes. La evolución se ve impulsada, internamente, por la evolución de las tecnologías, los procesos, los productos y las herramientas; y externamente, por la evolución de otros factores clave como la competencia de mercado, la economía y las preferencias de los usuarios. Esta evolución debe gestionarse como un proceso intencionado, pero también como un proceso oportunista. Debe de ser liderada. Este capítulo analiza las estrategias para gestionar la evolución de los SdS basadas tanto en los principios de gestión de los SdS como en los principios de los Sistemas de Innovación¹. El capítulo considera la evolución de los SdS como un desafío para el liderazgo empresarial y culmina con las cualidades que deben poseer los líderes de los SdS.

Palabras clave

Sistemas de Sistemas, Gestión de Sistemas de Sistemas, Liderazgo de Sistemas de Sistemas, Sistemas de Innovación.

¹. "Sistemas de Innovación" se refiere a redes organizadas de elementos (personas, empresas, universidades, laboratorios, gobiernos, infraestructura tecnológica, etc.) que interactúan para desarrollar, difundir y aplicar innovaciones tecnológicas o de procesos.



1. INTRODUCCIÓN: ACERCA DE LA EVOLUCIÓN DE LOS SISTEMAS DE SISTEMAS

Por definición, los Sistemas de Sistemas (SdS) se componen de Sistemas Constituyentes (SC) que son independientes desde el punto de vista operativo y de gestión [1]. Si bien esto es cierto, la mayoría de los SdS modernos también dependen de otros para obtener determinados conjuntos de capacidades, lo que difumina los límites de los SdS y complica su gestión. Con respecto a la evolución de los SdS, los cambios en los SdS siempre ocurrirán a nivel de los SC [2], lo que hace que el cambio en los SdS sea más complicado de gestionar que en los sistemas tradicionales. Los SC individuales también pueden cambiar independientemente de los SdS, creando efectos imprevistos a nivel de SdS. Dado que los SdS proporcionan capacidades que emergen más allá de las de cualquier SC individual, un SdS podría tener entradas y salidas únicas más allá de las proporcionadas por los SC individuales en sus usos independientes [3]. En la evolución de los SdS, se debe asegurar que estas capacidades únicas, así como las entradas y salidas relacionadas, se alcanzan, preservan y gestionan adecuadamente a nivel de SdS. Como resultado, la evolución de un SdS debe coordinarse a nivel de SdS y gestionarse a través de los SC y, quizás, de otros SdS.

Los SdS son cada vez más complejos. En las empresas comerciales, la mayoría de los productos actuales se desarrollan y suministran con conectividad a otros sistemas mediante redes comerciales y aplicaciones de software. La disponibilidad de información aumenta la autonomía e inteligencia de muchos de estos sistemas; incluso los productos más sencillos presentan hoy en día aspectos propios de los SdS. Las empresas de defensa, por su parte, están aumentando la conectividad en el campo de batalla y otros con activos entre sí, a la vez que incrementan la autonomía de cada SC individual. Las organizaciones relacionadas con la adquisición de sistemas militares también están fomentando la innovación mediante entidades y tecnologías comerciales, lo que reduce su influencia a nivel de los SC. Hoy en día, los SdS están poniendo énfasis en la rapidez de su despliegue, utilizando para ello sistemas que no son de desarrollo (hechos a medida), lo que también se traduce a una menor influencia a nivel de los SC. Estas tendencias conllevan una mayor independencia operativa y de gestión de los SC y una mayor complejidad para quienes deben liderar la evolución de los SdS. Desde 2019, cuando se publicó la norma ISO/IEC/IEEE 21840 [2], se han observado cambios significativos en los desafíos de la evolución de los SdS, entre los que se incluyen:

1. SdS compuestos por un mayor número de SC.
2. Mayor independencia operativa y de gestión de los SC.
3. Mayor conectividad entre los SC a través del SdS y entre SdS.
4. Mayor diversidad geográfica en los SC.
5. Mayor autonomía en los comportamientos de los SC y una autonomía emergente en los comportamientos de los SdS.
6. Incorporación más rápida de nuevos SC al SdS.
7. Mayor complejidad e incertidumbre en el diseño y la gestión de los SdS.

Como resultado, los atributos de liderazgo y de gestión necesarios para crear y mantener las capacidades de los SdS han cambiado. Este capítulo explora los enfoques para liderar y gestionar la evolución de estos, poniendo el énfasis en la innovación y el cambio sociotécnico. El capítulo comienza con un análisis de las características de los SdS que impulsan los comportamientos evolutivos. A continuación, se analizan los desafíos asociados al liderazgo y a la gestión de la evolución de los SdS. Posteriormente, se discuten los métodos para guiar la evolución de los SdS y se presentan las competencias de liderazgo necesarias para su gestión.

2. CARACTERÍSTICAS DE LA EVOLUCIÓN EN SISTEMAS DE SISTEMAS

2.1. Generalidades

La evolución de los SdS requiere de una serie de métodos para estructurar y gestionar el conjunto de SC interactuantes hacia un conjunto específico de objetivos o propósitos [4]. Un punto crítico, particularmente en la gestión de los SdS, es el desarrollo de métodos y herramientas que ayuden a guiar, predecir y gestionar los atributos y las capacidades emergentes [5]. La emergencia en los SdS es, a menudo, el resultado de una serie de innovaciones que hacen que se formen o se destaquen nuevas capacidades y atributos al nivel del SdS. Por lo tanto, una forma de ver la emergencia en un SdS es como una serie de innovaciones que, con el tiempo, alteran o transforman los SC o su uso operativo de una manera tal que se crean nuevas capacidades o atributos que son únicos del SdS. El principal desafío en la evolución de los SdS es influir en dichos cambios, que conduzcan hacia objetivos que den respuesta a las necesidades únicas de los grupos de las partes interesadas del SdS. Como tal, uno de los componentes en esta evolución es influir en esas partes interesadas, de tal manera que se obtengan las propiedades clave deseadas del SdS, lo que implica que la colaboración con las partes interesadas es necesaria a lo largo de todo el ciclo de vida de los SdS [6].

La mayoría de los SdS actuales pueden describirse como *Sistemas Sociotécnicos*¹. Los Sistemas Sociotécnicos son impulsados por las tecnologías y requieren de una importante participación humana y social, la cual a su vez, influye en la arquitectura y el diseño del sistema tecnológico [1]. A menudo, los SdS también son sistemas adaptativos complejos en los que, tanto la participación humana/social, como el sistema de ingeniería se adaptan mutuamente con el tiempo [7]. La característica distintiva de los SdS es que los comportamientos del “conjunto” provienen de sistemas constituyentes individuales que actúan de forma independiente y autónoma, y estos comportamientos no pueden reducirse, ni descomponerse, en los comportamientos de ninguno de los sistemas individuales [8].

Los sistemas que tienen interacciones sociales presentan varios patrones consistentes. Las siguientes características se han seleccionado como las más relevantes para el concepto de “evolución” [9]:

- **Autoorganización y jerarquía multiescala o multinivel:** los sistemas impulsados socialmente tienden a autoorganizarse en las capas inferiores y luego crean jerarquías a medida que crecen, generalmente impulsados por una serie de eventos. A menudo, la estructura de estas jerarquías también cambia con el tiempo. Los comportamientos que se desarrollan en las capas superiores no reflejan necesariamente los comportamientos de los agentes o grupos individuales. Incluso los *SdS dirigidos* presentan un cierto grado de autoorganización, a medida que los SC compiten por su inclusión en éstos.
- **Autonomía o interacción multi-agente:** los agentes individuales del sistema operan de forma autónoma, se adaptan y aprenden a medida que interactúan a lo largo del tiempo. Con el paso del tiempo, el grado de autonomía de los agentes individuales ha ido aumentando y seguirá aumentando, a un ritmo mayor. Los SdS se constituyen mediante acuerdos formales o informales entre agentes autónomos, lo que demanda una visión de los SdS como una empresa multi-agente.
- **Emergencia:** los comportamientos y propiedades emergentes surgen de las interacciones representativas de la totalidad de un SdS. La emergencia surge de la estructura de las partes y de su interacción, tal como se observan en un determinado contexto de interés. Estos comportamientos y propiedades surgen sin haber sido anticipados.

- **Desarrollo evolutivo:** las metas y objetivos, así como la estructura y la funcionalidad de un SdS, cambian constantemente a medida que se añaden, se modifican, o se eliminan entidades. Sin embargo, la evolución del conjunto es lenta en comparación con la de los agentes o componentes individuales [10] (aunque esta velocidad va en aumento). Además, las metas y los objetivos para el desarrollo de un nuevo SdS son muy diferentes a los del mantenimiento de uno ya existente. Los métodos para desarrollar las capacidades a nivel del SdS dependen, en gran medida, del contexto.
- **Conectividad:** la evolución del sistema está impulsada por la conectividad, la comunicación (flujo de información) y la colaboración. Esto aplica tanto al SdS como a las organizaciones que participan en él. Tanto la conectividad entre los SC como el flujo de información entre ellos están aumentando, al tiempo que los acuerdos de colaboración son cada vez más cruciales. La colaboración entre los SdS representa un desafío de liderazgo empresarial.
- **Complejidad:** estos sistemas son tan sumamente grandes, en términos del número de conexiones físicas, relaciones organizacionales e interacciones basadas en la información, que el comportamiento del sistema no puede analizarse completamente mediante descripciones convencionales (es decir, matemáticas) [11]. Por ese motivo, también son necesarias las herramientas y los enfoques cualitativos.

En la literatura sobre SdS, se plantea el concepto de *emergencia evolutiva*, o el estudio del proceso evolutivo que el sistema podría adoptar a lo largo del tiempo, y de cómo guiar eficazmente dicho proceso en presencia de otros atributos deseables de los sistemas (en contraposición con los indeseables). En este capítulo, consideramos esta evolución como un conjunto de *rutras de innovación*, donde determinados grupos de innovaciones se manifiestan con el tiempo al nivel del SdS, en función de su impacto disruptivo o transformador en los atributos y las capacidades del SdS.

2.2. Arquitecturas novedosas versus evolución incremental

Es importante considerar la evolución en los SdS como dos procesos separados: el surgimiento de un nuevo SdS, que a menudo resulta en una arquitectura “novedosa”, o sin precedentes, y el mantenimiento de un SdS ya existente, que requiere de unos fundamentos estables. La fase de surgimiento, a menudo, suele ser competitiva, donde múltiples SdS potenciales compiten por implementar nuevas tecnologías y modelos de negocio. La fase de surgimiento suele seguir el modelo de “*difusión de la innovación*” popularizado por Rogers, que mide la adopción por parte

1. El “Sistema Sociotécnico” se define como un sistema que integra de manera interdependiente elementos sociales (personas, organizaciones, procesos humanos) y elementos tecnológicos (tecnologías, infraestructuras, software, hardware) para lograr un propósito común.

de las partes interesadas [12]. Un SdS pasa a la fase de mantenimiento cuando ya ha sido adoptado por un número suficiente de usuarios como para lograr un modelo operativo estable.

Henderson y Clark [13] analizan dos formas de evolución relevantes para los SdS: la que modifica los componentes de los SC y la que modifica la estructura y las interacciones (la arquitectura del sistema), que también proporcionan los SC (generalmente definido como *infraestructura*). Los productos innovadores o radicalmente nuevos suelen utilizar nuevas arquitecturas para proporcionar nuevas capacidades. Los cambios verdaderamente transformadores suelen reflejarse en cambios en las arquitecturas. Sin embargo, las arquitecturas bien establecidas son difíciles de modificar debido al conocimiento implícito y a la inversión en infraestructura que conllevan. Por lo tanto, un componente adicional en el análisis de la evolución de los SdS debe separar las decisiones relacionadas con la arquitectura de las decisiones relacionadas con las capacidades. Con el tiempo, el conocimiento relacionado con la arquitectura se integra en la organización y los procedimientos de cada una de las empresas de los SC. En esas circunstancias, dado que dicho conocimiento es implícito, puede resultar difícil para un SC cambiar la arquitectura de sus productos. Por lo tanto, un SdS en su fase de mantenimiento tiende a presentar cambios incrementales hasta que otro SdS lo reemplaza.

Los SdS tienden a desarrollarse y a evolucionar en capas. La literatura sobre SdS generalmente las describe como:

- 1. Capa tecnológica:** los aspectos físicos del SdS, que incluyen hardware, redes y otros componentes de infraestructura física. Algunos de estos aspectos no son puramente tecnológicos; por ejemplo, los sistemas de gestión del tráfico aéreo dividen el espacio aéreo en zonas para mantener la seguridad en la arquitectura.
- 2. Capa de aplicaciones:** las capacidades del SdS, incluidas las aplicaciones de software que interactúan con los datos y la información entre los diferentes SC, para proporcionar las funcionalidades del SdS y la interacción con el usuario.
- 3. Capa de información:** los datos y la información que se intercambian entre los SC y otros SdS y que se proporcionan al SdS para que éste funcione eficazmente. Esta capa debe centrarse en la interoperabilidad.
- 4. Capa de negocio:** el conjunto de relaciones entre el SdS y los SC que responden a los factores externos cambiantes y a los modelos de negocio individuales del SdS y de sus SC.

Los SdS en fase de emergencia suelen estar impulsados por la innovación tecnológica en las tres primeras capas. Los SdS en la fase de mantenimiento suelen estar impulsados por las fuerzas del mercado y los modelos de negocio de la capa empresarial, así como por la necesidad de interoperabilidad continua entre los sistemas de control (utilizando estándares), que abarcan la capa tecnológica y la capa de información. Estas dinámicas son muy diferentes y requieren procesos muy distintos para gestionar su evolución. La literatura sobre estas dos fases (emergencia y mantenimiento) está fuertemente dividida entre el enfoque de la *Ingeniería de SdS* y el enfoque de los *Sistemas de Innovación*. El enfoque de la Ingeniería de SdS asume que podemos aplicar los principios de la Ingeniería de Sistemas y de la planificación de proyectos a la evolución de SdS, es decir, que dicha evolución puede ser “diseñada”. El enfoque de los Sistemas de Innovación, por su parte, se alinea más con los sistemas sociales y las teorías de la complejidad, y sugiere que sólo podemos “influir” en la evolución de los SdS, no planificarla explícitamente. En la práctica, se deben combinar ambos enfoques situacionalmente, en función de si las capacidades del SdS son emergentes o ya están en fase de mantenimiento.

2.3. Planificando la evolución de los Sistemas de Sistemas

El *Modelo de Ondas* (Figura 1) es un marco establecido para evaluar y planificar la evolución de los SdS. Es un marco “top-down”, derivado de los procesos de la Ingeniería de Sistemas. El *Modelo de Ondas* reconoce que la evolución está continuamente impulsada por la información del entorno externo (contexto) y, a diferencia de las perspectivas tradicionales de la Ingeniería de Sistemas, el análisis del cambio en el sistema es un proceso continuo, con múltiples incrementos superpuestos. El *Modelo de Ondas* considera la evolución como un proceso prospectivo con retroalimentación en cada iteración e intenta agrupar los cambios de los componentes en cambios en la arquitectura a nivel del SdS para generar eficiencia en el proceso de prueba y validación [14]. Los aspectos clave del *Modelo de Ondas* son la determinación de un *punto de partida* (en la figura, *Iniciar el SdS*), la realización de *Análisis al SdS*, el desarrollo y la evolución de la arquitectura del SdS y la planificación e implementación de las actualizaciones del SdS. El *punto de partida* de un SdS es particularmente difícil de definir, ya que el momento en el que se inicia implica que las cuatro capas de la arquitectura están lo suficientemente maduras como para permitir que se implementen y adopten nuevas capacidades (también llamado *modelo de difusión de la innovación*). El valor del *Modelo de Ondas* reside en el reconocimiento de que las actualizaciones de los SdS deben planificarse y de que existe una entidad que analiza las actualizaciones del SdS y las evoluciona a lo largo del tiempo. No obstante, hay que tener en cuenta que, salvo en los *SdS dirigidos*, tanto

las definiciones del *punto de partida* de los SdS como los análisis relacionados pueden ser bastante subjetivos.

Los principales resultados del Análisis del SdS (artefactos) incluyen: información de las capacidades del SdS (conceptos de operación y restricciones principales); información de los sistemas del SdS (vistas de la arquitectura de los sistemas, y las descripciones de los SC); información técnica del SdS (la jerarquía técnica: medidas de rendimiento y datos); líneas base arquitectónicas y técnicas (estándares, reglas de negocios, conexiones e interfaces); e información de la gestión del SdS (contratos y acuerdos). La evolución exitosa de un SdS depende, en gran medida, de lo bien que los líderes del SdS sean capaces de comunicar artefactos como estos y de lo bien que los comprendan las otras partes interesadas del SdS.

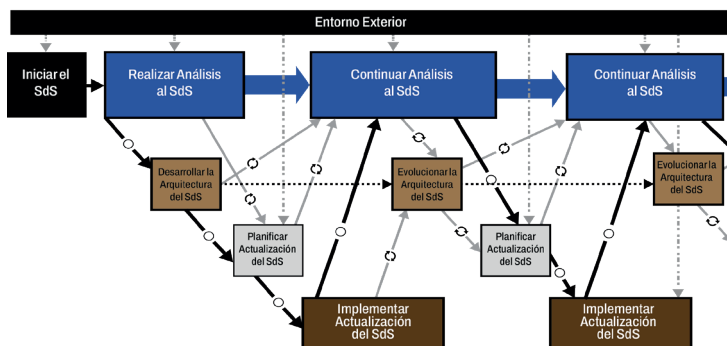


Figura 1. El Modelo de Ondas [14].

2.4. Los Sistemas de Sistemas como Sistemas de Innovación

La literatura sobre Sistemas de Innovación considera la emergencia de los SdS como una “transición tecnológica, que consiste en cambios relevantes en la configuración sociotécnica” [15]. El equivalente a la *Planificación* en el *Modelo de Ondas* de los SdS, en la literatura sobre innovación es la “*gestión de la transición*” [16]. Los modelos de Sistemas de Innovación reconocen la innovación como un proceso que abarca instituciones humanas y sociales, donde las innovaciones de nivel inferior, en los SC, forman nichos de adopción que, con el tiempo, producen cambios más amplios en el funcionamiento de los SdS ya establecidos, lo que finalmente resulta en la transformación del escenario existente. Hoy en día, podemos ver el desarrollo de las tecnologías civiles en vehículos sin conductor o en los modelos de *transformers* de la IA generativa, como una evolución “en proceso”. El aspecto principal de este modelo es que la innovación progresa a través de capas sociales y puede modelizarse como un fenómeno social multiescala o multicapa. En

la literatura sobre *gestión de la transición*, Geels [15] representa el *Modelo de Ondas* como un proceso de cambio “bottom-up” similar, pero mucho más desordenado, donde las arquitecturas de los SdS son regímenes sociotécnicos y las evoluciones progresan a través de muchos componentes de esos regímenes (tecnologías, políticas, infraestructuras, instituciones, industrias, etc.), habilitadas por un escenario abierto al cambio. La Figura 2 ilustra este proceso. Hay que considerar lo que sucede en cada momento como una instantánea en cada capa. Las innovaciones a nivel de los SC, que también introducen nuevas arquitecturas, son impulsores del cambio de régimen y deberían ser el foco de atención en la evolución de los SdS.

Mostafavi, Abraham, DeLaurentis y Sinfield propusieron un marco de *Innovación de Sistemas de Sistemas* (I-SdS) [17] en el que múltiples Sistemas de Innovación forman relaciones. Por ejemplo, la arquitectura de información, detección, computación y control del SdS que habilita los vehículos sin conductor comenzó como un desafío a nivel de SC (los *Grandes Desafíos* de DARPA) y surgió principalmente en laboratorios universitarios. En el SdS de vehículos sin conductor, el sistema de innovación que proporciona información de navegación (Sistema de Posicionamiento Global de la Fuerza Aérea de EE. UU., Google Maps, TomTom, etc.) tiene una relación de interdependencia con el sistema de innovación que desarrolla el control automatizado de los vehículos (los grandes desafíos de vehículos de DARPA, el Proyecto Chauffeur de Google – que, posteriormente, se convirtió en Waymo-, Tesla, etc.) y las empresas tecnológicas que proporcionan unidades de procesamiento gráfico (Nvidia) y LIDAR² de bajo coste (Luminar Technologies). Cabe destacar que existen múltiples innovadores, incluyendo universidades, agencias gubernamentales, grandes empresas y pequeños proveedores de tecnología, todos ellos involucrados en la migración de nichos tecnológicos a nuevos regímenes. Muchos SdS operan dentro de restricciones políticas que también deben participar en el sistema de innovación. Por ejemplo, en los primeros días del desarrollo de los coches sin conductor de Google, el Estado de California aprobó el Proyecto de Ley Senatorial 1298, que regulaba la circulación de los vehículos autónomos “en las carreteras de California” [18].

2. LIDAR, Light Detection and Ranging, es una técnica de teledetección que utiliza pulsos de luz láser para medir distancias y crear mapas 3D. La tecnología emite pulsos láser y mide el tiempo que tarda en regresar el reflejo, lo que permite calcular la distancia a un objeto o superficie.

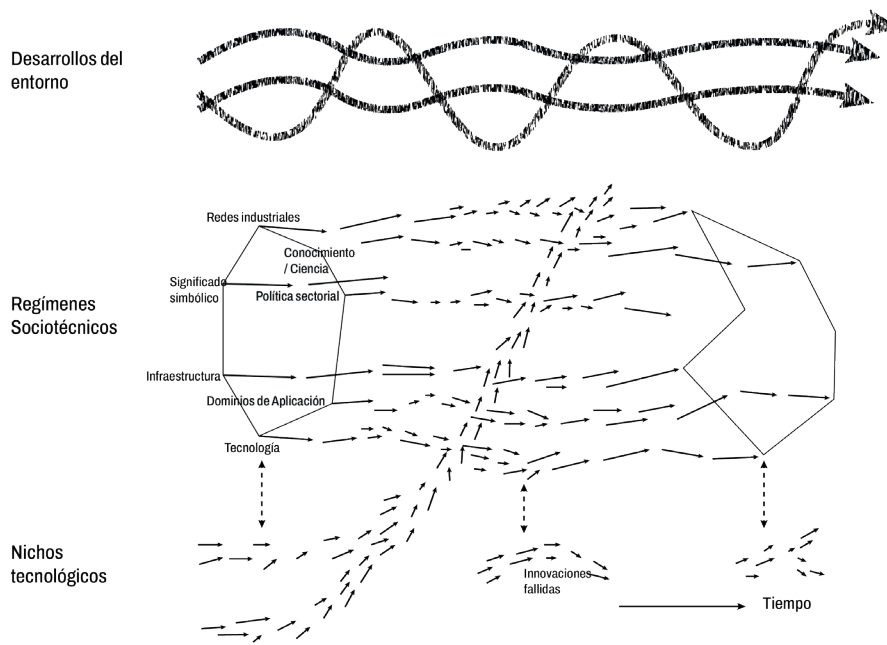


Figura 2. Una perspectiva dinámica multinivel acerca de las transiciones tecnológicas [15].

La evolución de los SdS implica el cambio en tres sistemas: el SdS de Interés, el Sistema de Innovación, y el Sistema Sociotécnico, más amplio, donde ocurren el desarrollo del escenario y los cambios de régimen [9]. La Figura 3 muestra los grados de superposición y separación entre los tres sistemas que conforman el *Modelo de los Tres Sistemas*. Esta visión se ha inspirado y adaptado del “modelo mental universal” de Lawson de un Sistema [19]. Cada sistema representa un entorno complejo de elementos que interactúan, pero en niveles de abstracción potencialmente diferentes. Los cambios en el SdS de Interés se representan como necesidades u oportunidades y que también tiene habilitadores y barreras que facilitan o dificultan el cambio. El Sistema de Innovación tiene activos existentes y sus propios habilitadores y barreras, que se pueden aplicar en un nuevo sistema que podría agregarse al SdS de Interés. Estos nuevos sistemas se describen como “intervenciones” en el SdS de Interés, que ya tiene una estructura y unas capacidades existentes que podrían alentar e incorporar nuevas capacidades. Los nuevos resultados del Sistema de Innovación deben integrarse con el Sistema Sociotécnico de Interés mediante la integración y la evaluación. Sin embargo, este proceso involucra al Sistema Sociotécnico en general, que cuenta con sus propios actores, infraestructura, políticas, etc. Para evolucionar, los actores del Sistema Sociotécnico deben integrar y

abordar las necesidades, los habilitadores y las barreras de los tres sistemas. Lo que tiene valor en un sistema puede estar en un nivel de abstracción diferente al de los otros dos. El ejemplo del vehículo sin conductor, en la Figura 4, ilustra claramente estas diferencias en los niveles de abstracción.

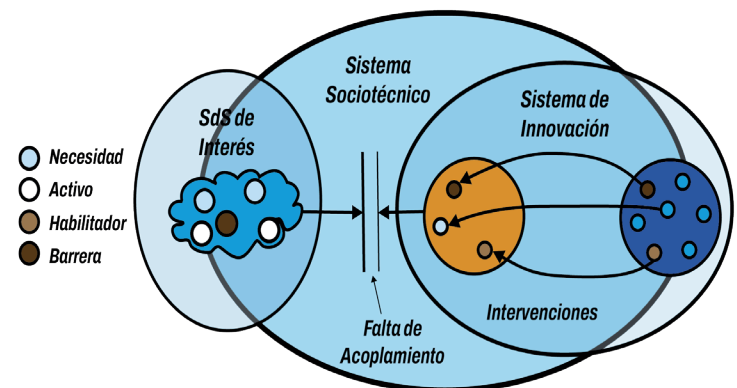


Figura 3. El Modelo de los Tres Sistemas (adaptado de Lawson) [19].

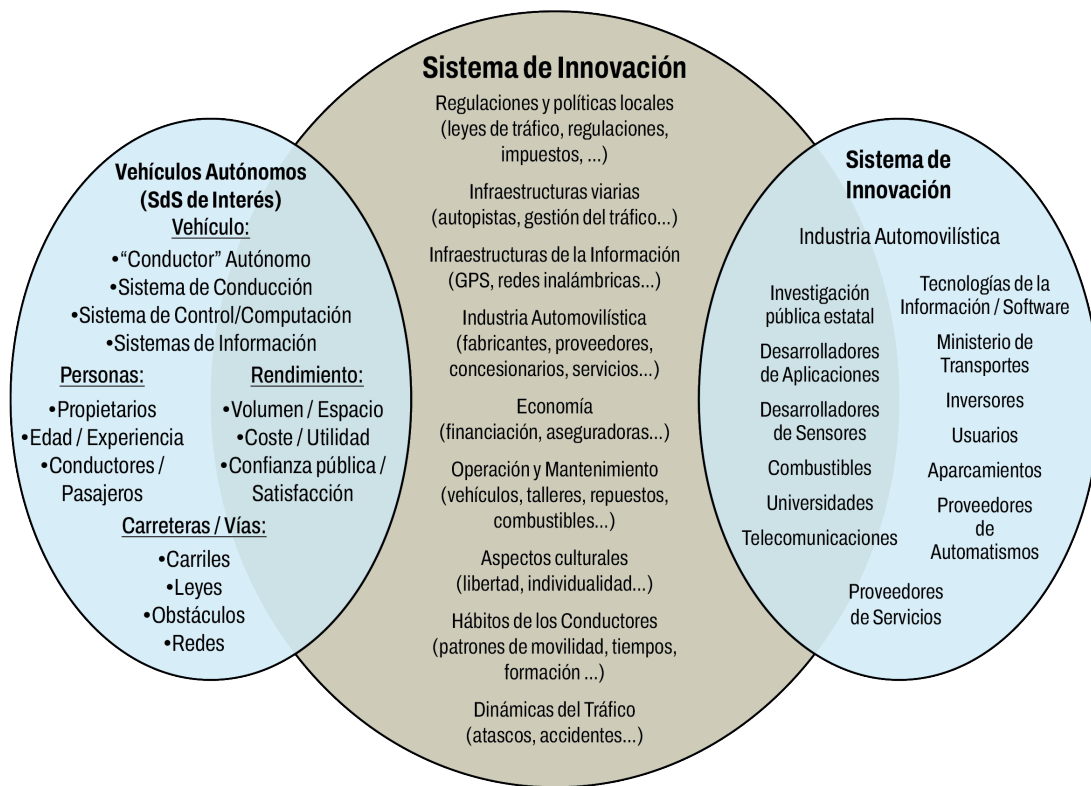


Figura 4. Modelo de los Tres Sistemas para la evolución de los vehículos sin conductor [9].

El SdS de Interés se compone principalmente de SC bien definidos y otros componentes críticos subyacentes del sistema. El Sistema de Innovación es un conjunto diverso de actores con interacciones altamente complejas. Ambos están habilitados por el Sistema Sociotécnico que, a su vez, es otro conjunto diverso de actores con interacciones complejas. Cada uno de los tres SdS está compuesto por muchos SC; las acciones que tienen lugar dentro de cada SC definen el nivel general de complejidad del SdS y su adaptabilidad a la evolución. Cabe destacar que algunos de los SC de un SdS pueden residir en el Sistema Sociotécnico de forma completamente independiente de la manifestación física en el SdS, pero también permiten que el SdS funcione eficazmente. Muchos de estos SC estarán relacionados con políticas, leyes y regulaciones. Al desarrollar el SdS de Interés, es necesario utilizar un enfoque holístico que abarque los tres sistemas. Muchos SdS emergentes, así como muchos cambios en el mantenimiento de los SdS, fracasan porque este liderazgo holístico del cambio no está presente o adopta una visión excesivamente técnica del SdS.

La mayor parte de la literatura sobre SdS aborda métodos para la gestión de los SdS y se centra en el SdS de Interés como unidad de análisis. El SdS de Interés tiende a ser menos complejo que el Sistema de Innovación que genera la evolución del propio SdS y que la del Sistema Sociotécnico que la habilita o inhibe. Tanto el Sistema de Innovación como el Sistema Sociotécnico son sistemas adaptativos complejos. Si bien los SdS de Interés pueden gestionarse, su evolución en los otros dos sistemas debe liderarse. Este liderazgo se concreta en *Procesos de Acuerdo* (acuerdos entre las partes interesadas que impulsan los resultados del SdS); las partes interesadas a nivel de SC; y las partes interesadas en los Sistemas de Innovación y el Sistema Sociotécnico. Algunos acuerdos se formalizan mediante contratos, pero muchos de los acuerdos son informales y operan en niveles inferiores de cada organización participante. Este liderazgo también debe planificar la experimentación y las “etapas de transición”, ya que es improbable que el SdS evolucione de forma simultánea. Por lo tanto, los contratos y otros acuerdos deben ser flexibles para contemplar cambios imprevistos en el esfuerzo, los costes y el plazo durante el proceso de transición.

2.5. La evolución de los Sistemas de Sistemas es liderazgo del cambio

Pero ¿quién lo lidera? Anderson señala [20]: «Desde la perspectiva de un SdS, es probable que la innovación provenga de una persona o grupo con un buen conocimiento de los sistemas que lo componen, que sepa cómo integrarlos en un SdS global (o que esté dispuesto a experimentar con la arquitectura del SdS) y que esté altamente motivado por una visión de los posibles beneficios, o capacidades, que el nuevo SdS proporcionará». La evolución del SdS suele estar impulsada por equipos de técnicos que visualizan y trabajan para lograr nuevas capacidades, pero que carecen de la autoridad para dirigir dichos cambios. Esto sucede aunque el SdS cuente con sólidas autoridades centrales (es decir, si está más *dirigido*). En el núcleo de la evolución de los SdS, existe un desafío sobre el liderazgo que debe abordarse.

Los SdS resultantes generalmente poseen las características de los sistemas adaptativos complejos, debido a la complejidad de los aspectos organizacionales en la gestión de los SdS [21]. Para tener éxito en la evolución de los SdS, es necesario centrarse más en las características de las organizaciones técnicas de los SdS que en las propias características técnicas del SdS de Interés. Muchos SdS exitosos surgen y se sustentan en la visión de un líder técnico que opera dentro de una organización asociada a uno o a varios de sus SC. Por ejemplo, para el desarrollo de vehículos autónomos, Google reclutó a los visionarios técnicos iniciales del equipo de la Universidad de Stanford que ganó la segunda carrera *DARPA Grand Challenge*, quienes, posteriormente, idearon y desarrollaron las tecnologías bajo el “*Proyecto Chauffeur*”, de Google, antes de transferir dicho proyecto a Waymo. Varios de los 16 miembros originales del equipo de Stanford aún participan en Waymo y en la evolución del SdS de taxis autónomos [22].

La evolución de los SdS, a menudo, surge de decisiones tomadas en el nivel más bajo de la evolución de los SC, lo que genera interdependencias entre individuos que necesitan trabajar juntos para hacer realidad su visión. Con el tiempo, estas relaciones a nivel del SdS se formalizan mediante métodos estándares de gestión empresarial. La gobernanza del SdS debe mantenerse separada de la gestión a nivel de los SC o, de lo contrario, la evolución del SdS se verá impulsada por cambios reactivos a corto plazo en lugar de una evolución a largo plazo [21]. No obstante, en realidad, se necesitan tanto actualizaciones/correcciones a corto plazo del SdS, como una evolución a largo plazo. En los SdS, los intentos de impulsar la innovación “desde arriba” probablemente se verán socavados por los responsables de la toma de decisiones de nivel inferior en los SC locales si se trata de una organización diferente. Al ser un sistema complejo y adaptativo, las organizaciones involucradas deben poder autoorganizarse en torno a las necesidades duales, tanto las del SdS como las de sus SC locales. Sin embargo, a medida que las propiedades emergentes del SdS maduran, las capacidades centrales de los SC tienden a converger en una sola organización mediante transferencias de personal o fusiones y adquisiciones corporativas.

2.6. Consideraciones acerca del ciclo de vida de la evolución de los Sistemas de Sistemas

Abbott afirma [10] que «Los Sistemas de Sistemas evolucionan de al menos tres maneras:

- 1) Cambios tecnológicos.
- 2) Cambios en el uso.
- 3) Cambios en los estándares e interfaces.

Los sistemas con estas propiedades no se prestan a un control jerárquico sencillo. Por otro lado, los sistemas de este tipo no son completamente amorfos. Cualquier sistema, para ser útil, debe ser capaz de realizar funciones específicas en momentos concretos. Los SdS logran este objetivo porque, en cualquier momento dado:

- 1) Incluyen un conjunto de sistemas participantes (relativamente estables).
- 2) Implementan un conjunto (relativamente estable) de estándares e interfaces.

Pero ni el conjunto de sistemas participantes, ni los estándares e interfaces, son inmutables. Evolucionan, aunque lentamente».

Debido a la naturaleza compleja de la evolución de los SdS, no existe un enfoque único para gestionar su evolución a lo largo de todo su ciclo de vida. Los SdS atraviesan periodos de cambios emergentes, a medida que las innovaciones a nivel de los SC ganan aceptación o se retiran, y periodos de estabilidad donde las capacidades a nivel de los SdS se integran en los regímenes y en el entorno social general. Las partes interesadas en los SdS deben tener cuidado en reconocer y ajustar su enfoque de gestión a la evolución de los SdS, en función de las necesidades y del contexto del cambio. Incluso si el SdS se encuentra en un periodo estable, un SC individual puede experimentar cambios:

- Es posible que las partes interesadas en un SdS desconozcan cuándo se realizará un mantenimiento o una actualización planificada de uno o más de los SC o, lo que es más importante, que no puedan sincronizar ni orquestar cuándo se van a producir estos cambios. Ante esta situación, permanecer atentos a los cambios es crucial.

- Una actualización de un SC puede deteriorar la arquitectura del SdS debido a la aparición de nuevos obstáculos a la interoperabilidad. Esto podría provocar cambios en el SC que afecten negativamente a las capacidades que proporciona al SdS o a cambios en los estándares e interfaces que siguen los distintos SC para participar en el SdS. Algún nivel de autoridad del SdS debe liderar la adaptación a la nueva estructura de este.
- La baja de un SC o del uso de dicho SC o de un estándar no implica necesariamente la baja del SdS. A menudo, la sustitución de una capacidad perdida debido a un SC que causa baja y que el SdS todavía necesita debe ser financiada por alguna autoridad del SdS.
- Las actualizaciones del SdS pueden no ser planificadas, sino que simplemente ocurren como consecuencia de la implementación de un nuevo SC independiente. Nuevamente, algún nivel de autoridad en el SdS debe abordar la integración, las pruebas y el adiestramiento de la comunidad de usuarios cuando se produzcan dichos cambios.

Por lo tanto, incluso cuando un SdS se encuentra en un período de estabilidad, quienes tienen un interés particular en sus capacidades y resultados deben esforzarse por estar al tanto de los cambios pendientes, o incluso planificados, a nivel de los SC. Los SdS tienden a atravesar períodos de mayor apertura a nuevos sistemas constituyentes y períodos de una mayor integración entre SC y, por consiguiente, de una mayor estabilidad. Una mayor integración y estabilidad suele equivaler a una mayor “dirección”, en términos de autoridad del SdS. Pero, en realidad, casi todos los SdS de gran éxito en el mercado lo logran gracias a una nueva arquitectura sin precedentes, que desalienta el control central en las primeras etapas de su ciclo de vida. Posteriormente, los SdS evolucionan hacia una mayor integración y un control organizacional unificado, lo que limita la innovación, pero genera estabilidad. Los conceptos de liderazgo del cambio y de transformación empresarial (de negocio), son más relevantes que los conceptos de gestión cuando un SdS está en proceso de evolución.

2.7. La evolución de los Sistemas de Sistemas debe ser liderada

Si bien las normas formales, como la ISO/IEC/IEEE 21839 y la 21840, detallan los procesos de gestión de los SdS, prácticamente no existe literatura que analice los enfoques de liderazgo en los SdS. Las búsquedas bibliográficas sobre el contenido relacionado con el “liderazgo de los SdS” arrojan, principalmente, contenido sobre el liderazgo en sistemas adaptativos complejos, con escaso contenido directo sobre

liderazgo en los SdS. Sin embargo, la gestión de los SdS representa claramente un desafío para el liderazgo. La norma ISO/IEC/IEEE 21840 establece [2]: “Los *Procesos de Acuerdo* son cruciales para los SdS porque establecen los modos de desarrollo y del control operacional entre las organizaciones responsables del SdS, y sus sistemas constituyentes, a menudo independientes. Los sistemas constituyentes, que son adquiridos y gestionados por diferentes organizaciones, a menudo tienen objetivos que, en sus orígenes, pudieran no estar alineados con los del SdS. Excepto en el caso de los *SdS dirigidos*, la organización del SdS no puede “asignar tareas” a una organización responsable de un sistema constituyente sin su necesaria cooperación. En un SdS “reconocido” o “colaborativo”, estas tareas se equilibran con las tareas propias de los SC, como un sistema de interés por derecho propio. Para el SdS virtual, estos *Procesos de Acuerdo* pueden ser informales, o considerarse solo para fines de análisis”. Fengali et al. señalan la siguiente pregunta fundamental [23]: “En un contexto de sistemas constituyentes independientes, ¿cuáles son las estrategias que se pueden emplear eficazmente para controlar/dirigir la evolución de un SdS, especialmente, en situaciones eventuales de falta de cooperación?” La literatura sobre el liderazgo y los *Procesos de Acuerdo* en sistemas adaptativos complejos ayuda a formar un modelo de liderazgo en los SdS, que se analizará en la sección 5.

Por lo tanto, los principales desafíos para la evolución de los SdS se asocian más con el liderazgo que con la gestión. La siguiente sección analiza los desafíos de los SdS en general, y su evolución en particular.

3. DESAFÍOS PARA LIDERAR LA EVOLUCIÓN DE LOS SISTEMAS DE SISTEMAS

Uno de los desafíos que presentan los SdS es lograr que evolucionen hacia nuevos resultados o conserven los deseables durante el tiempo que sea necesario, mientras que los SC independientes pueden, potencialmente, resistirse al cambio o evolucionar hacia resultados conflictivos a nivel de SC [23]. La Tabla 1 de la norma ISO/IEC/IEEE 21840:2019 describe, a grandes rasgos, los desafíos en la *evolución de los SdS*, que difieren de la *evolución de los Sistemas* [2].

De la Tabla 1 se desprende claramente que, si bien la evolución de los Sistemas puede tener éxito basándose en los cambios de tipo técnico, la evolución de los SdS solo puede tener éxito como un proceso de cambio a nivel organizacional. Tres referencias caracterizan los desafíos comunes de la evolución de los SdS desde tres perspectivas diferentes: Dahmann utiliza la perspectiva del *SdS de Interés* [14], Sage y Cuppan abordan los *impulsores organizacionales a nivel de los SC* [21] y Fendali et al. utilizan la perspectiva del *liderazgo de los SdS* [23]. Los desafíos mencionados se resumen en las columnas 1 a 3 de la Tabla 2. La columna 4 presenta un resumen, sobre el desafío clave derivado de los puntos en común entre los tres trabajos.

Los Sistemas tienden a...	Los Sistemas de Sistemas tienden a...
Tener un conjunto claro de partes interesadas.	Tener múltiples niveles de partes interesadas con intereses mixtos y, posiblemente, contrapuestos.
Tener objetivos y propósitos claros.	Tener objetivos y propósitos múltiples y, posiblemente, contradictorios.
Tener una estructura de gestión clara y responsabilidades claras.	Tener una estructura de gestión dispar, sin una rendición de cuentas clara.
Tener prioridades operativas claras, con mecanismos de escalamiento para resolver los conflictos entre prioridades.	Tener prioridades operativas múltiples y a veces diferentes, sin rutas de escalamiento claras para resolver los conflictos entre prioridades.
Tener un único ciclo de vida.	Tener múltiples ciclos de vida, con elementos que se implementan de forma asíncrona.
Tener un propietario claro, con la capacidad de mover recursos entre diferentes elementos.	Tener múltiples propietarios que toman decisiones individuales sobre los recursos.

Tabla 1. Comparación de la evolución de los Sistemas con la evolución de los Sistemas de Sistemas [2].

Fendali, et al. [23]	Dahmann [14]	Sage and Cuppan [21]	Resumen de los desafíos
<p>Dualidad de impulsores. Los SC mantienen una identidad propia fuera del SdS, persiguiendo sus propias misiones. Como resultado, el SdS estará expuesto a influencias que no ocurrirían si éste se gestionara de forma central. Lo que sea favorable a nivel local para los equipos y personas relacionados con los SC, impulsará sus tareas.</p>	<p>Autoridades del SdS. Cada SC tiene su propio responsable a nivel local, con sus partes interesadas, usuarios, procesos de negocio y enfoques de desarrollo. Como resultado, la estructura organizativa asumida por la Ingeniería de Sistemas tradicional, con una única autoridad responsable de todo el Sistema, está ausente en la mayoría de los SdS.</p>	<p>Las jerarquías gemelas son necesarias y útiles. En la evolución de los SdS, el rol, y no la jerarquía organizacional tradicional, dicta quién es el responsable. Es necesario establecer alianzas entre expertos, y las jerarquías de tareas asociadas a la evolución del SdS deben distinguirse de las jerarquías organizacionales de los SC.</p>	<p>Se debe desarrollar una estrategia para conectar y promover los resultados evolutivos a nivel de SdS en todos los equipos de los SC. Esto se realiza mediante tareas asociadas al rol, no a la estructura organizativa. Esto añade complejidad a la gestión, tanto de los SC como del SdS. Es posible que no exista una entidad con la autoridad suficiente para definir y gestionar esta estrategia.</p>
<p>Un entorno gerencial fragmentado. Sumado a la independencia de los SC, crea barreras para los SC responsables de las tareas de evolución del SdS, lo que resulta en desincentivos interorganizacionales, económicos y sociotécnicos.</p>	<p>Autoridades del SdS y liderazgo. La evolución en un entorno entre múltiples organizaciones supone un desafío para el liderazgo. La falta de control estructurado, normalmente presente en los SC, requiere alternativas para proporcionar coherencia y dirección, como pueden ser la influencia y los incentivos.</p>	<p>La autoridad del SdS debe obtenerse mediante el consentimiento de los desarrolladores de los SC. El liderazgo en el SdS debe entender que este consentimiento no se basa en la autoridad, sino en el interés compartido de abordar las necesidades del SdS.</p>	<p>Se debe crear un entorno de interés compartido hacia la obtención de resultados a nivel del SdS por líderes que pueden no ostentar autoridad en los SC. Este liderazgo entre organizaciones debe estar claramente definido y las personas deben estar incentivadas a participar en los resultados a nivel de SdS.</p>
<p>Visibilidad holística limitada o nula. Las barreras sociotécnicas entre los equipos de los SC crean situaciones en las que ni los roles a nivel de SdS ni los de los propios SC pueden asumir una visibilidad completa, oportuna y permanente de todos los aspectos relacionados con el SdS o su entorno completo.</p>	<p>Pruebas, Validación y Aprendizaje. Los SC, desafían la realización de pruebas al conjunto del SdS. Se requiere una comprensión clara de las expectativas y las medidas esperadas a nivel de SdS, con autoridad y financiación, o puede resultar muy difícil evaluar el nivel de rendimiento del SdS. Los ciclos de cambio de los SC pueden ser asincrónicos. Realizar pruebas al conjunto del SdS con cada cambio en cada SC resultaría prohibitivo. Esto suele resultar en un proceso de aprendizaje.</p>	<p>Lo que es bueno para mí debe ser bueno para la organización. A las personas que trabajan con las capacidades de los SC y del SdS se les debe permitir aprender y mejorar más allá de los límites de sus responsabilidades en los SC. Esto genera lealtad hacia los resultados del SdS.</p>	<p>Las oportunidades de colaboración entre los equipos desarrolladores de los SC y las personas deben ser creadas explícitamente por los líderes con responsabilidades en el SdS, para que las preocupaciones y los resultados a nivel de SdS sean visibles en todos los equipos de los SC. Esto representa un desafío de comunicación entre múltiples organizaciones.</p>
<p>Control holístico limitado o nulo. Ninguna de las partes, ni siquiera las elegidas para gestionar las consideraciones a nivel del SdS, puede asumir un control general o directo sobre todos los SC, ni sobre sus procesos de evolución. El liderazgo a nivel de SdS solo puede negociar con ellos, no controlarlos.</p>	<p>Perspectivas de los Sistemas Constituyentes. Alguna autoridad debe abordar técnicamente los problemas que surgen del hecho de que los SC identificados para el SdS pueden tener sus limitaciones en lo que se refiere a su capacidad para apoyarlo.</p>	<p>Gestión de espacios vacíos. Los equipos y las personas de los SC viven dentro de dos círculos concéntricos de responsabilidad: el círculo interno representa el núcleo de su responsabilidad y, el externo, el límite de su autoridad. La zona intermedia es el área discreta, donde se desarrollan las capacidades del SdS.</p>	<p>Las responsabilidades y la rendición de cuentas del SdS deben negociarse formalmente en contratos a nivel de SC o, informalmente, en modelos de negocio compartidos. Las relaciones negociadas, tanto formales como informales, pueden ser muy complejas, especialmente si no existe una autoridad central en el SdS.</p>
<p>Mayor potencial de incompatibilidad y conflictos. Donde un responsable de un SC ve un SdS en particular, un responsable de otro SC podría ver el SdS de forma diferente o no ver ninguno. Esta disparidad puede deberse a una simple falta de visibilidad o a determinados sesgos. Es difícil introducir cambios evolutivos sin generar inconsistencias.</p>	<p>Capacidades y requisitos. Las necesidades de un SdS evolucionan independientemente de las necesidades individuales de los SC, lo que a menudo exige que los SC asuman nuevos requisitos o se reemplacen unos SC participantes por otros SC participantes. Estas relaciones deben documentarse formalmente mediante estándares, requisitos, etc. Es posible que no todos estos se conozcan de antemano.</p>	<p>Responsabilidades de los equipos de los SC. Los desarrolladores de los SC deben "firmar su trabajo" o, lo que es lo mismo, rendir cuentas formalmente de sus responsabilidades para finalizar el trabajo que es necesario hacer para el SdS. Esto debe comunicarse a otros equipos de SC para gestionar las inconsistencias.</p>	<p>En la práctica, debe existir una estructura de rendición de cuentas a nivel de SdS para supervisar (si no gobernar) las responsabilidades individuales de los SC y crear una comunicación documentada de los acuerdos. Esto puede no estar presente en el desarrollo inicial de un SdS, pero debe evolucionar a medida que la estabilidad del SdS se normaliza. Esto suele ocurrir cuando se transfieren los SC a una autoridad común del SdS.</p>

Tabla 2. Desafíos en la evolución de los Sistemas de Sistemas.

4. MÉTODOS PARA GUIAR LA EVOLUCIÓN DE LOS SISTEMAS DE SISTEMAS

En el caso de los sistemas de ingeniería, existe cierta autoridad que impulsa los objetivos emergentes del SdS, así como un conjunto de partes interesadas con responsabilidades orientadas a las tareas, tanto a nivel de los SC como del SdS, para crear, gestionar y mantener el SdS en sus estados de evolución. Con los SdS, las autoridades tienden a evolucionar a lo largo del ciclo de vida, pero los responsables de las tareas tienden a persistir. El *Modelo de Ondas* (representado en la Figura 1) considera esto como un proceso de Ingeniería de Sistemas, en el que se puede realizar un análisis y una planificación intencionados del siguiente estado de evolución. El *modelo de gestión de la transición* lo considera un proceso de innovación, en el que estos estados de evolución deben guiarse, pero no pueden planificarse discretamente. En la práctica, ambos modelos son relevantes. En base a los desafíos resumidos en la Tabla 1, el análisis del siguiente estado de evolución debe reflejar no solo cómo ocurrirá esta evolución, sino también quién la habilitará y creará los cambios necesarios.

4.1. Organización para la evolución de los Sistemas de Sistemas

El modelo RACI [26] es un marco organizativo útil para la claridad de los acuerdos en la evolución de los SdS y puede utilizarse para clasificar dónde se puede aplicar con mayor eficacia el liderazgo en los SdS. El acrónimo RACI se refiere (en inglés) a “*Responsible*³, *Accountable*⁴, *Consulted*⁵, and *Informed*⁶”. En consonancia con Sage y Cuppan, la evolución de los SdS se produce, en realidad, a partir de individuos o equipos que pertenecen a la categoría “*Responsible*” (aquellos que completan las tareas necesarias para la evolución de los SdS), no a la categoría “*Accountable*” (la autoridad o la persona responsable última de los resultados del SdS). Además, en algunos SdS, no existe una única autoridad responsable. Pertenecen a la categoría “*Consulted*” aquellos cuyas opiniones se solicitan

para facilitar la evolución, a menudo expertos en la materia, con quienes los “*responsables*” y la “*autoridad*” deben mantener una comunicación bidireccional. Este proceso de colaboración debe facilitarse entre las partes interesadas, responsables y autoridades de los SC. Finalmente, pertenecen a la categoría “*Informed*” aquellos que necesitan mantenerse al día sobre la evolución del SdS, en particular sobre la finalización de las tareas. Se podría considerar esta labor de información como una forma para que los detalles del estado actualizado del SdS se transfieran a todos los SC. Esto último puede resultar especialmente difícil en un SdS, ya que los SC individuales pueden tener razones competitivas para proteger los detalles de su estado real [25].

Aplicando el modelo RACI a los desafíos resumidos de la Tabla 2, podemos sugerir cambios en las prácticas de la Tabla 3:

Resumen del desafío	Cambios en la práctica de la Evolución de los SdS
<p><i>Se debe desarrollar una estrategia para conectar y promover los resultados evolutivos a nivel del SdS en todos los equipos de los SC. Esto se realiza mediante tareas asociadas al rol, no a la estructura organizativa. Esto añade complejidad a la gestión, tanto de los SC, como a la del propio SdS. Es posible que no exista una entidad con autoridad suficiente para definir y gestionar esta estrategia.</i></p>	<p>Es posible que no exista una autoridad central para dirigir la evolución del SdS. Habrá uno o más líderes responsables de los resultados del SdS, que puedan tener distintos niveles de control o influencia sobre las tareas a ejecutar a nivel de los SC que conducirán a dicha evolución. Las estrategias para crear y mantener las capacidades y los resultados del SdS evolucionarán con la propia evolución del SdS, y los líderes responsables pueden evolucionar con ellas. La dirección de la evolución del SdS debe estar basada en una estrategia. Rouse lo define como un proceso de transformación empresarial [26].</p>
<p><i>Se debe crear un entorno de interés compartido para obtener los resultados a nivel del SdS por líderes que no ostenten autoridad en los SC. Este liderazgo entre organizaciones debe estar claramente definido y las personas deben estar incentivadas a participar en los resultados a nivel del SdS.</i></p>	<p>El SdS debe construirse sobre intereses o preocupaciones compartidas. Una parte interesada con una posición de liderazgo debe dedicar esfuerzos a establecer y mantener un interés compartido. Los líderes deben identificar intereses comunes viables entre los equipos de los SC y comunicarlos a un conjunto amplio y, posiblemente, remoto, de equipos responsables. Esto se corresponde con la heurística n° 2 de Fendali et al. [23].</p>

3. Responsable (ejecutor): quien realiza el trabajo o ejecuta la tarea.

4. Responsable último (autoridad): quien asume la responsabilidad final de que la tarea se haga correctamente.

5. Consultado: personas que deben ser consultadas antes o durante la ejecución. Suelen ser expertos o partes interesadas.

6. Informados: Personas que deben ser informadas del progreso o los resultados, pero no participan activamente.

Resumen del desafío	Cambios en la práctica de la Evolución de los SdS
<p><i>Las oportunidades de colaboración entre los equipos de los SC y las personas se deben crear explícitamente por los líderes con responsabilidades o autoridad en el SdS, para que las preocupaciones y los resultados a nivel de SdS sean visibles en todos los equipos de los SC. Este es un desafío de comunicación entre múltiples organizaciones.</i></p>	<p>Una persona con una posición de liderazgo en el SdS, ya sea responsable o con autoridad, debe crear un entorno que facilite el establecimiento de relaciones a nivel de los SC, donde los responsables de las tareas puedan colaborar y desarrollar interdependencia en sus responsabilidades. Esta interdependencia sirve para distribuir el poder entre los SC y debe ser un proceso activo durante la actualización o evolución de las capacidades del SdS. Esto incluye la colaboración entre los "consultados" en nombre de los SC y mantener "informadas" a las demás partes interesadas.</p>
<p><i>Las responsabilidades y la rendición de cuentas del SdS deben negociarse formalmente en contratos a nivel de SC o, informalmente, en modelos de negocio compartidos. Las relaciones negociadas, tanto formales como informales, pueden ser muy complejas, especialmente si no existe una autoridad central en el SdS.</i></p>	<p>Los intereses compartidos son necesarios, pero insuficientes, para el éxito en las tareas necesarias para la evolución del SdS cuando las prioridades de los SC interfieren con las responsabilidades de los propios SC para con el SdS. Los incentivos para desarrollar el trabajo a nivel del SdS deben aplicarse y plasmarse en acuerdos documentados, negociados entre los SC, ya sean contratos formales u otras relaciones comerciales más indirectas. Estos incentivos pueden ser económicos, basados en la colaboración, definidos en las políticas, etc. [23]</p>
<p><i>En la práctica, debe existir una estructura responsable a nivel de SdS para supervisar (si no gobernar) las responsabilidades individuales de los SC y crear una comunicación documentada de los acuerdos. Esto puede no estar presente en el desarrollo inicial del SdS, pero debe evolucionar a medida que se normaliza la estabilidad del SdS. Esto suele ocurrir al transferirse los SC a una autoridad común del SdS.</i></p>	<p>Esta es la práctica más difícil de definir, ya que existen muchas formas de generar la "responsabilidad" relacionada con las capacidades y los resultados del SdS. La más sencilla es adoptar un enfoque más directo, utilizando contratos explícitos con los SC para los requisitos del SdS. El riesgo de este enfoque radica en que estos contratos son difíciles de modificar a medida que surgen las capacidades del SdS y se aprende su valor, lo que sugiere que también deberían evolucionar y ser flexibles en las primeras etapas del SdS. La tendencia en los SdS comerciales es que la organización que obtiene el mayor valor del SdS aumente gradualmente su autoridad colocando a distintos SC bajo su autoridad, a menudo mediante fusiones y adquisiciones. Este es un proceso más evolutivo. También existe la oportunidad de aprovechar los efectos de red, que a menudo se forman en torno a políticas, estándares o intereses económicos comunes (como las regulaciones) [23].</p>

Tabla 3. Desafíos de la evolución del Sistema de Sistemas y cambios en la práctica de su evolución.

La Tabla 3 enfatiza que, al planificar la evolución del SdS, el análisis de las relaciones entre las partes interesadas en el SdS suele ser más importante que las relaciones entre las capacidades del SdS. Las formas para desarrollar estas relaciones entre las partes interesadas no están bien cubiertas en la literatura existente sobre el SdS, pero pueden derivarse de la literatura sobre los Sistemas de Innovación y los sistemas empresariales. Los líderes del SdS son responsables últimos ("accountables") de la evolución del SdS, pero sólo en algunas ocasiones tienen la autoridad para implementar los cambios necesarios. Resumiendo la Tabla 3 desde una perspectiva de liderazgo, los líderes del SdS deben esforzarse por:

- Definir y comunicar estrategias para la evolución del SdS.
- Definir y crear áreas de acción de interés compartido entre los distintos equipos de los SC.
- Crear un entorno donde se puedan establecer relaciones a nivel de los SC y donde los responsables de ejecutar las tareas puedan colaborar entre ellos.
- Definir y aplicar incentivos para el trabajo a nivel del SdS y negociarlos para incorporarlos en acuerdos entre los SC.
- Desarrollar modelos de negocio y relaciones que responsabilicen a los SC de los resultados a nivel del SdS.

A alto nivel, estos cambios en las prácticas relacionadas con la evolución de los SdS implican que quienes son responsables de las tareas:

- Conocen la estructura del SdS y el contexto asociado o los impulsores externos, así como las estrategias para su evolución.
- Conocen a las principales partes interesadas, al menos entre los equipos a nivel de SC donde existen interdependencias, con quienes podrían compartir intereses.
- Conocen suficientemente a las partes interesadas a nivel del SdS, a las que se debe consultar y mantener informadas.
- Responden a los objetivos y estrategias de cambio a nivel del SdS con tareas de trabajo definidas que se negocian en acuerdos.
- Aceptan la responsabilidad de la implementación ("firman su trabajo").

Generalmente, para aquellos que son “responsables”, estos intereses y preocupaciones son de tipo técnico. Los “responsables últimos” (aquellos que rinden cuentas, incluso si carecen de autoridad) deben preocuparse por el SdS como una empresa con objetivos técnicos, comerciales y de misión. Estos últimos incluyen aspectos de carácter sociotécnico. Quienes son “consultados” desempeñan un papel importante en la generación de conocimiento en todas las dimensiones sociotécnicas del SdS.

4.2. Preparando la evolución de Sistemas de Sistemas

El valor del *Modelo de Ondas*, al menos como estrategia, reside en que define el “Análisis del SdS” como un proceso activo y continuo a nivel del SdS. Basándonos en el marco I-SoS y en investigaciones adicionales sobre transformación empresarial [26], recomendamos a quienes son responsables últimos (“accountables”) de los objetivos de cambio a nivel del SdS, que realicen análisis periódicos de la evolución del SdS en cuatro dimensiones: *Definición del SdS, Actores de SdS, Objetivos de cambio del SdS e Implementación del SdS*.

- 1. Definición del SdS:** al analizar las rutas de evolución de un SdS, se debe comenzar por definir claramente el estado actual del SdS de Interés, incluido el contexto (o entorno actual), que constituye el contexto sociotécnico en el que opera el SdS. Téngase en cuenta que en la Figura 3 esto incluye las necesidades de las partes interesadas que el SdS satisface, el conjunto actual de activos que utiliza (no solamente un listado de SC, sino también los detalles de cómo cada uno contribuye al SdS) y los habilitadores y las barreras para el cambio del SdS (un aspecto muy importante, que a menudo se descuida).
- 2. Actores del SdS:** Los responsables últimos (“accountables”) de los objetivos de cambio del SdS deben definir todos los actores en cada capa de SC/SdS y “lo que aportan”: sus modelos de negocio, recursos, redes, instituciones, etc. Esto también debe incluir el conocimiento, si está disponible, sobre qué otros programas están impulsando el cambio a nivel de los SC.
- 3. Objetivos de cambio del SdS:** Este análisis debe reflejar el estado final deseado de la evolución del SdS en términos de nuevas capacidades, quién las utilizará y cuál es la prioridad de cada una. El objetivo es identificar las principales medidas de rendimiento del SdS actual, las medidas de rendimiento de la evolución deseada y quién sería el encargado de lograrlas.

- 4. Implementación del SdS:** La dimensión de la “implementación” sirve para construir un modelo en todas las otras dimensiones o aspectos, considerando los nuevos resultados (u objetivos) del SdS y las interacciones que los generan. Es probable que el estado final deseado no se alcance en un único paso, por lo que dividirlo en períodos de transición puede facilitar el proceso de planificación para la evolución. La dimensión de la implementación debe incluir las estimaciones del plazo y del coste, así como la procedencia de los recursos necesarios. Cabe destacar que, en la Figura 3, lo anterior se denominan “intervenciones” en el SdS actual, y la dimensión de la implementación debe utilizar los recursos del Sistema de Innovación para generar los cambios deseados en el SdS.

Por lo general, no es posible predecir exactamente qué es lo que emerge, pero se pueden observar las tendencias del cambio y los eventos que están precipitando dicho cambio tales como las señales de cómo nuevas capacidades emergentes deberían ocurrir en el SdS. Por ejemplo, ¿qué emerge en el contexto del SdS relacionado con los vehículos autónomos, independientemente de los vehículos como sistemas? En uno de los casos de negocio de *Waymo*, esta “emergencia” es un nuevo tipo de servicio de taxi que opera sin taxistas. ¿Cuáles son los SC independientes que conforman un servicio de taxi autónomo? ¿Cuáles son sus costes relativos, rendimiento, disponibilidad, interfaces de usuario, etc.? ¿Quiénes son las organizaciones independientes que contribuyen a este SdS? ¿Cómo ha evolucionado?

Para planificar la evolución del SdS, los líderes responsables últimos (“accountables”) de este deben evaluar [27]:

- ¿Cuál es el estado actual del SdS? (“SdS de Interés”, según la Figura 3).
- ¿Cuáles son las nuevas capacidades deseadas del SdS? (La Figura 3 las describe como “intervenciones”).
- ¿Cuál es la prioridad de cada capacidad deseada?
- ¿De qué recursos se puede disponer? (estos son los recursos del Sistema de Innovación).
- ¿Cuándo se desea introducir estas capacidades?
- ¿Cuál será el modelo de negocio resultante?

El marco de análisis deseado, que captura todas las intenciones de las partes interesadas, representaría entonces a todos los componentes de los SdS relacionados, distinguidos por estrato social y mediante un modelo de interrelación que considera los habilitadores y las barreras, actores, interacciones y resultados de la evolución deseada del SdS. Este modelo de interrelación puede mapearse y visualizarse como una herramienta para lograr el consenso

entre las partes interesadas en la estructura del SdS y los objetivos de rendimiento y para modelizar computacionalmente el sistema [29, 30]. En la siguiente sección, se presentan dos marcos de herramientas cualitativas. Ambas son herramientas holísticas. Como señala Abbot [10]: «Un Sistema de Sistemas se entiende mejor no como una jerarquía de sistemas que lo componen, sino como un entorno dentro del cual operan otros sistemas, y que puede permitir la incorporación de nuevos sistemas que se basen en los sistemas ya existentes. Además, para comprender plenamente un Sistema de Sistemas, no solo debe verse como un entorno para otros sistemas, sino que debe entenderse en términos del entorno más amplio en el que él y sus sistemas participantes coexisten. En otras palabras, una perspectiva de Sistema de Sistemas requiere mirar *hacia afuera*, desde el Sistema, en lugar de *hacia adentro*, hacia sus componentes jerárquicos».

4.3. Dos herramientas para analizar las rutas de evolución de los Sistemas de Sistemas

Volviendo al *Modelo de los Tres Sistemas* (ver Figura 3), la mayoría de los análisis de sistemas separan o evitan por completo la complejidad de estos tres sistemas al diseñar una “intervención” o evaluar la viabilidad de un proyecto. Contar con un conjunto de herramientas que respalde un enfoque de SdS, teniendo en cuenta los tres sistemas para un análisis integrado, proporciona un enfoque novedoso en el campo de la Ingeniería de Sistemas y los sistemas adaptativos complejos. Un conjunto de herramientas diseñado para considerar estos tres sistemas en su totalidad proporciona un análisis de las interacciones, relaciones y evolución entre ellos y dentro de ellos, ofreciendo una perspectiva privilegiada para evaluar el potencial impacto impulsado por la innovación [8].

El análisis de los SdS representa un desafío para el pensamiento sistémico. Las herramientas, al menos inicialmente, son cualitativas y están diseñadas para fomentar el pensamiento holístico sobre los tres sistemas en juego. En el análisis tradicional de sistemas sociales, se nos anima a ver a los SdS en tres niveles: *macro* (social, institucional), *meso* (grupos, organizaciones) y *micro* (individuos). Paralelamente, los SdS deben observarse como un sistema empresarial multinivel, con cuatro fenómenos estratificados: *capa social*, *capa institucional/económica*, *capa física/de proceso* y *capa humana* [29]. La Figura 5 muestra el valor de este tipo de análisis, tanto en la planificación del Modelo de Ondas de los SdS, como en la planificación de la *gestión de la transición*, que sigue un conjunto similar de cambios estratificados. La implementación de la evolución de los SdS se da en las capas humana y física, mientras que la planificación y la ejecución son procesos sociotécnicos y socioeconómicos de las instituciones que participan en el SdS. La visión multicapa apoya directamente el análisis, tanto “top-down” como “bottom-up”, de los fenómenos que impulsan la evolución del SdS, en particular los aspectos humanos para una eficaz rendición de cuentas (“Accountability”) en los SdS.

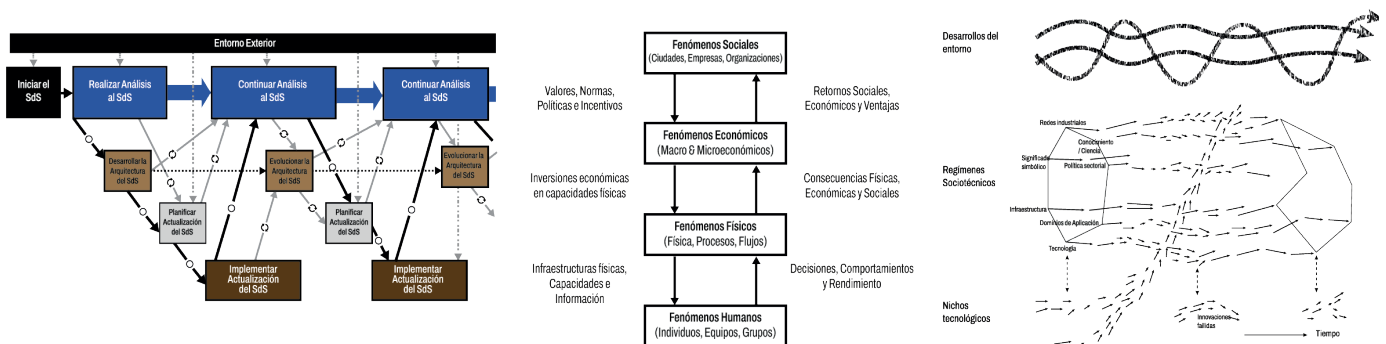


Figura 5. Fenómenos multicapa en la evolución de los Sistemas de Sistemas.

Con esto en mente, se presentan dos herramientas de pensamiento sistémico que, juntas, ayudan al análisis del SdS: la “herramienta de análisis del contexto” y la “herramienta multinivel de planificación de la evolución de los SdS”.

4.3.1. Herramienta de análisis del contexto

La herramienta de análisis del contexto en la Tabla 4 [30] puede utilizarse para organizar la información esencial que refleje los resultados deseados de la evolución de los SdS, los actores e instituciones que liderarán (o se opondrán) a la evolución, las actividades que realizan y cómo interactuarán para influir en la evolución y cuáles son los factores habilitadores y las barreras del proceso de evolución. Cada uno de estos factores se organiza en los cuatro fenómenos multicapa de la Figura 5 (social, económico, físico y humano). El valor de esta herramienta reside en institucionalizar la reflexión sobre la naturaleza multicapa de los SdS y la importancia de analizar los factores habilitadores y las barreras del cambio.

	Habilitadores y Barreras	Actores (y lo que aportan)	Interacciones / Actividades	Resultados
Dominio Social	Regulaciones gubernamentales, leyes, políticas, infraestructuras, medio ambiente, geografía, condiciones económicas, seguridad, estrategias de inversión globales, nacionales y regionales.	Gobiernos, estados-nación, regiones, organizaciones de gestión de infraestructuras, cooperativas, organizaciones no gubernamentales.	Tendencias económicas, tendencias demográficas, cambios en la gobernanza, atributos de red, ideologías, tendencias del mercado.	Adhesión y aceptación, cambios en el empleo (macro), logro de objetivos, cambios de políticas, nuevas infraestructuras, crecimiento (macro), cambio (macro).
Instituciones y Economía	Propiedad, entorno/capacidad de producción, redes de distribución, capacidad logística, controles de importación/exportación, normas.	Organizaciones gubernamentales, organizaciones comerciales, universidades, instituciones de investigación, grupos de presión, organizaciones de normalización, sindicatos, incubadoras.	Acceso y control del mercado, acceso a los recursos, acceso al crédito, transferencia de tecnología y patentes, relaciones organizacionales, emprendimiento.	Capacidades, disrupción y continuidad, resiliencia, eficiencia, medidas de cambio, estrategias, líneas de productos y sistemas, procesos.
Fenómenos físicos, Procesos y Flujos	Estructuras legales, demanda, restricciones de precios, restricciones de procesos, estándares, acuerdos, contratos, preparación/madurez tecnológica.	Liderazgo, prácticas laborales, estructuras de gestión organizacional, formación, prácticas de innovación, prácticas de desarrollo, prácticas de mercadotecnia, prácticas de empleo.	Flujos de información, flujos de recursos, transferencia de conocimiento, flujos financieros, flujos de fabricación, actividades laborales, interacciones organizacionales, interacciones con los clientes, interacciones con el mercado.	Productos, compra, venta, adopción de tecnología, participación de mercado/liderazgo, innovaciones, colaboraciones, métricas de rendimiento.
Personas y Activos	Cultura, redes, estructuras comunitarias, acceso a la información, proximidad a los recursos y otros, derechos de propiedad.	Personas (individuos y clases), empresarios, líderes, ejecutivos, habilidades, competencias, salud, tecnologías, recursos materiales, capital financiero.	Producción, consumo/compra, automatización del trabajo, ingresos, membresía, trabajo en equipo, aprendizaje.	Casos de uso, herramientas y aplicaciones, empleo, roles laborales, resultados laborales, inclusión, salud, nivel de vida, educación, confianza.

Tabla 4. Herramienta de análisis del contexto [29].

El valor adicional de la *herramienta de análisis del contexto* reside en la recopilación holística de los factores que contribuyen a la evolución del SdS. Puede servir como una base de datos continua de información útil para la planificación evolutiva del SdS. Puede capturar perspectivas importantes, como los intereses compartidos (como habilitadores) y las relaciones comerciales (como flujos), de una forma sencilla de usar y mantener. Para utilizar esta herramienta, se recomienda seguir los siguientes pasos:

1. Los resultados (“*outcomes*”) del SdS representan generadores de valor para las partes interesadas del sistema, derivados de las nuevas capacidades deseadas para el SdS. Se comienza con la columna de “Resultados” y se enumeran los resultados esperados de la evolución del SdS en cada nivel. Esto debe centrarse en las nuevas capacidades deseadas del SdS y los SC apropiados, así como en su valor económico en la fila de las “Instituciones y Economía”. En la fila inmediatamente superior se presentan los nuevos resultados esperados en el “Dominio Social”. En las filas inferiores se presentan los resultados a nivel de “Procesos” y de “Personas”, que describen la evolución del SdS.
2. Se revisan y registran a las partes interesadas que apoyarían (o se opondrían) a estos resultados como “Actores” y los “Habilitadores (o Barreras)” para la evolución del SdS. Estos incluirían a las partes interesadas responsables, las autoridades con responsabilidad última, las partes consultadas o, simplemente, las partes informadas.
3. Finalmente, se identifican las “Interacciones” entre las partes interesadas y lo que aportan (tecnologías, recursos, etc.), así como las “Actividades” necesarias para lograr la evolución deseada del SdS. Esta columna analiza los procesos, las actividades y los comportamientos. Es necesario centrarse en las afirmaciones que describen las transformaciones que ocurren entre las entidades del sistema. Por ejemplo, la fabricación de sensores de menor coste y la simulación de millones de kilómetros de conducción representan dos fenómenos que contribuyen a la comercialización de los vehículos autónomos. El primero es un proceso y, el segundo, un conjunto de actividades. La lluvia de ideas y la consulta con expertos contribuyen a este paso.

Normalmente, se requieren varios “talleres” con las partes interesadas para completar esta tabla-herramienta. Éstas son buenas oportunidades para reunir a las partes interesadas a nivel de servicio al cliente y generar un interés común en torno a la evolución del SdS. Se debe mantener esta tabla a un nivel resumido, ya que está diseñada como herramienta de comunicación. Se actualizará periódicamente y, según sea necesario, para planificar las estrategias de evolución del SdS.

El propósito de esta herramienta radica en la importancia de su uso. Su objetivo es facilitar la planificación integral de la evolución del SdS, considerando todos los aspectos del *Modelo de los Tres Sistemas* y la naturaleza multicapa sociotécnica y socioeconómica de las grandes empresas. Incluso si no utiliza la herramienta activamente, se debe considerar la Tabla 4 como una guía útil para todos los fenómenos que impulsan la evolución de los SdS.

4.3.2. *Herramienta multinivel de planificación de la evolución de los Sistemas de Sistemas*

La *herramienta multinivel de planificación de la evolución de los SdS*, presentada en la Tabla 5, es una herramienta fundamental de planificación holística para la evolución de los SdS. Fue desarrollada originalmente para visualizar los detalles de los *procesos de gestión de la transición* [31].

La herramienta integra la información principal para la toma de decisiones de todos los aspectos del análisis de los SdS en un único marco, que permite al equipo visualizar las rutas de transición viables para el conjunto del SdS desde el nivel inferior representado por las organizaciones y sistemas de los SC. Estas rutas se basan en los cambios deseados dentro del espacio del problema y el potencial de innovación en el contexto de interés. La mitad superior de la tabla describe el contexto de la evolución del SdS en términos de *tendencias, eventos y señales de cambio* a nivel macro. La mitad inferior define la evolución del SdS en términos de posibles innovaciones, habilitadores y barreras de cambio relevantes, descritos al nivel de SC del SdS.

Al igual que ocurre con la *herramienta de análisis del contexto*, el uso de la Tabla 5 comienza identificando las nuevas capacidades deseadas del SdS en la columna de la derecha. Este análisis se centra menos en el contexto general del *Modelo de los Tres Sistemas* y más en las nuevas capacidades deseadas (y prioridades) específicas de la evolución del SdS.

	Estado Actual	Señales de Cambio	Periodo de Iteración		Estado final deseado
Eventos a nivel SdS	Se enumeran todos los eventos pasados que presentaron oportunidades para la evolución del SdS.	¿Qué acontecimientos indican que el SdS debe cambiar?	Se describe el plan para la implementación de estos cambios y cómo se evaluarán.		Se proporciona una narrativa de cómo se introducirán estas capacidades del SdS y qué impacto tendrán.
Tendencias a nivel de SdS	¿Qué resultados ofrece el SdS, y a quién? ¿Qué cambios a nivel de SC se están desarrollando o contemplando actualmente?	¿Cuáles son las tendencias actuales que indican que se necesitan nuevas capacidades del SdS, o que se están convirtiendo en oportunidades para su evolución?	Se describe cómo integrará, probará, implementará y medirá la adopción por parte de los usuarios de la evolución planificada.		¿Qué nuevos resultados debería proporcionar el SdS? ¿Cuáles son las nuevas capacidades deseadas? ¿Qué SC deben añadirse, eliminarse o modificarse? ¿Cómo priorizaría estas capacidades?
	Capacidad principal actual del SdS		Capacidades evaluadas en la iteración		Capacidad primaria deseada para el SdS
Habilitadores	Se enumeran las políticas, tecnologías, procesos, organizaciones, etc. habilitadores de la tabla de Análisis de Contexto.	¿Cuáles son las tendencias actuales que indican cambios en un habilitador o barrera del SdS? Nuevas regulaciones, panorama económico, panorama de amenazas, cambios en las políticas, etc	Cambios esperados en políticas, tecnologías, procesos, organizaciones, etc. habilitadoras, de la tabla de Análisis de Contexto.	Durante la fase de transición, ¿se identificaron habilitadores adicionales o barreras que hubiera que gestionar?	¿Qué cambios en las políticas, tecnologías, procesos, organizaciones, etc. habilitadoras son necesarios para alcanzar el estado deseado (de la tabla de Análisis de contexto)?
Barreras	Se enumeran las posibles barreras actuales para la evolución de la tabla de Análisis de Contexto.		Cambios esperados en las barreras potenciales a la evolución, de la tabla de Análisis de Contexto.		¿Qué barreras deben superarse para alcanzar el estado deseado? (de la tabla de Análisis de Contexto).
Institucional / Organizacional	Se describen en detalle las organizaciones actuales a nivel de SdS y SC, así como otras en el contexto de uso, con descripciones de su importancia	¿Cuáles son los nuevos conceptos de uso operativo que el SdS podría ofrecer y quién podría proporcionarlos?	¿Qué nuevas organizaciones a nivel de SdS y de SC participarán? ¿Quiénes participarán en el concepto de uso operativo?	¿Cómo se mide el éxito de la transición a las nuevas capacidades deseadas para el SdS?	¿Cuáles son los nuevos casos de uso para las capacidades deseadas y quién podrá proporcionarlas?
Programas y Actores involucrados	Se describen en detalle los actores/partes interesadas actuales a nivel de SdS y de SC y lo que se sabe sobre su trabajo		¿Qué programas (a nivel de cambio en los SC) o qué programas para producir nuevos SC se integrarán en este período?		¿Qué programas (a nivel de cambio en los SC) o qué programas para producir nuevos SC se deberían planificar, integrar e implementar en este estado final?

Tabla 5. Herramienta multinivel de planificación de la evolución de los Sistemas de Sistemas [32].

Las nuevas capacidades deseadas en la evolución del SdS (y su prioridad) se describen mejor, inicialmente, como una narrativa que captura un “evento” que implementa las nuevas capacidades para la comunidad usuaria y el impacto que estos cambios tendrán en dicha capacidad. Esta es la perspectiva “macro” de los resultados esperados para el SdS. Esta narrativa, basada en eventos, está respaldada por las tendencias a nivel del SdS y de los SC que la impulsan.

Cada período evolutivo de un SdS comienza con un estado actual, pasa por un período de iteración donde las nuevas capacidades “resurgen” (a menudo hay múltiples iteraciones) y alcanzan el estado final deseado. Estas iteraciones proporcionarán información sobre las rutas evolutivas del SdS y ayudarán a organizar la implementación real de las nuevas capacidades. Éstas deben describirse, primeramente, a nivel (macro) en el SdS. Habrá múltiples iteraciones (o “transiciones”) en la evolución del SdS, aunque la tabla sólo representa una, por limitaciones de espacio.

Comenzando por la parte inferior de la tabla, se enumeran los “Programas” y los “Actores” del cambio al nivel de los SC en su “Estado actual”, en el “Periodo de Iteración” y en el “Estado final deseado”. A continuación, subiendo una fila, se indican las “Instituciones u Organizaciones” involucradas y las autoridades y modelos de negocio representativos que respaldarían el cambio a nivel del SdS. Un aspecto único de esta tabla son las dos filas siguientes, que capturan los “Habilitadores” y las “Barreras” para el cambio a nivel del SdS. Dado que los líderes de SdS son responsables últimos (“accountables”), pero pueden no tener autoridad directa sobre las actualizaciones de los SC, generalmente sólo pueden habilitar su implementación o crear barreras para el cambio a nivel de aquellos SC que pueden tener un impacto negativo en el rendimiento a nivel de SdS.

4.4. Revisión de las capas: Enfoque socioeconómico de la evolución de los Sistemas de Sistemas

Al utilizar estas herramientas para realizar análisis del SdS, es importante aplicar un marco de referencia socioeconómico en lugar de un marco de referencia técnico. Esto implica, en ambas herramientas, centrarse en los factores (y las filas) relacionados con las “Instituciones” y la “Economía”. Si bien los SdS suelen existir en las categorías “Reconocidos” y “Colaborativos”, siempre debe haber una organización líder que aproveche el valor económico de las capacidades de los SdS y se convierta en el líder del cambio. Como se indica en la sección 2.2, los SdS se desarrollan y evolucionan en cuatro capas: la *capa tecnológica*, la *capa de aplicaciones*, la *capa de información* y la *capa de negocio*.

Adicionalmente, también se debería adoptar una estratificación socioeconómica de los SdS [32]:

1. El conjunto de organizaciones e individuos independientes que funcionan como parte del SdS y cómo se ven influenciados. Estos se recogen en la capa de “Instituciones y Economía” de la *herramienta de análisis del contexto*.
2. El entorno de infraestructura, representado por diversos fundamentos tecnológicos, instituciones, regulaciones, procedimientos y mecanismos a nivel del SdS. Se encuentra en la capa de “Dominio Social” de la *herramienta de análisis del contexto*.
3. Los sistemas de comunicación y de logística que proporcionan los procesos de interacción de los componentes organizacionales del sistema. Se encuentran en la capa de “Fenómenos Físicos, Procesos y Flujos” de la *herramienta de análisis del contexto*.
4. El Sistema de Innovación. El conjunto de actividades, cada una localizada en el espacio y en el tiempo, destinadas a adaptar el ecosistema a los cambios del entorno externo. Se encuentra en la capa de “Personas y Activos” de la *herramienta de análisis del contexto*.

¿Qué significa esto para el liderazgo de los SdS y para liderar la implementación de la evolución de SdS?

En primer lugar, la implementación debe comenzar con las organizaciones involucradas y el valor económico de los cambios. De la Tabla 5, *herramienta multinivel de planificación de la evolución de SdS*: «Describe en detalle las organizaciones actuales a nivel de SdS y SC, así como otras en el contexto de uso, con descripciones de su importancia».

Segundo paso hacia el Sistema de Innovación: *¿Qué Programas (a nivel de cambio en los SC), o qué Programas para producir nuevos SC se integrarán en este período? ¿Quién los produce?*». A partir de ahí, el líder debe avanzar hacia los métodos de liderazgo de la Tabla 3: *“¿Qué esfuerzo se debe dedicar para establecer y mantener un interés común viable entre los equipos de los SC, y cómo se lo comunicará a un conjunto amplio y, posiblemente remoto, de equipos responsables?”*.

En tercer lugar, evaluar o reevaluar los sistemas de comunicación y de logística. Aquí es donde el análisis de habilitadores y barreras de la Tabla 4 y la Tabla 5 resulta más útil. Como se indica en la Tabla 3: «Un líder debe crear un entorno donde se puedan establecer relaciones a nivel de SC y donde los responsables de las tareas puedan colaborar y desarrollar interdependencia en sus responsabilidades».

El entorno de infraestructura es una preocupación fundamental para el desarrollo de un nuevo SdS, pero es probable que ésta se mantenga estable en la evolución de un SdS en la fase de mantenimiento. Los cambios en el

entorno de infraestructura deben analizarse en las filas de “Tendencias y Eventos” de la Tabla 5 y, si son habilitadores o barreras para el cambio, deben anotarse en dichas filas.

5. COMPETENCIAS DE LIDERAZGO PARA LA GESTIÓN DE LOS SISTEMAS DE SISTEMAS

Cuando se crea un nuevo SdS, o uno existente experimenta una evolución significativa, el éxito dependerá de las cualidades del liderazgo. El marco para el liderazgo en megaproyectos y megasistemas puede aplicarse eficazmente en el contexto de la evolución de los SdS [33]. Cuatro capacidades esenciales del liderazgo son:

- Coordinación entre las diversas partes interesadas, para lograr resultados compartidos.
- Abordar la complejidad y la incertidumbre del sistema mediante el aprendizaje.
- Crear flexibilidad en las estrategias evolutivas.
- Gestionar el riesgo relacionado con las capacidades del SdS.

Las estrategias de gestión tienden a ser más transaccionales y se sustentan en procesos transaccionales como el ciclo “Plan-Do-Check-Act “ (PDCA)⁷. En los períodos de estabilidad, gran parte del cambio a nivel del SdS implica planificar, implementar, verificar los resultados y actuar según los hallazgos (*Modelo de Ondas*). En períodos evolutivos, el

7. Planificar-Hacer-Verificar-Actuar

liderazgo puede describirse como más transformacional. El modelo de liderazgo transformacional de Bass definió dos características transaccionales y cuatro transformacionales que contribuyen a una alta eficacia en el liderazgo [35]. La Figura 5 muestra cómo estas seis características se integran en un entorno de proyecto.

Mientras que el liderazgo transformacional motiva e inspira a los seguidores, el liderazgo transaccional se centra más en los “intercambios” entre el líder y los seguidores, en términos de tareas, penalizaciones y recompensas [35]. Este marco es importante para el liderazgo con una responsabilidad final del SdS, porque son necesarias habilidades de liderazgo tanto transaccionales (gestión de proyectos) como transformacionales (influencia y seguimiento). Las dos características transaccionales están más relacionadas con el modelo PDCA e incluyen: la **gestión por excepción** (gestionar comportamientos desalineados a nivel de los SC, planificar y llevar a cabo integraciones en el SdS, estabilizar las relaciones negociadas de SdS); y la **recompensa contingente** (fondos, incentivos). Las cuatro características transformacionales son más motivadoras, e incluyen: la **consideración individualizada** (coaching, delegación, oportunidades de formación), la **influencia idealizada** (modelizar comportamientos esperados, utilizar una comunicación carismática), el **compromiso inspirador** (ganar compromiso con el trabajo y el SdS, trabajo en equipo), y la **estimulación intelectual** (asegurarse de la necesidad del cambio, proporcionar una estrategia, construir apoyo interno, asegurar apoyo externo, proporcionar recursos si es necesario, institucionalizar los cambios) [35]. Se requieren aspectos transformacionales del liderazgo para motivar a los actores responsables de los SC a lograr los resultados al nivel del SdS, y a desempeñarse más allá de sus tareas esperadas de nivel de los SC.



Figura 6. Modelo de liderazgo transformacional de Bass [33].

En todos aquellos casos de estudio analizados por los autores, los períodos de creación y de evolución del SdS contaron con líderes con una sólida formación técnica en el ámbito de interés del SdS, además de notable experiencia en la resolución de situaciones difíciles en cadenas de suministro complejas. Entre las habilidades cruciales para los líderes de SdS, según [36], se incluyen:

- Son altamente abiertos a nuevas experiencias, auto disciplinados, participativos, estables y con un alto nivel de inteligencia emocional.
- Para ellos, la gestión de proyectos sigue siendo importante, pero en un contexto de cooperación y no solamente con métodos transaccionales (como suele ser habitual).

- Prefieren dedicar tiempo a la gestión del personal, la alineación de expectativas y la comunicación, en lugar de a los procesos de trabajo.
- Son muy conscientes de su propia capacidad de aprendizaje.

Aplicando este modelo a los *cambios en la práctica en la evolución de los SdS* de la Tabla 3, podemos caracterizar las habilidades de liderazgo de aquellos que son responsables de los resultados a nivel de los SdS:

Cambios en la práctica de la Evolución de SdS	Habilidades de liderazgo [33]
<p>Es posible que no exista una autoridad central para la evolución de los SdS. Habrá uno o más líderes responsables de los resultados de los SdS y que puedan tener distintos niveles de control o influencia sobre las tareas a nivel de los SC que conducen a dicha evolución. Las estrategias para crear y mantener las capacidades y los resultados de los SdS evolucionarán con la evolución de los SdS y los líderes responsables últimos ("accountables") pueden evolucionar con ellas. La dirección de la evolución de los SdS debe estar basada en una estrategia.</p>	<ul style="list-style-type: none"> • <i>Los líderes de SdS demuestran astucia política:</i> la capacidad de mostrar confianza y diplomacia profesional, a la vez que se relacionan eficazmente con personas de todos los niveles, tanto internos como externos. • <i>Los líderes de SdS anticipan situaciones futuras:</i> los líderes con visión de futuro pueden planificar mejor los resultados a largo plazo y adaptarse a entornos de proyectos en constante evolución. • <i>Los líderes de SdS aceptan la incertidumbre y la ambigüedad.</i> • <i>Los líderes de SdS son conscientes de sí mismos y automotivados:</i> reevalúan continuamente su propio juicio y decisiones, a la vez que muestran entusiasmo, pasión y un fuerte compromiso con los objetivos del SdS. • <i>Los líderes de SdS tienen una mentalidad abierta:</i> están dispuestos a revisar sus planes en base a nueva información para lograr mejores resultados del SdS. • <i>Los líderes de SdS son valientes:</i> están dispuestos a tomar decisiones difíciles, abordar el bajo rendimiento y desafiar expectativas o suposiciones poco realistas.
<p>El SdS debe construirse sobre intereses o preocupaciones compartidas. Una parte interesada con una posición de liderazgo debe dedicar esfuerzos a establecer y mantener un interés compartido. Los líderes deben identificar intereses comunes viables entre los equipos de los SC y comunicarlos a un conjunto amplio y, posiblemente remoto, de equipos responsables.</p>	<ul style="list-style-type: none"> • Se necesitan líderes con sólidas habilidades técnicas y del dominio en el SdS para generar un interés compartido. • Los líderes del SdS deben estar dispuestos a aprender junto con las partes interesadas responsables de los SC. • Los líderes del SdS deben ser fiables y establecer relaciones sólidas entre líder y seguidor, en línea con los intereses compartidos. • Se requiere pensamiento estratégico para establecer metas realistas y asegurar la alineación entre los objetivos del SdS y los de los SC. • Establecimiento de visión/objetivos: una visión clara es crucial para guiar la toma de decisiones y mantener la alineación con las partes interesadas. • Los líderes del SdS dominan todos los aspectos de la comunicación efectiva y la convierten en un eje central de su trabajo. La comunicación debe ser a un nivel superior y más visionario en el liderazgo del SdS.

Cambios en la práctica de la Evolución de SdS	Habilidades de liderazgo [33]
<p>Una persona con una posición de liderazgo en el SdS, ya sea responsable o autoridad (responsable último), debe crear un entorno que facilite el establecimiento de relaciones a nivel de los SC, y donde los responsables de tareas puedan colaborar y desarrollar interdependencia en sus responsabilidades. Esta interdependencia sirve para distribuir el poder entre los SC y debe ser un proceso activo durante la actualización o evolución de las capacidades del SdS. Esto incluye la colaboración con las partes interesadas con el rol de "consultados" en nombre del SC y mantener "informadas" a las demás partes interesadas.</p>	<ul style="list-style-type: none"> • Los líderes del SdS fomentan una cultura de apertura y comunicación y están abiertos a nuevas ideas y perspectivas. • Los líderes de SdS fomentan culturas de colaboración en los equipos de los SC. • Los líderes del SdS crean un entorno que demuestra activamente su convicción de que los responsables de las tareas actúan de buena fe y se esfuerzan al máximo para ayudar al SdS a alcanzar sus objetivos. • Los líderes de SdS capacitan y asesoran a otros, independientemente de la organización de los SC; ayudan a desarrollar las habilidades de liderazgo de los demás, fomentan la colaboración y mejoran la confianza y el rendimiento. • Los líderes de SdS construyen relaciones; crean y mantienen conexiones con las personas para crear un sentido de equipo.
<p>Los intereses compartidos son necesarios, pero insuficientes, para tener éxito en las tareas necesarias para la evolución de los SdS cuando las prioridades de los SC interfieren con las responsabilidades de éste. Los incentivos para el trabajo a nivel de SdS deben aplicarse y plasmarse en acuerdos documentados negociados entre los SC, ya sea mediante contratos formales u otras relaciones comerciales más indirectas. Estos incentivos pueden ser económicos, basados en la colaboración, definidos en las políticas, etc.</p>	<ul style="list-style-type: none"> • Habrá conflictos entre las partes interesadas y de prioridad dentro del equipo. Gestionar las partes interesadas requiere habilidades de negociación y diplomacia. Los líderes de SdS deben ser capaces de reunir a las partes interesadas en decisiones críticas a nivel del SdS. • Los SdS son complejos, involucran a una amplia variedad de partes interesadas y requieren la integración de múltiples disciplinas. Los líderes de SdS ayudan a resolver problemas complejos. Comprenden que el intercambio y la transferencia de conocimiento dentro y entre redes sociales es crucial y visualizan activamente el conocimiento para desarrollar un entendimiento compartido entre las partes interesadas. • Los líderes de SdS crean un entorno que fomenta y prioriza la adquisición de nuevos conocimientos y habilidades. El aprendizaje es recompensado. • Cuentan con amplia experiencia en la gestión de cadenas de suministro complejas en múltiples SC. • Poseen una gran perspicacia empresarial. Saben cuándo ser más directos, utilizando contratos explícitos con los SC para los requisitos de los SdS. Además, crean flexibilidad en los contratos y las relaciones comerciales que incentivan la resolución de problemas de integración por encima de simples factores de coste y plazo. • Aprovechan los efectos de red, utilizando eficazmente políticas, estándares o el interés económico común [22].

6. CONCLUSIÓN

La evolución de un SdS es un reto de liderazgo que abarca dos tipos de fenómenos: el surgimiento de un nuevo SdS y el mantenimiento de uno ya existente. Esto debe considerarse un proceso sociotécnico y socioeconómico, no sólo relacionado con nuevas capacidades derivadas de un proceso de ingeniería. Se ha hecho hincapié en que el análisis y la implementación de la evolución de un SdS deben comenzar con las organizaciones involucradas y el valor económico de los cambios. A partir de ahí, la evolución surgirá del *Sistema de Innovación*, independientemente del SdS de Interés y de la capacidad de los líderes para generar un interés compartido entre múltiples equipos (principalmente de ingeniería). Todo SdS debe ser liderado, aunque parte de este liderazgo sea informal. El líder debe crear un entorno donde se puedan establecer relaciones a nivel de los SC y donde los responsables de las tareas para la evolución de los sistemas puedan colaborar y ser interdependientes en sus responsabilidades. Finalmente, los líderes de un SdS deben ser holísticos y capaces de incorporar las tendencias y otros eventos emergentes que afectarán las capacidades o a los modelos de negocio del SdS en un plan de transición. Hay un conjunto único de habilidades de liderazgo asociadas con los SdS y las personas con estas habilidades tienden a surgir en el desarrollo de nuevos SdS y se debe confiar en ellas y apoyarlas.

REFERENCES

1. M. Maier, "Architecting Principles for Systems of Systems," Proc. of the Sixth Annual International Symposium, International Council on Systems Engineering, Boston, MA, (1996), pp. 567- 574.
2. International Standards Organization (ISO)/ Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 21840:2019(E) Systems and software engineering - Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems, 2019.
3. International Standards Organization (ISO)/ Institute of Electrical and Electronics Engineers (IEEE), ISO/IEC/IEEE 21839:2019(E) Systems and software engineering - System of systems (SoS) considerations in life cycle stages of a system, 2019.
4. M. Jamshidi, "System-of-Systems Engineering – a Definition," IEEE SMC (October 2005).
5. J. Dahmann. 2014. Systems of systems pain points. INCOSE International Symposium on Systems Engineering, 2014.
6. C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Comput. Surv. 48, (2), September 2015.
7. J.H. Holland (2006). "Studying Complex Adaptive Systems". Journal of Systems Science and Complexity. 19 (1): 1–8.
8. J. Boardman and B. Sauser, "System of systems - The meaning of 'of,'" In 2006 IEEE/SMC International Conference on System of Systems Engineering. IEEE.
9. McDermott, T. and Nadolski, M. 2017. "Emergence as Innovation in Systems of Systems - a Three Systems Model," 12th International Conference on System of Systems Engineering (SoSE), Waikoloa, US-HI.
10. R. Abbott, "Open at the top; open at the bottom; and continually (but slowly) evolving," In 2006 IEEE/SMC International Conference on System of Systems Engineering. IEEE.
11. Y. Bar-Yam, M. A. Allison, R. Batdorf, H. Chen, H. Generazio, H. Singh, and S. Tucker. 2004. The characteristics and emerging behaviors of system of systems. NECSI: Complex Physical, Biological and Social Systems Project, 1–16.
12. Rogers, Everett. Diffusion of Innovations, 5th Edition. Simon and Schuster. 16 August 2003.
13. R.M. Henderson, K.B. Clark, "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," Administrative Science Quarterly, Vol. 35, 1990. J. Dahmann, G. Rebovich, J. Lane, R. Lowry and K. Baldwin, "An implementers' view of systems engineering for systems of systems," 2011 IEEE International Systems Conference, Montreal, 2011, pp. 212-217.

14. J. Dahmann, G. Rebovich, J. Lane, R. Lowry and K. Baldwin, "An implementers' view of systems engineering for systems of systems," 2011 IEEE International Systems Conference, Montreal, 2011, pp. 212-217.
15. F.W. Geels, "Technological transitions as evolutionary reconfiguration processes: A multi-level perspective and a case-study," *Research Policy*, December 2002, 31 (8-9), 1257-1274.
16. J. Rotmans, R. Kemp, & M. van Asselt. "More evolution than revolution: transition management in public policy," *Foresight* 3 (1).
17. A. Mostafavi, D. M. Abraham, D. DeLaurentis, & J. Sinfield, "Exploring the Dimensions of Systems of Innovation Analysis: A System of Systems Framework," *IEEE Systems Journal*, 2011, 5 (2), pp. 256-265.
18. State of California, Senate Bill No. 1298, Chapter 570, Approved by Governor on September 25, 2012. Retrieved 4/2025 from http://www.leginfo.ca.gov/pub/11-12/bill/sen/sb_1251-1300/sb_1298_bill_20120925_chaptered.pdf.
19. H. Lawson, *A Journey Through the Systems Landscape*. London: College Publications, 2010.
20. Gary T. Anderson Opportunities for Innovation in a System of Systems Framework, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA - October 2009.
21. Sage, Andrew P. and Cuppan, Christopher D. 'On the Systems Engineering and Management of Systems of Systems and Federations of Systems '. 1 Jan. 2001 : 325 – 345. Print.
22. Waymo. Wikipedia. Retrieved 4/2025 from <https://en.wikipedia.org/wiki/Waymo>.
23. Mohamed Hichem Fendali, Isabelle Borne, Djamel Meslati, Labiba Souici-Meslati. "How to steer evolution of a system-of-systems" *Systems Engineering*. 2024;1–17. ©2024Wiley Periodicals LLC.
24. Project Management Institute (PMI). *A Guide to the Project Management Body of Knowledge (PMBOK Guide) (2004 edition)*. Newtown Square, PA: Project Management Institute.
25. Smith, Michael L.; Erwin, James. "Role & Responsibility Charting (RACI)" (PDF). Project Management Institute California Inland Empire Chapter. p. 5. Archived from the original on 16 September 2022. Retrieved 18 May 2023.
26. Rouse, William B (Ed.). *Enterprise Transformation: Understanding and Enabling Fundamental Change*. Wiley 2006.
27. J. Lane, Planning for SoS Evolution, presentation to the San Diego Chapter of INCOSE 23 July 2016. Retrieved 4/2025 from <https://sdincose.org/wp-content/uploads/2016/10/SD-INCOSE-Tutorial-Planning-for-SoS-Evolution.pdf>
28. W. B. Rouse, "Policy Flight Simulators For Transforming Large-Scale Public-Private Enterprises;" Third International Engineering Systems Symposium CESUN 2012; 18-20 June 2012.W. B. Rouse & D. Bodner, Multi-level modeling of complex socio-technical systems – phase 1, A013 - final technical report, SERC-2013-TR-020-2, Systems Engineering Research Center.
29. W. B. Rouse & D. Bodner, Multi-level modeling of complex socio-technical systems – phase 1, A013 - final technical report, SERC-2013-TR-020-2, Systems Engineering Research Center.
30. McDermott, Tom. "Applying Systems-of-Systems Principles to Purposeful Design of Human Systems." 2018 13th Annual Conference on System of Systems Engineering (SoSE) (2018): 150-156.
31. Kemp, Loorbach, & Rotmans; "Transition Management as a model for managing processes of co-evolution towards sustainable development;" *International Journal of Sustainable Development & World Ecology* 14(2007) 1-15.
32. Kleiner, G. "Socio-economic ecosystems in the light of system paradigm." *System analysis in economics-2018*, pp. 4-12. 2018.
33. T. McDermott & N. Hutchison, Critical skills for successful leadership of large complex projects, preprint, 22nd Annual Acquisition Research Symposium & Innovation Summit, 2025.
34. Aarons, G.A. Transformational and transactional leadership: association with attitudes toward evidence-based practice. *Psychiatr Serv*. Aug 2006; 57(8):1162-9.
35. Van Wart, M. Evaluating Transformational Leaders: The Challenging Case of Eric Shinseki and the U.S. Department of Veterans Affairs. *Public Administration Review*, 75(5): 760-769. September|October 2015.
36. Mellow, E., & Nandurdikar, N. *Leading Complex Projects*. John Wiley & Sons. 2018.

BIOGRAFÍAS

TOM MCDERMOTT

Tom McDermott es el Director de Tecnología en el Systems Engineering Research Center (SERC) y profesor de la School of Systems and Enterprises en el Stevens Institute of Technology, en Hoboken, Nueva Jersey. En el SERC, desarrolla nuevas estrategias de investigación y lidera la investigación sobre transformación digital, educación, seguridad y aplicaciones de inteligencia artificial. Anteriormente, ocupó cargos como profesor y director de investigación en el Georgia Tech Research Institute, así como director y gerente de productos integrados en el equipo de aviónica del F-22 Raptor, en Lockheed Martin. El Sr. McDermott imparte clases de arquitectura de sistemas, pensamiento crítico y de sistemas, y liderazgo. Ofrece consultoría a nivel ejecutivo como experto en ingeniería de sistemas y estrategia organizacional. Es fellow del International Council on Systems Engineering (INCOSE) y, recientemente, cumplió tres años como Director de Integración Estratégica de INCOSE.



MIGUEL ÁNGEL COLL MATAMALAS

Miguel Ángel Coll Matamalas es Ingeniero de Sistemas en Isdefe, y trabaja como consultor en la Jefatura de Apoyo Logístico de la Armada en el desarrollo de doctrina logística y actividades relacionadas con la obtención de Apoyo Logístico Integrado para los Programas de Obtención en curso. En particular, colabora en el desarrollo del Gemelo Digital de las fragatas F-110. También participa en el Grupo de Trabajo de la OTAN responsable de revisar la norma ALP-10 (NATO Guidance for Integrated Life Cycle Support). Miguel Ángel es Ingeniero Naval y Oceánico por la Universidad Politécnica de Madrid y ha cursado un Executive MBA en el IESE (Universidad de Navarra). En su etapa profesional anterior, ha ocupado diversos cargos relacionados con el mantenimiento y la gestión de activos, tanto en el sector marítimo como en infraestructuras del sector del agua. Con el objetivo de aplicar esta experiencia previa en diferentes áreas del Sosténimiento, Miguel Ángel trabaja en la mejora de la soportabilidad de los sistemas mediante la influencia en el diseño, en el desarrollo de gemelos digitales, del mantenimiento prescriptivo y del concepto de Apoyo Logístico Basado en Modelos (MBPS).





Nuevos métodos de análisis, modelado y simulación para SdS

José Luis de Rosario, *Isdefe (jlderosario@isdefe.es)*

Dr. Paul Grogan, *Arizona State University (paul.grogan@asu.edu)*

Dr. Alessandro Golkar, *Technical University of Munich (golkar@tum.de)*

Dr. Amro Farid, *Stevens Institute of Technology (amro.farid@stevens.edu)*

Dr. Young-Jun Son, *Purdue University (yjson@purdue.edu)*

Dr. Nil Egin, *The Pennsylvania State University (nhe2@psu.edu)*

Dr. John Little, *Virginia Tech (jcl@vt.edu)*

Resumen

Este capítulo presenta una visión general de métodos de análisis y simulación novedosos que son útiles para la ingeniería de SdS, incluyendo: la coordinación de sistemas, la federación de sistemas, el modelado heterofuncional, el modelado y la simulación de elementos federados, el modelado basado en agentes y enfoques de SdS para el modelado.

Palabras clave

Coordinación de sistemas, federación de sistemas, modelado heterofuncional, modelado y simulación de elementos federados, modelado basado en agentes.



1. INTRODUCCIÓN

Como se ha comentado a lo largo del libro, la ingeniería de Sistemas de Sistemas (SdS) presenta un conjunto de desafíos que trascienden a los afrontados en la ingeniería de sistemas tradicionales. Los SdS se caracterizan por la independencia operativa y de gestión de sus sistemas constituyentes, el desarrollo evolutivo, el comportamiento emergente y, a menudo, la heterogeneidad en la función, el propósito y el diseño. Estas características requieren métodos de análisis y simulación que puedan dar cabida a interacciones dinámicas, a un control descentralizado y a múltiples capas de abstracción.

Este capítulo presenta una visión general de métodos de análisis y simulación novedosos que han surgido o madurado en respuesta a las necesidades específicas de la ingeniería de SdS. Estos métodos proporcionan a los ingenieros herramientas para representar, razonar y probar conceptos de los SdS de maneras que los enfoques tradicionales no pueden apoyar eficazmente. El capítulo cubre enfoques que facilitan la coordinación de sistemas para lograr objetivos compartidos y aquellos que apoyan la federación de sistemas mientras se preserva la autonomía local. Se exploran técnicas de modelado heterofuncional que tienen en cuenta la diversidad de roles e interacciones entre sistemas y se presentan estrategias de modelado y simulación de elementos federados que permiten que modelos independientes interoperen de forma coherente. El capítulo también cubre el modelado basado en agentes, un método poderoso para capturar el comportamiento de entidades autónomas y sus interacciones, y enfoques de modelado específicos de SdS que proporcionan abstracciones y formalismos adaptados para capturar la complejidad de creaciones de ingeniería de gran envergadura.

Al presentar estos métodos, este capítulo tiene como objetivo proporcionar a los ingenieros de SdS una hoja de ruta para seleccionar, combinar y aplicar técnicas adecuadas para el análisis y la simulación a lo largo del ciclo de vida de un SdS.

2. COORDINACIÓN DE SISTEMAS

2.1. Introducción

La ingeniería de sistemas es un enfoque tradicionalmente gestionado de forma centralizada para el diseño, integración, prueba, operación y retiro con éxito de sistemas ingenieriles. Los enfoques centralizados ayudan a abordar los objetivos a nivel de sistema al dirigir los esfuerzos de ingeniería para considerar propiedades emergentes que pueden no estar dentro del alcance de los equipos de subsistemas individuales. Resulta característico el reciente interés en los enfoques de

ingeniería de sistemas basados en modelos (MBSE) que enfatizan los sistemas de información centralizados para recopilar, gestionar y compartir conocimientos de diseño a través de un equipo de diseño [1].

Los SdS desafían los conceptos convencionales de control en la ingeniería de sistemas porque ningún actor a nivel de sistema posee autoridad completa sobre los demás. Los problemas de la ingeniería de SdS exhiben estructuras más descentralizadas que influyen en el tipo y la naturaleza de los métodos empleados. El trascendente trabajo de Maier describe los principios de arquitectura para SdS, referidos con más precisión como sistemas colaborativos, que enfatizan un control más indirecto a través de formas intermedias estables, el triaje de políticas, el aprovechamiento en las interfaces y los incentivos para asegurar la cooperación [2].

El tipo y grado de interacción dentro de un SdS abarca un continuo de esfuerzos colectivos [3]. La coordinación es un paradigma de planificación y compartición de información débilmente acoplado que persigue objetivos individuales reduciendo el conflicto. La cooperación es un paradigma de compartición de recursos moderadamente acoplado que persigue objetivos mutuos con resultados compartidos. Finalmente, la colaboración es un paradigma fuertemente acoplado que persigue un objetivo conjunto que los individuos son incapaces de lograr por sí solos. Los avances en las técnicas de modelado y análisis de SdS buscan promover acciones de diseño que conduzcan a resultados colectivos mutuamente deseables, mientras se mitigan los indeseables.

2.2. Modelado de la coordinación con Teoría de Juegos

La teoría de juegos es un campo económico que estudia la interacción estratégica. Se basa en modelos y métodos clave de la teoría de decisiones, que estudia la toma de decisiones individual bajo incertidumbre. La distinción clave entre la teoría de juegos y la teoría de decisiones radica en la consideración de procesos de toma de decisiones interactivos entre actores que consideran iterativamente la perspectiva de los demás sobre un problema acoplado al diseñar estrategias de toma de decisiones. La teoría de juegos es particularmente aplicable a los problemas de SdS porque considera explícitamente el control descentralizado sobre las acciones y los múltiples objetivos de unos actores que potencialmente pueden competir.

Los constructos fundamentales de la teoría de juegos definen a los actores, sus acciones disponibles y los resultados compartidos obtenidos. Cada actor posee una medida de preferencia basada en la utilidad de los resultados. Las preferencias siguen una lógica similar a las funciones

objetivo en el diseño basado en el valor [4]; sin embargo, los resultados acoplados evaluados por cada función de utilidad específica del actor pueden llevar a preferencias conflictivas sobre los resultados.

La estrategia de un actor describe el proceso completo mediante el cual selecciona una acción. Las estrategias puras seleccionan una acción con certeza (por tanto, las acciones también se denominan comúnmente estrategias) y las estrategias mixtas seleccionan combinaciones probabilísticas de acciones. Para identificar estrategias racionales (es decir, aquellas que conducen a resultados preferidos), la teoría de juegos considera explícitamente la interacción estratégica entre los actores, en lugar de simplemente tratar las acciones de los demás como fuentes de incertidumbre exógena.

Existen una variedad de formulaciones de problemas teóricos de juegos y métodos de análisis; sin embargo, esta sección presenta uno básico: un juego en forma normal. Se representa mediante una matriz que cuantifica la preferencia por los resultados (recompensa) para cada actor como resultado de combinaciones de acciones simultáneas, a menudo sobre espacios de acción discretos. Los juegos en forma normal suelen modelar problemas de decisión de una sola etapa; sin embargo, los juegos repetidos también pueden estudiar la estrategia de decisiones secuenciales. Los juegos no cooperativos se centran en la dinámica de toma de decisiones de base que surge de los valores de las recompensas, mientras que los juegos cooperativos permiten acuerdos dentro del juego para modificar dichos valores.

Los métodos de análisis de la teoría de juegos se centran en identificar estrategias de equilibrio. Un equilibrio de Nash es un conjunto de estrategias donde ningún actor tiene un incentivo individual para desviarse a una estrategia diferente. Representa una posible “solución” a un juego porque encuentra un punto de decisión naturalmente estable que es inmune a los efectos interactivos. Los juegos pueden tener cero, uno o muchos equilibrios y los actores pueden tener diferentes preferencias para cada equilibrio. La presencia de un equilibrio no sugiere que sea mutuamente deseable para todos los actores.

En lugar de resolver problemas realistas del día a día, la teoría de juegos se esfuerza por anticipar el comportamiento estratégico entendiendo las fuerzas subyacentes que moldean las acciones en entornos más abstractos. Muchos problemas de teoría de juegos consideran un pequeño conjunto de actores y sus acciones. Una sorprendente variedad de dinámicas estratégicas emergen de juegos con dos actores y dos acciones basadas en el número y la naturaleza de las estrategias de equilibrio presentes. Por ejemplo, el famoso juego del dilema del prisionero es un juego de dos actores, cooperar o traicionar, que exhibe

una estrategia de equilibrio (ambos traicionan) que no conduce al resultado más deseable (ambos cooperan). La tensión entre los beneficios individuales y colectivos ayuda a comprender las dinámicas estratégicas detrás de muchos esfuerzos colectivos.

Además de identificar estrategias de equilibrio, los métodos de teoría de juegos también proporcionan análisis comparativos de estrategias dentro o entre juegos relacionados. Por ejemplo, una caza de ciervos es un juego de dos actores, colaborar o traicionar, que exhibe una dinámica de biestabilidad donde las dos estrategias de equilibrio (ambos colaboran o ambos traicionan) intercambian fuentes de riesgo y recompensa. El análisis de la teoría de juegos sobre el predominio del riesgo evalúa el atractivo comparativo de las dos estrategias midiendo la estabilidad relativa de la colaboración, lo que ayuda a identificar y perseguir oportunidades de colaboración robustas [5].

Las aplicaciones más amplias del diseño de mecanismos aprovechan el análisis de la teoría de juegos para proponer cambios estructurales en el juego que se anticipan dando lugar a resultados más deseables colectivamente. En este contexto, la teoría de juegos cooperativos permite alcanzar acuerdos aplicables dentro de coaliciones de actores cambiando el enfoque basado en maximizar las recompensas individuales por otro orientado a compartir las recompensas colectivas. Como en la mayoría de los problemas de diseño, el diseño de mecanismos opera dentro de un vasto espacio de soluciones de diseño creativo que equilibran factores que compiten entre sí, como la eficiencia económica y la equidad.

2.3. Ejemplo ilustrativo: coordinación de satélites

Se considera el ejemplo de un problema de coordinación de SdS con dos agencias espaciales independientes: Red y Blue. Red opera un pequeño satélite de investigación y Blue opera un gran satélite meteorológico. Como ejemplo de coordinación bilateral, las agencias deben decidir sobre acciones para mitigar la amenaza de una colisión en respuesta a un evento de conjunción peligrosa.

Ambos actores tienen dos acciones disponibles: no hacer nada o maniobrar su satélite, lo que lleva a cuatro posibles resultados. La Tabla 1 ilustra la representación de este juego de coordinación utilizando valores teóricos de utilidad (beneficio). Cabe destacar que la estrategia mixta se ha añadido como una fila y una columna adicional para facilitar la exposición. Las utilidades se cuantifican para cada resultado, que a su vez puede componerse de una distribución de eventos (por ejemplo, el resultado donde ambos actores no hacen nada puede llevar probabilísticamente a una colisión). Las unidades financieras (por ejemplo, USD \$M) proporcionan un indicador relativamente imparcial para

		Agencia Blue		4% No maniobrar 96% Maniobrar	
		No maniobrar	Maniobrar		
Agencia Red	No maniobrar	-50	0	-2	-6.88
	Maniobrar	-2	-2	-2	-2.88
3% No maniobrar, 97% Maniobrar		-3.44	-1.94	-2	-3

Tabla 1. Problema de coordinación de colisiones de satélites (Utilidades en USD \$M).

valorar la utilidad a pesar de verse afectadas por actitudes humanas variables frente al riesgo. Para simplificar, este ejemplo asume una toma de decisiones neutral al riesgo. El resultado donde ambos actores no hacen nada da lugar a consecuencias negativas severas, representadas como -\$50M para Red y -\$100M para Blue. Los otros resultados dan lugar a consecuencias ligeramente negativas debido al consumo de propulsor que reduce la vida operativa del satélite, representadas como -\$2M para Red y -\$3M para Blue. Los valores de utilidad generalmente no son comparables entre actores debido a las diferencias en sus preferencias; sin embargo, las cantidades seleccionadas están destinadas a resaltar las asimetrías entre los dos actores.

El análisis de equilibrio busca conjuntos de estrategias estables, revelando dos equilibrios puros y uno mixto. La primera estrategia de equilibrio puro ocurre si Red elige no hacer nada y Blue elige maniobrar. En este caso, Red no elegiría unilateralmente maniobrar, porque la amenaza ya está mitigada. De manera similar, Blue no elegiría unilateralmente no hacer nada, porque da lugar a un peor resultado que proceder con una maniobra. La segunda estrategia de equilibrio puro ocurre en un escenario similar donde Red elige maniobrar y Blue elige no hacer nada con la misma justificación. El equilibrio mixto produce indiferencia en la selección de estrategias. En este juego, cuando Red elige maniobrar con una probabilidad de $(-100+3) / (-100-0+3-3) = 0.97$, Blue tiene una utilidad esperada de -\$2M para todos los resultados. Cuando Blue elige maniobrar con una probabilidad de $(-50-0) / (-50+2-0-2) = 0.96$, Red tiene una utilidad esperada de -\$3M para todos los resultados.

Los actores no tienen la misma preferencia por las tres soluciones de equilibrio. Red prefiere el equilibrio donde Blue siempre maniobra. De manera similar, Blue prefiere el equilibrio donde Red siempre maniobra. Ninguno de los actores prefiere el equilibrio mixto. La teoría de juegos no proporciona mucho más progreso hacia una solución y pueden ser necesarios factores externos como la negociación o la confrontación para resolver la tensión estratégica. De hecho, este ejemplo es una instancia de juegos descritos de diversas maneras como el de la gallina, el del halcón y la paloma o el de la trampa de la nieve, que representan un modelo teórico de conflicto intrínseco.

La teoría de juegos cooperativos permite la aplicación de acuerdos que impactan materialmente sobre los resultados. En el problema anterior, es comparativamente menos costoso para Red maniobrar su pequeño satélite que para Blue; sin embargo, Red asume el costo de maniobrar su propio satélite. El diseño de mecanismos propone incentivos para promover soluciones eficientes. Por ejemplo, considera un mecanismo donde Blue acuerda compensar a Red por maniobrar su satélite más pequeño para mitigar una posible colisión, independientemente de la acción de Blue. La Tabla 2 representa el intercambio de valor como una deducción de Y para Blue y una adición de X para Red. Cabe destacar que X e Y pueden no ser iguales si Red y Blue valoran de manera diferente el recurso que se intercambia o si la fricción transaccional erosiona parte del valor.

		Agencia Blue	
		No maniobrar	Maniobrar
Agencia Red	No maniobrar	-50 -100	0 -3
	Maniobrar	X-2 0-Y	X-2 -3-Y

Tabla 2. Problema de coordinación de colisiones de satélites con incentivo.

Para valores $X > \$2M$, sólo queda una estrategia de equilibrio donde Red siempre elige maniobrar y Blue siempre elige no hacer nada. En comparación con el juego original, esta solución es mejor que el mejor equilibrio de Red y mejor que el peor equilibrio de Blue para valores $Y < \$3M$. Además, esta solución evita la incertidumbre estratégica asociada con múltiples equilibrios. Sin embargo, surge un nuevo problema potencial: Red ahora tiene un incentivo perverso para beneficiarse de los peligros de colisión como una fuente de ingresos adicionales. Se requerirá una ampliación del alcance del sistema y un análisis adicional para diseñar soluciones coordinadas del SdS que resistan la explotación.

3. FEDERACIÓN DE SISTEMAS

3.1. Introducción

Las Federaciones de Sistemas (FdS) son una clase de SdS que pueden operar durante su ciclo de vida de forma independiente o pueden cooperar voluntariamente entre sí con la intención de obtener beneficios mutuos. A diferencia de los sistemas jerárquicos o los estrechamente integrados, las federaciones son una clase de sistemas distribuidos que mantienen la independencia en la gestión, operación y objetivos de los sistemas que los componen. Al mismo tiempo, están diseñadas de tal manera que promueven la colaboración entre ellos. El paradigma de FdS es particularmente relevante en escenarios donde el intercambio de recursos puede mejorar la generación de valor mientras se hacen frente a los desafíos de coste y complejidad, especialmente para sistemas caracterizados por altos costes de capital o altos costes operativos recurrentes.

Las características que definen a las FdS se describen en la Tabla 3. La motivación detrás de las FdS radica en el potencial para mejorar la utilización de los recursos y la posibilidad de proporcionar valor añadido promoviendo la colaboración entre sistemas tradicionalmente aislados. La colaboración entre sistemas es particularmente importante para aprovechar las oportunidades que surgen de la subutilización de recursos.

Definición de las características de las federaciones de sistemas			
<p>Cooperación Voluntaria: Participación basada en un acuerdo mutuo, incentivada por los objetivos individuales de los sistemas involucrados.</p>	<p>Beneficios mutuos: Federaciones motivadas por la aparición de beneficios mutuos derivados de la colaboración. Ejemplos de beneficios mutuos son el mejor rendimiento, la reducción de costes o la capacidad mejorada.</p>	<p>Gestión independiente: Limitar la colaboración entre sistemas a áreas de interés mutuo, dejando a cada sistema con su propia autonomía.</p>	<p>Alineamiento objetivos operacionales: Maximizar los objetivos grupales de la federación, sujeto a la independencia operativa y la búsqueda de objetivos individuales.</p>

Tabla 3. Características de las Federaciones de Sistemas (FdS).

Los SdS emplean mecanismos estructurados para garantizar la integración cuando sea necesario. Dicha integración se implementa a través de protocolos de comunicación centralizados. También son posibles los marcos de gobernanza para permitir colaboraciones flexibles, a menudo de naturaleza oportunista.

El paradigma de los Sistemas Satelitales Federados (FSS, por sus siglas en inglés) [7] representa un ejemplo de federaciones de sistemas. Los satélites en los FSS cooperan para explotar recursos que de otro modo quedarían sin usar porque no se necesitan en ciertos puntos de su vida operativa. El ancho de banda, el almacenamiento de datos y la capacidad computacional son ejemplos de recursos que se pueden compartir en una federación. La participación en una federación permite reducir las ineficiencias operativas y, por tanto, hacer un mejor uso de la capacidad en órbita, un objetivo igualmente importante para un espacio sostenible y responsable. Uno de los problemas que surgen al implementar la idea de federaciones en órbita es el de los mecanismos necesarios para sus operaciones. En otras palabras, es necesario definir acuerdos operativos e incentivos de naturaleza económica o de utilidad que permitan a los satélites operar de forma eficiente mientras mantienen su autonomía. Uno de los mecanismos considerados para este propósito es el de las subastas inversas con ofertas restringidas [8]. Este enfoque permite la creación de un ecosistema comercial de activos espaciales en órbita, generando así oportunidades para la innovación y la mejora de las capacidades disponibles para los satélites individuales.

El estudio de las federaciones de sistemas es de naturaleza multidisciplinar y no se limita únicamente al sector espacial. Se pueden proponer fácilmente otros ejemplos de posibles federaciones en diferentes campos industriales. Por ejemplo, en el sector energético, es posible concebir federaciones de sistemas energéticos como turbinas eólicas y parques solares, así como unidades de almacenamiento de baterías. En este caso, la electricidad y su almacenamiento son los recursos disponibles. El objetivo de la federación se convierte así en poner a disposición el exceso de energía, promoviendo el equilibrio de la red, así como la dinámica de precios entre la oferta y la demanda. Otros impactos de este enfoque incluyen la posible reducción de los costes operativos y la minimización del impacto ambiental a través de la utilización eficiente en términos de almacenamiento, transporte y distribución de energía. Un área adicional de interés para las federaciones es el sector de las telecomunicaciones. En este último, las redes inalámbricas federadas pueden permitir a los proveedores de servicios independientes compartir sus recursos, como estaciones base y ancho de banda, para, por ejemplo, expandir la cobertura y mejorar la fiabilidad del servicio.

La motivación detrás de la idea de las federaciones de

sistemas es la necesidad de eficiencias en el uso de recursos frente a las limitaciones de la complejidad técnica y económica que a menudo son difíciles para sistemas complejos como los de industrias con altos costes de capital. A través de la promoción de la colaboración entre entidades independientes, las federaciones de sistemas permiten aumentar las capacidades de los sistemas individuales a un coste marginal más bajo de lo que sería posible en ausencia de oportunidades de colaboración. Este enfoque también permite la reducción de los costes operativos y el equilibrio de la utilización de recursos cuando se diseña dentro de una red de intercambio. En este capítulo se exploran los principios básicos, los desafíos y las aplicaciones prácticas de las federaciones de sistemas. Se utilizarán los sistemas satelitales federados como un caso de estudio de FdS. El objetivo del capítulo es proporcionar una exposición integral sobre el potencial de las federaciones de sistemas como un enfoque viable para optimizar la eficiencia de los sistemas que operan en redes, así como un enfoque para promover la escalabilidad y la sostenibilidad en la utilización de recursos.

3.2. Conceptos fundamentales de federaciones

El concepto de sinergia es fundamental para la propuesta de valor de las federaciones de sistemas. En ese contexto, la sinergia se define como el beneficio agregado resultante de la cooperación de sistemas que operan dentro de una federación. La sinergia se define como el beneficio mutuo resultante de la cooperación de sistemas independientes. Esta cooperación permite que los sistemas participantes logren colectivamente mayores resultados de los que podrían alcanzar en aislamiento. Matemáticamente, la sinergia se define de la siguiente manera: La exploración del espacio de intercambio es un enfoque

$$S_{y_{N \leftrightarrow N}} = \sum_i^N \Delta C_i^j \quad (1)$$

$$\Delta C_i^j = F_i(U^j) - C^j$$

Donde en (1), $S_{y_{N \leftrightarrow N}}$ es la sinergia entre un conjunto de N sistemas, y ΔC_i^j es el delta de coste de una arquitectura específica j para un sistema dado i. ΔC_i^j se cuantifica como la diferencia entre la frontera de Pareto del espacio de intercambio del sistema representada por el mapeo de coste-utilidad $F_i(U^j)$, y el coste de la arquitectura C^j . La sinergia positiva ocurre cuando la suma de las funciones delta de coste es positiva, lo que significa que se ha logrado una mejora neta en el rendimiento coste-beneficio en toda la federación y se trata como un heurístico para guiar las decisiones sobre la arquitectura. La sinergia surge de cambios correlacionados en los espacios de diseño de sistemas federados, como se ve en sus compensaciones de coste-utilidad [4]. En los sistemas satelitales federados (FSS), compartir ancho de banda o capacidades de retransmisión mejora el rendimiento de las misiones de observación de la Tierra mientras se reducen los costes. Las mejoras en la utilidad local de un sistema afectan directamente el rendimiento coste-beneficio de otros, formando una red de interdependencias [10].

establecido utilizado para el análisis de rendimiento de las federaciones de sistemas. Este enfoque metodológico evalúa el espacio de posibles soluciones para identificar arquitecturas de sistemas con compensaciones óptimas entre coste, rendimiento y otras variables de decisión de interés [11, 12]. La exploración del espacio de intercambio permite que los sistemas que forman parte de (o son candidatos para formar parte de) una federación exploren el valor añadido por su participación en el conjunto de recursos agrupados. Por ejemplo, en los sistemas satelitales federados (FSS), el análisis del espacio de intercambio se utiliza para evaluar la compartición coste-beneficio de ancho de banda y capacidades de procesamiento de datos. Este enfoque puramente analítico permite una evaluación cuantitativa de las condiciones bajo las cuales las federaciones son más beneficiosas e identifica escenarios en los que pueden no proporcionar los beneficios esperados.

No en todos los casos se pueden considerar beneficiosas las federaciones de sistemas. De hecho, también presentan desafíos únicos a nivel de arquitectura de sistemas. Uno de los principales desafíos es la interoperabilidad entre los sistemas independientes que conforman la federación [13]. La interoperabilidad requiere algoritmos, hardware y software, cuya complejidad puede superar los posibles beneficios de la colaboración. Ejemplos de elementos necesarios para la interoperabilidad incluyen protocolos de comunicación, así como la estandarización de formatos de datos e interfaces. La estandarización en sí misma es un proceso costoso y complejo de coordinación entre entidades heterogéneas. Además, la naturaleza descentralizada de las FdS complica

la creación de estándares universales, ya que los sistemas individuales pueden priorizar la optimización local sobre la compatibilidad global.

Otro desafío arquitectónico es evaluar el potencial de escalabilidad de los sistemas federados. El número de posibles interacciones y dependencias entre los sistemas federados aumenta exponencialmente a medida que aumenta el número de sistemas que participan en la federación. Por este motivo, ofrecer garantías de rendimiento o fiabilidad se vuelve particularmente desafiante, especialmente frente a las incertidumbres operativas que enfrentan dichos sistemas. Una consideración crítica en este sentido es el equilibrio entre los costes de crear y mantener interfaces y los beneficios esperados de la cooperación. La gestión de interfaces que permiten la interacción de sistemas federados es un tema clave que se ha abordado en el contexto de los sistemas satelitales federados [9]. Las dos estrategias principales consideradas son cambios directos, como reemplazar o agregar interfaces, y cambios indirectos, que involucran middleware o nodos negociadores [14].

Las modificaciones directas de interfaces a menudo requieren acceso físico a los satélites, lo cual es muy poco práctico para las naves espaciales que no están específicamente diseñadas para el mantenimiento en órbita [15]. Este último problema ha suscitado un interés particular en los últimos años, especialmente en el contexto de enfoques modulares para el diseño de sistemas espaciales [16]. El desafío del mantenimiento en órbita de satélites no es nuevo y se ha abordado varias veces durante la evolución de

la industria espacial. Un ejemplo es el Telescopio Espacial Hubble, que fue diseñado con componentes modulares para permitir misiones de mantenimiento regulares. El diseño del telescopio permitió a los astronautas reparar defectos, como la aberración esférica inicial del espejo primario, y actualizar instrumentos y subsistemas, extendiendo significativamente su vida operativa y utilidad científica [17]. Sin embargo, tradicionalmente el mantenimiento en órbita ha implicado costes significativos que sólo se justifican para misiones espaciales emblemáticas del calibre de Hubble y, por tanto, no son asequibles para la mayoría de las misiones espaciales. Esto puede cambiar en un futuro cercano gracias a la reducción de los costes de lanzamiento y los avances en la tecnología de servicio en órbita [18]. Un desafío adicional para la interfaz de satélites son las restricciones regulatorias y la heterogeneidad de los protocolos de comunicación entre satélites. Se necesita una evaluación cuidadosa de estos desafíos para determinar si los costes y riesgos de adaptar satélites existentes superan los beneficios esperados.

Las modificaciones indirectas, como el uso de nodos negociadores, ofrecen un enfoque más factible para habilitar la interoperabilidad [14]. Los negociadores actúan como intermediarios que facilitan el intercambio de recursos y la comunicación entre satélites, como a través del uso de tecnologías de radio definida por software (SDR). El trabajo presentado en [14] evalúa el concepto de cargas útiles alojadas o satélites negociadores independientes equipados con antenas reconfigurables y SDR. Estas tecnologías digitales permiten la adaptación dinámica a diferentes protocolos de comunicación y frecuencias, abordando así la heterogeneidad de los sistemas en una federación. Sin embargo, incluso con este enfoque, la compensación entre la complejidad adicional introducida por los negociadores y los beneficios de una mayor colaboración debe analizarse cuidadosamente.

El caso de los sistemas satelitales federados muestra que, dadas las condiciones operativas y económicas adecuadas, la integración de interfaces puede desbloquear sinergias significativas, permitiendo que los sistemas logren objetivos colectivos que superen sus capacidades individuales [7].

3.3. Desafíos y oportunidades en las federaciones de sistemas

Los sistemas de ingeniería federados enfrentan desafíos derivados de su naturaleza descentralizada y la independencia inherente de los sistemas constituyentes. Estos desafíos incluyen la confianza y la seguridad. Estos son particularmente críticos ya que afectan la disposición de las partes interesadas a participar y mantener la federación. Los problemas de confianza en las FdS provienen de problemas con la autenticación de datos, la integridad y la

confidencialidad. Sin mecanismos robustos para abordar estos problemas, el riesgo de manipulación de datos, acceso no autorizado y desinformación aumenta, lo que podría socavar la colaboración necesaria para una federación exitosa.

Una solución propuesta para estos desafíos es la adopción de sistemas blockchain [19]. Los registros blockchain junto con el cifrado de datos son una posible solución para gestionar el intercambio de datos y metadatos, a través de mecanismos transparentes que puedan garantizar la privacidad de los datos cuando sea necesario. Otra característica útil de blockchain en este contexto es su naturaleza a prueba de manipulaciones que permite proporcionar pruebas sobre las fuentes de datos y las transacciones. Como resultado, las partes interesadas tienen garantías de confianza y, en el contexto de los sistemas federados, blockchain puede utilizarse para registrar de manera segura las transacciones de intercambio de recursos, como el ancho de banda o los intercambios de procesamiento de datos, asegurando que todas las partes tengan un registro verificable de los acuerdos y las entregas.

Un enfoque complementario a blockchain son los protocolos de Infraestructura de Clave Pública (PKI). PKI permite una comunicación segura aprovechando las claves criptográficas para la autenticación y el cifrado. En el contexto de las federaciones, este enfoque permite establecer una base de confianza segura al permitir que las partes interesadas verifiquen la identidad de las partes que se comunican y aseguren la integridad de los datos compartidos. Por ejemplo, en los sistemas satelitales federados, PKI puede facilitar la comunicación segura entre satélites al garantizar que solo las partes autenticadas participen en las operaciones de intercambio de recursos y que los datos intercambiados permanezcan confidenciales y sin alteraciones [20].

Además de la confianza y la seguridad, los sistemas federados enfrentan el desafío de las topologías de red dinámicas y de la demanda y la oferta de recursos inciertas, ya que los nodos pueden dar y revocar su disponibilidad de intercambio de forma individual. La pertenencia a una federación puede ser incierta y, por tanto, volátil. Esta volatilidad plantea desafíos para coordinar, equilibrar y optimizar las operaciones. Una posible solución a este problema son los procesos de decisión de Markov (MDP). Los MDP son una herramienta para modelar las operaciones del sistema en condiciones de incertidumbre y pueden usarse para predecir y optimizar el comportamiento de la federación bajo incertidumbre [21]. Los operadores pueden analizar los posibles estados del sistema utilizando MDP, evaluar las probabilidades de transición entre estos estados y determinar las estrategias óptimas para la asignación de recursos y la cooperación. Por ejemplo, en los sistemas satelitales federados, los MDP pueden usarse para modelar escenarios en los que los satélites asignan dinámicamente el ancho de banda en función de las demandas operativas cambiantes y la membresía estatal.

3.4. FdS caso de estudio: sistemas de satélites federados.

A diferencia de otros sistemas satelitales distribuidos (DSS), que pueden involucrar constelaciones, clústeres o enjambres diseñados para operaciones integradas bajo control centralizado, los sistemas satelitales federados (FSS) implican la cooperación voluntaria entre naves espaciales de propiedad independiente que colaboran de manera oportunista [22]. Esta distinción diferencia a los FSS de otras arquitecturas DSS al introducir el oportunismo en el intercambio de recursos. Los satélites actúan tanto como clientes como proveedores de recursos, dependiendo de su estado operativo y capacidades, lo que permite misiones más dinámicas y rentables.

La Estación Espacial Internacional (ISS) ha sido considerada como un proveedor de recursos satelitales, como la potencia de cálculo [22], un concepto que más tarde se exploró para una infraestructura de computación en la nube a bordo de la ISS [23]. En el escenario de la ISS, la Estación contribuye con recursos como el ancho de banda de enlace descendente, el almacenamiento de datos y la potencia de cálculo a una red de naves espaciales clientes. El estudio muestra que incorporar la ISS en un sistema federado puede ayudar a compensar los altos costes operativos y de ciclo de vida de los programas espaciales tripulados, que a menudo superan los cientos de miles de millones. La sinergia de evaluación del valor de la federación combina análisis técnicos y económicos para identificar clientes potenciales, clasificarlos por asequibilidad y evaluar la viabilidad financiera de las operaciones de los sistemas satelitales federados (FSS). Los resultados muestran que los FSS no solo mejoran la sostenibilidad de las misiones espaciales, sino que también crean mercados comerciales escalables para el intercambio de recursos en órbita.

A diferencia de otras arquitecturas satelitales distribuidas, los satélites federados también ofrecen ventajas para abordar las ineficiencias relacionadas con los recursos satelitales infrautilizados. Por ejemplo, cada satélite experimenta períodos de inactividad durante sus operaciones en los que el ancho de banda disponible o la potencia de procesamiento no se utilizan. Los sistemas satelitales federados (FSS) permiten compartir estos recursos que de otro modo se desperdiciarían. Esta capacidad tiene un impacto particular en entornos orbitales que presentan variabilidad natural en el suministro de recursos, como la cobertura intermitente de estaciones terrestres en la órbita terrestre baja (LEO), que a menudo resulta en latencia y cuellos de botella de recursos. Aprovechando la estructura cooperativa de los FSS, los satélites pueden enlazar datos a través de pares con acceso más inmediato a estaciones terrestres, reduciendo así la latencia y mejorando la capacidad de respuesta general de la misión [22].

La distinción entre los sistemas satelitales federados y otros sistemas distribuidos radica en su enfoque en los beneficios mutuos y el control descentralizado. Mientras que las constelaciones optimizan la cobertura global y los tiempos de revisita a través de la coordinación centralizada, las federaciones priorizan la flexibilidad y la eficiencia económica al ajustar dinámicamente la oferta y la demanda de recursos en órbita. Los enjambres, que a menudo dependen de la autoorganización y la asignación autónoma de tareas, se diferencian de los sistemas satelitales federados (FSS) en que carecen de marcos económicos explícitos para el intercambio de recursos [22].

4. INTRODUCCIÓN CONCEPTUAL A LA TEORÍA DE GRAFOS HETEROFUNCIONALES¹

4.1. Introducción

La teoría de grafos heterofuncionales (HFGT) aprovecha la ingeniería de sistemas basada en modelos y la ciencia de redes para capturar fenómenos heterogéneos en sistemas heterogéneos dentro de un paradigma de modelado común [24]. HFGT se puede utilizar para modelar un número arbitrario de sistemas en red de topología arbitraria conectados de manera arbitraria [24]. Más específicamente, HFGT utiliza múltiples estructuras de datos basadas en grafos para apoyar un análisis cuantitativo basado en matrices, heredando la heterogeneidad de los constructos conceptuales y ontológicos encontrados en la ingeniería de sistemas basada en modelos, incluyendo la forma del sistema, la función del sistema y el concepto del sistema [24]. Además, se puede utilizar para conciliar y posteriormente generalizar tanto las redes multicapa [25, 54], como los modelos de diseño axiomático [52, 56], los modelos de dinámica de sistemas [40], los modelos de grafos de enlace y los modelos de grafos lineales [39]. Esta capacidad para conciliar modelos de sistemas de fuentes disciplinarias dispares, así como la capacidad para modelar SdS de topología arbitraria, permite a HFGT realizar análisis estructurales novedosos [57, 58], simulaciones dinámicas [59] y problemas de decisión óptimos [60].

El proceso de modelado abstrae un sistema del “mundo real” y representa la abstracción con un modelo. El modelo se refiere al sistema del mundo real, pero esta referencia siempre es indirecta, ya que siempre se hace una abstracción en el proceso de modelado. Aunque la abstracción del sistema del mundo real puede ser conceptual o lingüística, es importante distinguir entre una abstracción conceptual, que reside en la mente, y una abstracción lingüística, que reside dentro de un lenguaje predefinido generalmente asociado ontológicamente con un dominio de conocimiento del mundo real. Los modelos de grafos heterofuncionales son el modelo abstraído del

¹ Esta sección resume varios trabajos publicados anteriormente en una introducción cualitativa a la teoría de grafos heterofuncionales [24, 25].

sistema del mundo real, mientras que la teoría de grafos heterofuncionales proporciona los medios para representar conceptualizaciones ontológicamente heterogéneas de dominios del mundo real en un lenguaje matemático y computacional común.

La HFGT se ha aplicado a SdS en numerosos dominios de aplicación individualmente, incluyendo sistemas de transporte electrificados multimodales [50, 65, 66], sistemas de producción habilitados por microrredes, sistemas de entrega de atención médica personalizada [48, 59, 67], sistemas de hidrógeno-gas natural [60], el nexo energía-agua [68] y el sistema energético multimodal estadounidense [58].

4.2. Arquitecturas instanciadas, de referencia y meta-arquitecturas para Sistemas de Sistemas

Para comprender mejor el desafío de convergencia asociado con sistemas complejos de sistemas, incluidos los desafíos sociales del Antropoceno, considera lo que sucede cuando el sistema del mundo real es un SdS ontológicamente heterogéneos: el dominio real es un dominio-real-de-dominios-reales que une los dominios individuales en uno solo. Por ejemplo, el estudio del nexo agua-energía-alimentos en general (en lugar de para una región específica) puede constituir tal dominio-real-de-dominios-reales. Necesita ser abstraído por los humanos, en la mente, en una conceptualización-de-dominio-de-conceptualizaciones-de-dominio. De manera similar, eso debe ser descrito utilizando un lenguaje-de-lenguajes. Inmediatamente surgen varios desafíos de convergencia. Primero, las personas suelen estar entrenadas en una sola conceptualización de dominio, en lugar de múltiples dominios de conocimiento. De hecho, no está claro que siquiera exista un sólo humano (y mucho menos muchos) que tenga suficiente conocimiento de la conceptualización-de-dominio-de-conceptualizaciones-de-dominio. En ausencia de tal individuo, un grupo de individuos, cada uno con sus propias conceptualizaciones de dominio individuales, debe colaborar de alguna manera para discutir el dominio-real-de-dominios-reales (por ejemplo, el nexo agua-energía-alimentos independiente de una región específica). Inmediatamente se encuentra que cada conceptualización de dominio viene con su lenguaje asociado y emerge un lenguaje-de-lenguajes. Debido a que cada uno de estos lenguajes se desarrolló de manera completamente independiente para abordar las necesidades de su dominio real asociado, el lenguaje-de-lenguajes es altamente divergente y es difícil lograr una comprensión común y convergente entre

lenguajes. Para superar este impasse, es posible que el lenguaje-de-lenguajes desarrolle una capacidad de traducción entre cada uno de los lenguajes pertenecientes a cada dominio real. Si bien esta estrategia es relativamente sencilla para sólo dos lenguajes con un único traductor, no escala cuando hay N dominios reales que requieren N(N-1) traductores entre N lenguajes. La única alternativa es invertir en el desarrollo de un lenguaje-de-lenguajes que concilie los lenguajes individuales en un solo lenguaje común.

HFGT adopta el enfoque posterior, donde un único lenguaje común sirve como un lenguaje-de-lenguajes. El desarrollo de un único lenguaje común para una conceptualización-de-dominio-de-conceptualizaciones-de-dominio requiere tres tipos de arquitecturas de sistema. Como se muestra en la Figura 1, éstas son las arquitecturas instanciadas, de referencia y meta-arquitecturas.

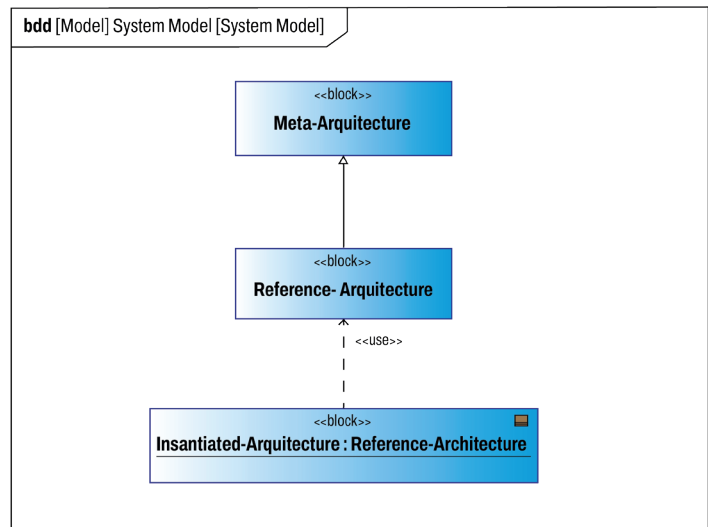


Figura 1. Diagrama de Definición de Bloques SysML. La arquitectura de sistemas puede representarse en tres niveles de abstracción con arquitecturas instanciadas, de referencia y meta-arquitecturas.

Aquí se concibe una arquitectura de sistemas compuesta por tres partes: la arquitectura del mundo real o física, la arquitectura funcional y la asignación de esta última a la primera en un concepto de sistema (o arquitectura asignada). La arquitectura física es una descripción de los elementos que componen el sistema sin ninguna especificación de las características de rendimiento de los recursos físicos que componen cada elemento. La arquitectura funcional es una descripción de los procesos del sistema de una forma neutral en cuanto a la solución, estructurada en serie, en paralelo y potencialmente en acuerdos jerárquicos. El

concepto de sistema como una asignación de la arquitectura funcional a la arquitectura física completa la arquitectura del sistema. Para que un modelo de gráfico hetero-funcional esté correctamente especificado, se asume que la totalidad de cualquier proceso dado debe ser completada por un recurso dado.

Una arquitectura de sistemas instanciada es una arquitectura específica de un caso, que representa un escenario del mundo real. A este nivel, la arquitectura física consiste en un conjunto de recursos instanciados y la arquitectura funcional consiste en un conjunto de procesos del sistema instanciados. La asignación en el concepto de sistema define qué recursos realizan qué procesos.

La arquitectura de referencia generaliza las arquitecturas de sistemas instanciadas. En lugar de usar instancias individuales como elementos de la arquitectura física y funcional, la arquitectura de referencia se expresa en términos de clases específicas del dominio de estas instancias. De esta manera, la arquitectura de referencia captura la esencia de las arquitecturas instanciadas existentes. También proporciona una visión de las necesidades futuras que puede guiar el desarrollo de nuevas arquitecturas de sistemas instanciadas. Esa arquitectura de referencia facilita una comprensión compartida a través de múltiples disciplinas u organizaciones sobre la arquitectura actual y su evolución futura. Una arquitectura de referencia se basa en conceptos probados en la práctica. La mayoría de las veces, las arquitecturas precedentes se exploran en busca de estos conceptos probados. Por tanto, la arquitectura de referencia generaliza las arquitecturas de sistemas instanciadas para definir una arquitectura que es generalmente aplicable en una disciplina. Sin embargo, la arquitectura de referencia no se generaliza más allá de la conceptualización del dominio.

La meta-arquitectura generaliza aún más las arquitecturas de referencia. En lugar de elementos específicos del dominio, se expresa en términos de clases neutrales al dominio. Una arquitectura de referencia está compuesta por “elementos primitivos” que generalizan los elementos funcionales y físicos específicos del dominio en sus equivalentes neutrales al dominio. Aunque no se ha desarrollado una meta-arquitectura de sistemas de ingeniería única para todos los propósitos, se han desarrollado varias metodologías de modelado que abarcan varios dominios específicos de disciplinas. En el diseño de sistemas dinámicos, los gráficos de enlace [27-29] y los gráficos lineales [30-34] utilizan condensadores, resistencias, inductores, giradores y transformadores generalizados como elementos primitivos. En la dinámica de sistemas de negocios, las existencias y los flujos a menudo se utilizan como elementos primitivos [35, 36]. Finalmente, la teoría de grafos formal [37, 38] introduce nodos y aristas como elementos primitivos. Cada uno de estos dominios tiene sus respectivos conjuntos de aplicaciones. Sin embargo, su suficiencia debe ser probada

en última instancia mediante un análisis ontológico de solidez, completitud, lucidez y concisión [24]. La teoría de grafos hetero-funcional, como se elabora en la siguiente sección, utiliza su propia meta-arquitectura que, en los últimos años, ha demostrado generalizar gráficos lineales, gráficos de enlace, dinámica de sistemas y teoría de grafos formal [25, 39, 40]. Dada la importancia de la claridad ontológica, la HFGT ha tenido especial cuidado en la traducción de esta meta-arquitectura desde su descripción en el lenguaje de modelado de sistemas (SysML) [41-43] hasta su representación matemática.

4.3. Elementos esenciales de la teoría hetero-funcional

Esta sección presenta los elementos esenciales de la teoría de grafos hetero-funcionales en términos de su meta-arquitectura subyacente. A diferencia de otras meta-arquitecturas, la teoría de grafos hetero-funcionales se basa en la estructura universal del lenguaje humano con sujetos y predicados, y estos últimos compuestos por verbos y objetos [24, 25]. Un sistema de ingeniería del mundo real incluye un conjunto de recursos como sujetos, un conjunto de procesos del sistema como predicados y un conjunto de operandos como sus objetos constituyentes [44-46], donde:

- Un **operando del sistema** es un activo u objeto que se opera o consume durante la ejecución de un proceso.
- Un **proceso del sistema** es una actividad que transforma o transporta un conjunto predefinido de operandos de entrada en un conjunto predefinido de operandos de salida.
- Un **recurso del sistema** es un activo u objeto que facilita la ejecución de un proceso. Se definen tres tipos: recursos de transformación, búferes independientes y recursos de transporte.

Como se muestra en la Fig. 2, estos operandos, procesos y recursos están organizados en una meta-arquitectura [41-43]. Es importante destacar que los operandos en el sistema de ingeniería tienen varios tipos, incluyendo materia, energía, organismos vivos, información y dinero, lo que hace que la HFT sea intrínsecamente ciber-física [24]. Curiosamente, se entiende que los operandos, en general, tienen algún tipo de estado en el tiempo. La evolución de este estado de operando se describe mediante una red de operandos como un tipo de red de Petri [46]. Las relaciones entre los diferentes tipos de procesos del sistema y los recursos que pueden ejecutarlos se representan más detalladamente en la meta-arquitectura funcional de la teoría de grafos hetero-funcionales (ver Fig. 3).

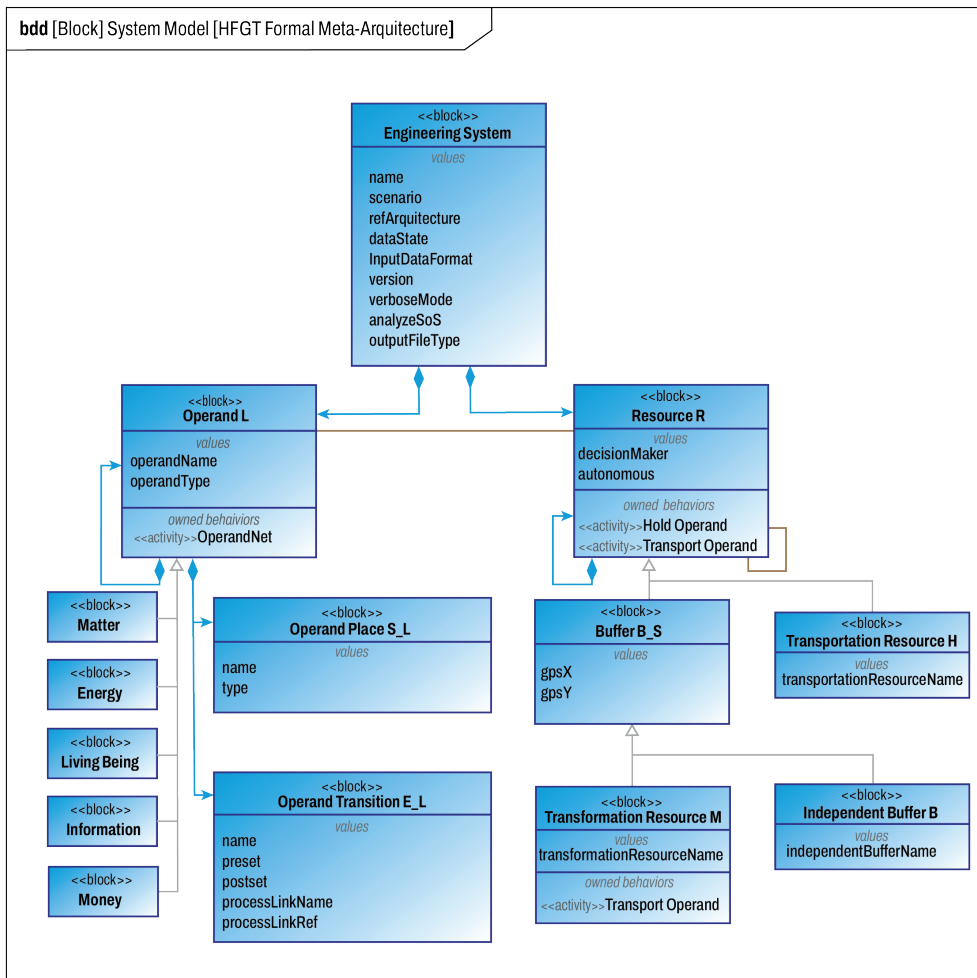


Figura 2. Diagrama de bloques SysML de la meta-arquitectura formal de HFGT.

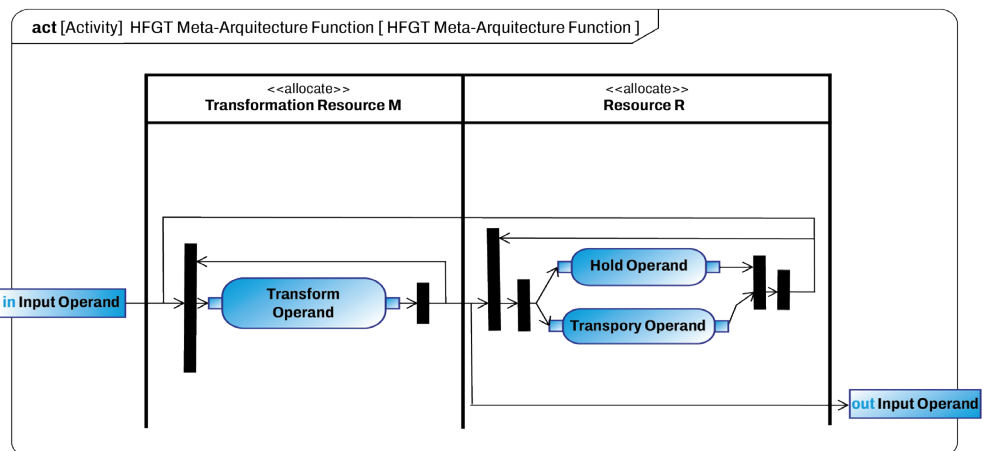


Figura 3. Diagrama de actividad SysML de la meta-arquitectura funcional HFGT funcional.

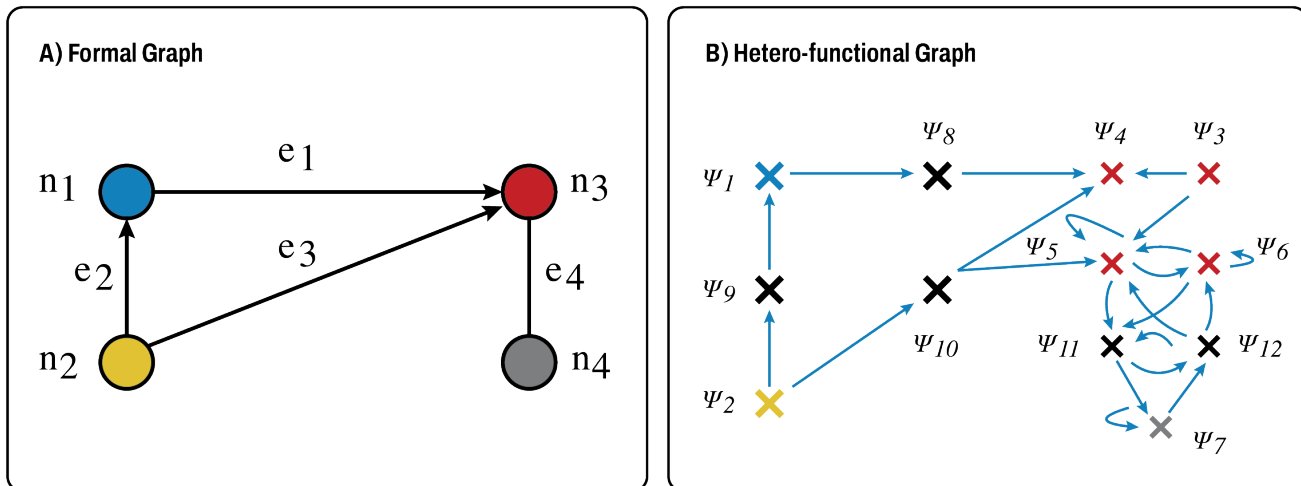
Los recursos son capaces de ejecutar uno o más procesos del sistema para producir un conjunto de capacidades [24]. Intuitivamente, una capacidad se articula como una oración de *sujeto + verbo + operando*. Es importante reconocer que, aunque las capacidades son entidades distintas por sí mismas, en realidad se forman mediante la asignación de un proceso a un recurso. En la Fig. 2, estas capacidades aparecen como comportamientos propios de sus respectivos bloques. En la Fig. 3, estas capacidades aparecen como acciones en sus respectivos carriles de natación (swim lanes en inglés). A nivel de un sistema de ingeniería, estas asignaciones se describen en el concepto del sistema, que puede capturarse como una matriz binaria cuyos elementos son iguales a uno cuando una acción está disponible como un proceso del sistema siendo ejecutado por un recurso. En otras palabras, el concepto del sistema forma un grafo bipartito entre el conjunto de procesos del sistema y el conjunto de recursos del sistema [51].

Una vez que se han definido las capacidades del sistema de ingeniería, es necesario entender cómo interactúan entre sí. Estas interacciones se muestran más claramente como flechas dirigidas entre las acciones asignadas a los carriles de natación en la Fig. 3. Matemáticamente, HFGT describe estas interacciones funcionales con tensores de incidencia.

4.4. Ejemplo ilustrativo

Esta breve introducción conceptual a la teoría de grafos hetero-funcionales también puede explicarse gráficamente a través del ejemplo ilustrativo mostrado en la Fig. 4

La Fig. 4 ilustra la diferencia entre un grafo formal y un grafo hetero-funcional (HFG). El grafo formal en la Fig. 4a muestra un sistema compuesto por cuatro nodos: una instalación de tratamiento de agua, un panel solar fotovoltaico, una casa con energía solar en el tejado y un lugar de trabajo. Estos están conectados por cuatro aristas: una tubería de agua, dos líneas eléctricas y dos carreteras. En contraste, la Fig. 4b muestra el grafo hetero-funcional asociado. En lugar de cuatro nodos que representan instalaciones puntuales, el grafo hetero-funcional ahora tiene 12 nodos que representan las capacidades del sistema conectado. La instalación de tratamiento de agua, el panel solar fotovoltaico y el lugar de trabajo aparecen sin cambios entre los dos grafos porque cada uno tiene solo una capacidad. En contraste, la casa con energía solar en el tejado proporciona cuatro capacidades en el HFG. Las aristas en el grafo formal ahora aparecen como capacidades de transporte en el HFG. Finalmente, las aristas dirigidas en el HFG indican las secuencias lógicas de estas capacidades, de modo que, si se siguen, emerge una "historia" de capacidades. Por ejemplo, la instalación de tratamiento de agua trata el agua (Ψ_1) y luego la tubería de agua transporta el agua desde la instalación de tratamiento de agua hasta la casa (Ψ_8).



Legend: Nodes: $\{n_1$ -Water Treatment Facility, n_2 -Solar PV, n_3 -House with Rooftop Solar, n_4 -Work Location} Edges: $\{e_1$ -Water Pipeline, e_2 -Power Line 1, e_3 -Power Line, e_4 -Road} System Capabilities: $\{\Psi_1$ -water treatment facility treats water, Ψ_2 -solar PV generates electricity, Ψ_3 -house generates electricity, Ψ_4 -house consumes water, Ψ_5 -house charges EV, Ψ_6 -house parks EV, Ψ_7 -work location parks EV, Ψ_8 -water pipeline transports water from water treatment facility to house, Ψ_9 -power line 1 transports electricity from solar PV to water treatment facility, Ψ_{10} -power line 2 transports electricity from solar PV to house, Ψ_{11} -road discharges EV from house to work location, Ψ_{12} -road discharges EV from work location to house}

Figura 4. Comparación visual de un modelo de grafo formal y un modelo de grafo hetero-funcional del mismo sistema hipotético [24].

5. MODELADO Y SIMULACIÓN FEDERADOS

5.1. Introducción

La mayoría de los SdS son demasiado complejos para el análisis matemático. Generalmente, el comportamiento de las redes de sistemas resultantes depende de sus enlaces y su entorno, donde el reduccionismo científico es incapaz de definir completamente el comportamiento. La simulación se ha convertido en una de las herramientas de análisis más utilizadas para sistemas a gran escala porque puede tener en cuenta la aleatoriedad y abordar modelos agregados, así como muy detallados. Además, a medida que la velocidad de cálculo ha aumentado y la comunicación ha mejorado, ha ido existiendo aún más motivación para utilizar la simulación para problemas cada vez más grandes, como una cadena de suministro. Aplicar la simulación a SdS lleva de forma natural a la simulación distribuida (referida como federaciones), donde los modelos de simulación heredados (referidos como elementos federados) se integran en una red de múltiples computadores. La existencia de modelos de sistemas constituyentes heredados es una fuerza impulsora para el modelado y la simulación federados. Otra es permitir que cada subsistema (por ejemplo, un miembro de la cadena de suministro) oculte cualquier información propietaria en la implementación de la simulación individual pero aún proporcione suficiente información para analizar SdS (por ejemplo, cadena de suministro) como un todo.

5.2. Federación de la simulación y sincronización temporal

El diseño y desarrollo de una red distribuida de simulaciones (es decir, una federación) es complejo y requiere experiencia en varias disciplinas, incluyendo expertos en el dominio, diseño y especificaciones del sistema, en el modelado de simulación y en computación y redes distribuidas. Una vez que existen modelos de sistemas constituyentes (por ejemplo, modelo de eventos discretos, dinámica de sistemas, basado en agentes o basado en física), la sincronización temporal de múltiples modelos de sistemas constituyentes (es decir, elementos federados) y el intercambio de información entre ellos son problemas importantes.

Para ilustrar cómo se construye y opera una federación, se utiliza una cadena de suministro como ejemplo en esta sección. La Figura 5 muestra la federación de una cadena de suministro, integrando elementos federados geográficamente dispersos, modelados por sistemas de software de simulación comercial (es decir, Arena para el proveedor, ProModel para la planta de ensamblaje y AutoMod para el transportista).

Dado que existen elementos federados, primero se necesita un servidor (por ejemplo, RTI en la Figura 5) que se comunicará con los elementos federados. En segundo lugar, para habilitar dichas comunicaciones, cada elemento federado necesita un interfaz (por ejemplo, Adaptador en la Figura 5), que enviará/recibirá mensajes hacia/desde el

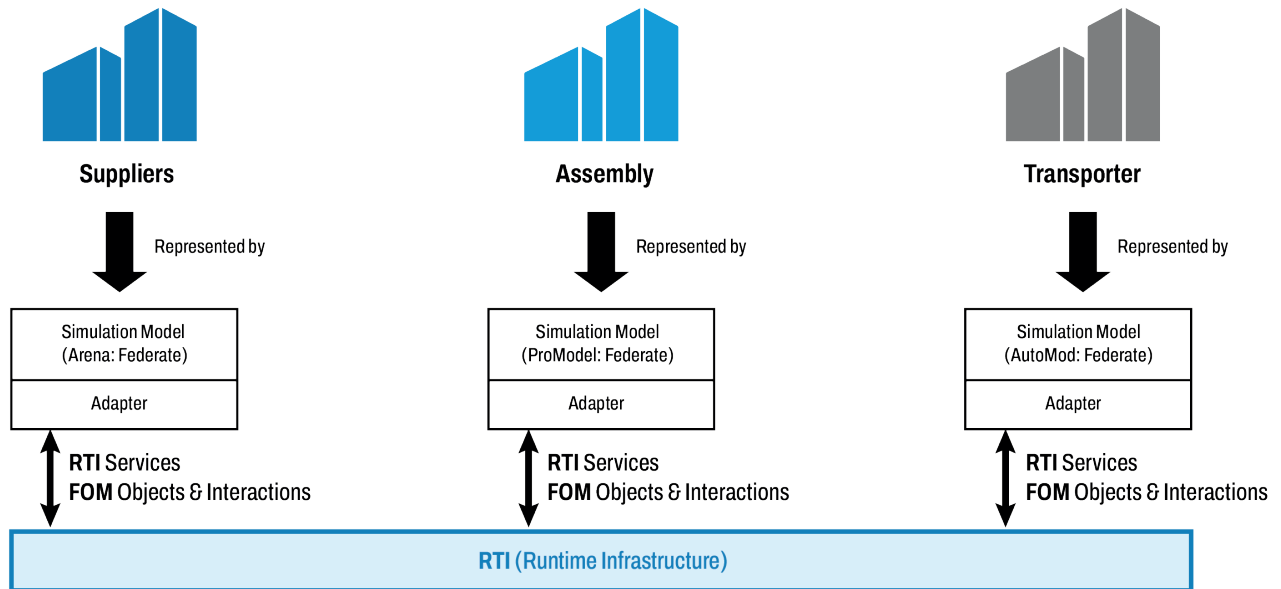


Figura 5. Federación de cadena de suministro utilizando el marco IEEE Std 1516-2010 [69].

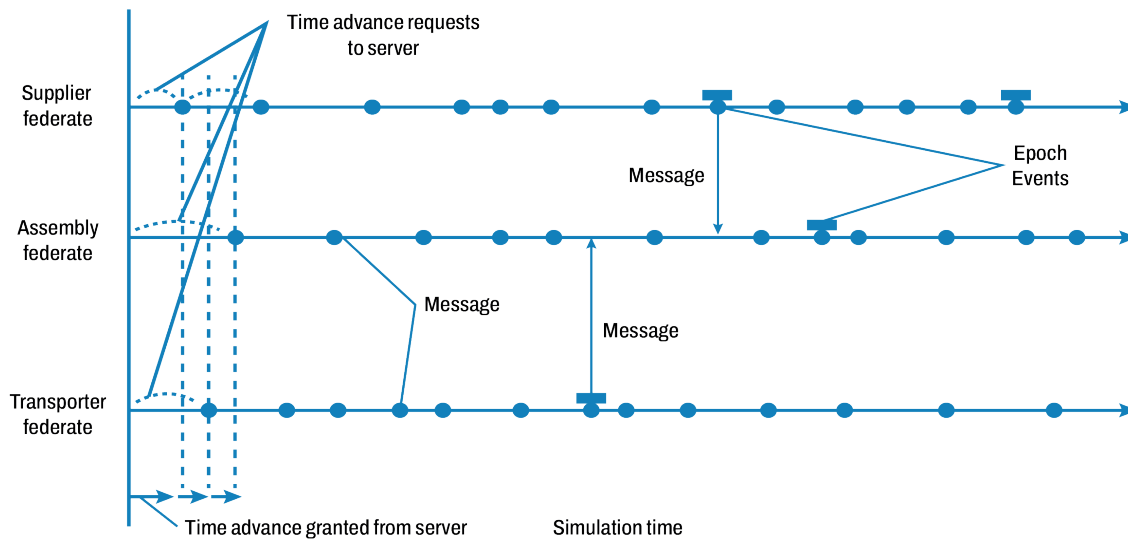


Figura 6. Enfoque de sincronización de tiempo conservador basado en eventos [70].

servidor (ver Figura 6). Las principales responsabilidades del servidor son la sincronización temporal y el intercambio de información entre los elementos federados.

Como se muestra en la Figura 6, cada elemento federado tiene dos tipos de eventos: (1) eventos internos (los modelos continuos necesitan ser discretizados para crear eventos internos) y (2) eventos de interacción. Los eventos internos son los eventos que no influyen en otros elementos de la federación [70]. Un evento de interacción se define como un evento que influye en el comportamiento de otro(s) elemento(s) federado(s). La información sobre los eventos de interacción junto con los parámetros correspondientes se comunica a los elementos federados receptores y los elementos federados receptores ajustan su comportamiento en consecuencia para permitir que el sistema funcione correctamente. Si un elemento federado recibe un evento de interacción cuya marca de tiempo es anterior al reloj de simulación actual del elemento federado, entonces se produce una violación de la "restricción de causalidad"; si no se resuelve, esta violación puede invalidar los resultados de la simulación. Por eso es importante la sincronización temporal.

Se han utilizado dos clases principales de enfoques para sincronizar elementos federados: conservador y optimista. En el enfoque de sincronización temporal conservador, los elementos federados determinan el tiempo para su evento inmediato siguiente y solicitan permiso al servidor para avanzar a ese tiempo. Después de obtener solicitudes de todos los elementos federados, el servidor determina el paso de tiempo más bajo solicitado y otorga permiso a los elementos federados para avanzar. Cada elemento federado

ejecuta el siguiente evento y solicita permiso para avanzar al evento inmediato siguiente, y el ciclo continúa. En el enfoque conservador, no se hace distinción entre los eventos internos y los eventos de interacción en términos de gestión del tiempo. En cada evento de interacción, se enviará un mensaje desde el elemento federado al servidor, que luego reenviará el mensaje al elemento federado correspondiente.

En el enfoque optimista, los elementos federados avanzan en paralelo en incrementos fijos o variables. Si ocurre una interacción entre elementos federados, se retrocede al momento en que ocurrió la interacción. Sin embargo, es extremadamente difícil implementar un mecanismo de retroceso en la simulación.

5.3. Marcos que permiten la federación

Existen varios marcos disponibles en la literatura para habilitar la simulación distribuida [71-76].

Como ejemplo, el IEEE Std 1516-2010 describe el marco y las reglas de la Arquitectura de Alto Nivel (HLA), que es un enfoque integrado para proporcionar una arquitectura común para simulaciones federadas. Siguiendo una arquitectura de publicación y suscripción, HLA puede ser aplicable a varios tipos de sistemas operativos, software, aplicaciones y lenguajes. Por ejemplo, permite la integración de una amplia gama de software: AnyLogic, Simio, Arena, ProModel, Repast, DynusT (simulador de tráfico), hardware (robots, máquinas, drones), Unity (motor de juegos) y más. Para mantener o gobernar los modelos, se puede utilizar la herramienta de código abierto Pórtico ([<http://porticoproject>].

org/](http://porticoproject.org/)) o un RTI comercial (por ejemplo, MAK Technologies), y se necesitan esfuerzos para desarrollar la gobernanza técnica. Además, es necesario establecer una estructura de gobernanza y un acuerdo entre patrocinadores y usuarios.

6. MODELADO BASADO EN AGENTES PARA SISTEMAS DE SISTEMAS

6.1. Paradigma de modelado basado en agentes

El modelado basado en agentes (ABM, por sus siglas en inglés) es una técnica computacional utilizada para estudiar y comprender el comportamiento de sistemas complejos. El enfoque adopta una perspectiva fundamentalmente diferente, caracterizada como de abajo hacia arriba, modelando las entidades individuales que conforman el sistema y sus interacciones. Estas entidades individuales, denominadas agentes, pueden representar individuos, grupos, organizaciones o cualquier tipo de entidad autónoma. Dichos modelos son adecuados para apoyar el análisis y la ingeniería de los SdS y comprender sus propiedades también. El ABM ofrece a los analistas flexibilidad para modelar propiedades clave de los SdS, incluyendo la autonomía de sus sistemas constituyentes, la pertenencia de estos a la misión del SdS, la conectividad entre los sistemas y las diversas propiedades de los mismos [77]. La metodología de modelado también sirve como un laboratorio digital para explorar propiedades emergentes y comportamientos no intuitivos de los SdS. Esta sección introduce los conceptos básicos del modelado basado en agentes y expone el valor del ABM para el análisis y la ingeniería de SdS a través de algunas aplicaciones ilustrativas en la literatura.

La evolución del ABM desde conceptos teóricos hasta aplicaciones prácticas abarca varias décadas en diversas disciplinas. El concepto fundamental de agentes autónomos y sus interacciones fue propuesto por primera vez por John von Neuman [83], que sentó las bases para el trabajo de John Conway sobre autómatas celulares, el Juego de la Vida, donde reglas simples aplicadas a células en una cuadrícula llevan a la emergencia de comportamientos complejos [84]. Este trabajo influyó en el desarrollo de los primeros ABM de simulaciones sociales, como el modelo de segregación de Thomas Schelling, el modelo Sugarscape de Joshua Epstein y Robert Axtell [78], que fue un ABM mejorado del Juego de la Vida de Conway y el modelo de segregación de Schelling. Otros modelos influyentes, como el trabajo de Robert Axelrod sobre la evolución de la cooperación [86], demostraron las potenciales fortalezas de explorar la dinámica de las interacciones de los agentes, la emergencia y la complejidad en sistemas sociales y económicos. La metodología se expandió a una

amplia gama de disciplinas a lo largo de los años, como el análisis de la propagación de enfermedades, la dinámica de los ecosistemas y la dinámica simulada de los mercados. Los avances computacionales han acelerado el uso del ABM en los últimos años. Ahora se utiliza en una amplia gama de dominios de aplicación, incluyendo diversas aplicaciones de SdS, como el modelado de infraestructuras, el análisis de sistemas de defensa y la planificación urbana. Esta sección primero proporciona una breve instantánea del paradigma de modelado y cómo se construye. A continuación, se discute el valor del ABM para el análisis y la ingeniería de SdS con algunos ejemplos de problemas de ingeniería de SdS.

6.2. Agentes de modelado

Aunque existen diferentes opiniones sobre la definición de lo que constituye un agente, en la práctica los agentes tienen las siguientes características en el contexto del ABM [80]:

- **Autonomía y autodirección:** Las acciones de un agente son autodirigidas e independientes, tanto en su entorno como en sus interacciones con otros agentes.
- **Características identificables:** Cada agente en el modelo tiene un conjunto identificable de características, comportamiento y capacidades de gestión.
- **Interacción con otros agentes:** Los agentes tienen un conjunto de reglas que determinan cómo interactúan con otros agentes.

Otras propiedades adicionales que pueden considerarse al modelar agentes incluyen [4]:

- **Situados en un entorno:** Los agentes pueden actuar e interactuar dentro de un entorno donde su comportamiento depende de las interacciones con otros agentes y con el entorno.
- **Comportamiento orientado a objetivos:** Los agentes pueden tener objetivos que son criterios de decisión que utilizan para evaluar la efectividad de su comportamiento.
- **Capacidad de aprender y adaptarse:** Los agentes pueden tener la capacidad de aprender y adaptar su comportamiento basado en experiencias previas.
- **Recursos:** Los agentes pueden tener recursos como energía, información, riqueza que cambian dinámicamente según las interacciones.

El modelado de agentes en ABM implica, como mínimo, identificar las características de los agentes e identificar los comportamientos de los agentes. Las características de los agentes definen atributos específicos que distinguen a un agente de otros tipos de agentes. Por ejemplo, en el tráfico,

el tamaño/peso del vehículo es una característica que determina si el vehículo es una motocicleta, un coche o un camión. Otros atributos de un vehículo podrían ser la velocidad o el consumo de combustible. La selección de características/atributos de los agentes depende del dominio de interés y del nivel de detalle necesario para capturar el problema del mundo real. Cada agente también debe tener un comportamiento que es un conjunto de reglas que el agente sigue al interactuar con el entorno y con otros agentes. El comportamiento de un agente puede ser en base a reglas simples de tipo “si-entonces” o puede describirse mediante modelos de comportamiento complejos como modelos de decisión cognitiva o modelos de inteligencia artificial. Los modelos de comportamiento también pueden incorporar adaptabilidad, donde el agente cambia dinámicamente su comportamiento en respuesta a sus experiencias. Mientras que algunos de los modelos de comportamiento de los agentes se basan en datos empíricos o conocimientos del dominio, los modelos de comportamiento avanzados de los agentes a menudo se basan en conceptos teóricos de diversas disciplinas como la teoría de juegos, la ciencia cognitiva, el aprendizaje por refuerzo y la inteligencia artificial. La selección del tipo de modelo de comportamiento de los agentes depende del dominio y del propósito del ABM. Por ejemplo, si el propósito del modelo es evaluar el impacto de una regla de señalización específica en la congestión del tráfico, puede que no sea necesario modelar la adaptación de los agentes.

El modelo de agentes también determina el nivel de abstracción del problema del mundo real. Mientras que otras metodologías de modelado son adecuadas para un nivel específico de abstracción, el ABM proporciona la flexibilidad para abstraer el sistema del mundo real en diferentes niveles de detalle. La Figura 7 compara el ABM con otras metodologías de modelado. El modelo de agentes puede representar una entidad a nivel macro, como una organización, frente a una abstracción de nivel inferior, como un equipo de diseño de ingeniería, o a nivel macro se puede analizar la dinámica del transporte multimodal abstrayendo los transportistas ferroviarios, el transporte de carga aérea, el comportamiento de los transportistas de camiones frente a la dinámica de nivel inferior mediante el modelado de la dinámica de los sistemas individuales.

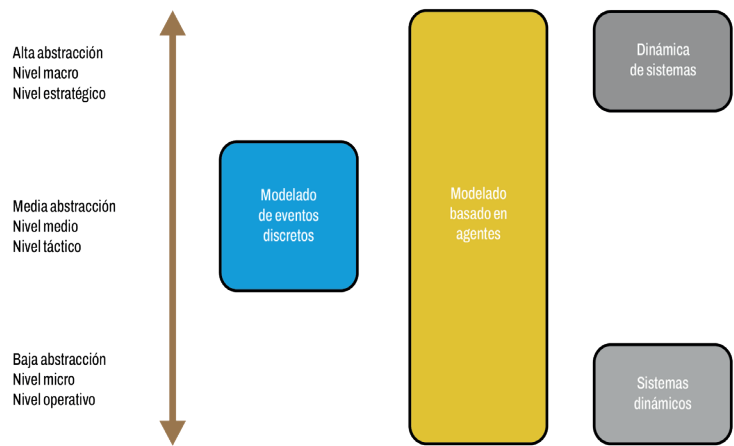


Figura 7. Niveles de abstracción del ABM comparados con otros métodos de modelado [80].

6.3. Modelado del entorno

El entorno donde los agentes interactúan puede modelarse de varias maneras. La Figura 8 ilustra los tipos de entornos que pueden modelarse en ABM. Esto podría ser un modelo de un espacio físico en 2-D, una estructura de cuadrícula, una red o una representación espacial compleja como un mapa geográfico. También es posible que los agentes interactúen en un modelo de entorno no espacial. Independientemente del tipo de modelo, el entorno proporciona el contexto para la interacción de los agentes.

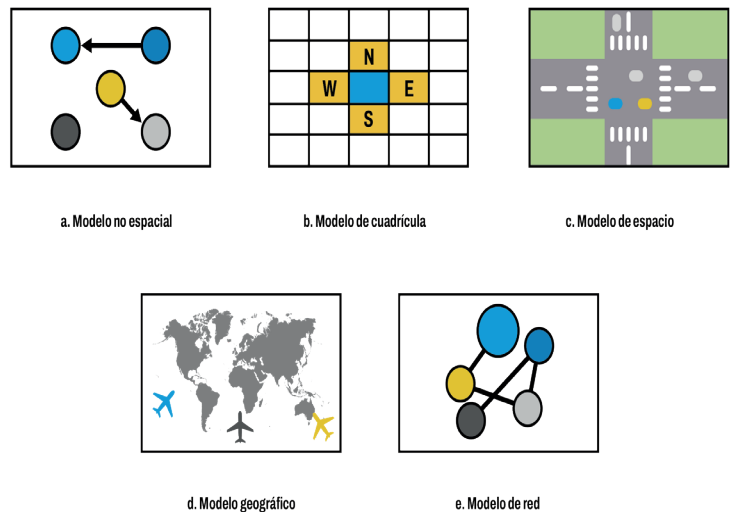


Figura 8. Modelado del entorno [80].

6.4. Modelado de interacciones de agente

Las interacciones entre agentes pueden incluir la comunicación, la competencia, la cooperación y la compartición de recursos. Independientemente del modelo de entorno utilizado para conectar los agentes, el punto principal del modelado de las interacciones de los agentes es identificar las reglas de las interacciones locales entre los agentes y la transferencia de recursos locales entre los agentes. Esto significa que los agentes interactúan con un número limitado de otros agentes en la población total. La naturaleza de estas interacciones a menudo conduce a comportamientos emergentes.

Para dar un ejemplo de cómo se construye el ABM, se consideran unos Vehículos Aéreos No Tripulados (UAV) que se utilizan en varias misiones de SdS, como la monitorización de grandes áreas para proporcionar datos situacionales para vigilancia y reconocimiento, la localización de personas en peligro para misiones de búsqueda y rescate, o la entrega de bienes para servicios de entrega [79]. En una misión de búsqueda y rescate, los UAV forman un SdS donde cada UAV opera de manera autónoma utilizando sus sensores y se comunica con otros UAV para evitar áreas de búsqueda superpuestas. Si un UAV detecta a una persona en peligro, puede comunicarse con los UAV cercanos para enviar la información a un centro de mando central. Se puede modelar un ABM de UAV para analizar la cobertura efectiva del área de búsqueda, el uso eficiente de los recursos y evaluar el tiempo de respuesta del SdS. El modelo simula el comportamiento y las interacciones de los UAV autónomos. A continuación, se proporciona una descripción de cómo se puede aplicar el ABM para esta misión de SdS:

- **Definir tipos de agentes y sus características:** En este contexto, los UAV son los agentes autónomos. Se pueden considerar varios atributos de los UAV, como posición, velocidad, tipos de sensores y nivel de batería. Los objetivos de la misión pueden distinguir cada tipo de UAV también. Por ejemplo, algunos de los UAV pueden asignarse para vigilancia y otros para misiones de reconocimiento.
- **Modelar el comportamiento de los agentes:** Se definen los comportamientos de los UAV. Los UAV perciben su entorno utilizando sus sensores y, según su estado actual y las reglas, deciden qué acciones tomar, como cambiar de dirección o velocidad. Finalmente, ejecutan su decisión y actualizan su posición y su estado. Para apoyar la percepción y la toma de decisiones, por ejemplo, se pueden incorporar algoritmos para evitar colisiones y de planificación de rutas en los modelos de comportamiento de los UAV. También se pueden incorporar capacidades de adaptabilidad en el modelo de comportamiento. Por ejemplo, el UAV puede adaptar su movimiento replanteando su ruta según los cambios en el entorno y los objetivos de la misión.

- **Modelar el entorno:** El modelo del entorno captura el espacio aéreo en el que operan los UAV. Se puede modelar un mapa geográfico del área con información sobre el terreno, zonas restringidas o áreas objetivo para los UAV para representar el entorno operativo.
- **Modelar las interacciones de los agentes:** Los protocolos de comunicación para compartir información entre los UAV modelan la interacción entre los UAV. La información compartida incluye obstáculos y el estado de los UAV. Esta información ayuda a los UAV a coordinar movimientos y evitar colisiones. El modelo de interacción también puede incorporar la asignación de tareas a los UAV según su estado actual.
- **Definir el proceso de simulación:** Inicializar el estado de los UAV, que incluye posiciones iniciales, velocidades y objetivos de la misión, así como configurar los parámetros del modelo del entorno. Luego, la simulación se ejecuta en pasos de tiempo discretos. En cada paso de tiempo, cada agente UAV percibe, decide y ejecuta un movimiento para actualizar su posición y estado.
- **Analizar el comportamiento del sistema:** Mediante la simulación, se puede obtener conocimiento sobre la dinámica de búsqueda y rescate y sobre los posibles comportamientos emergentes que pueden surgir de la interacción de los UAV durante el tiempo de simulación. Estas observaciones proporcionan información sobre la dinámica del sistema.

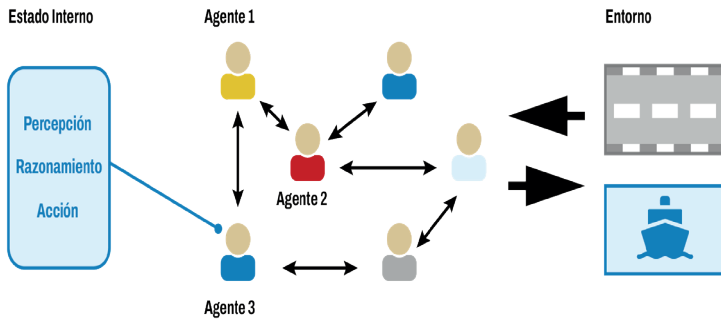
Para resumir, ABM implica varios pasos clave para crear una simulación dinámica que modele la complejidad de los sistemas del mundo real:

- Definir tipos de agentes y sus características.
- Modelar el comportamiento de los agentes definiendo reglas que gobiernen el comportamiento y las interacciones de los agentes. Esto también podría incluir el comportamiento de adaptación, donde los agentes adaptan su comportamiento basado en la experiencia y la realimentación del entorno. Los algoritmos de aprendizaje o reglas simples para simular comportamientos evolutivos pueden incorporarse en los modelos de comportamiento de los agentes.
- Modelar el entorno seleccionando un modelo topológico.
- Modelar las interacciones de los agentes definiendo reglas de interacción de los agentes.
- Definir el tiempo de simulación identificando cómo progresa el tiempo en la simulación. Esto podría ser pasos de tiempo discretos o tiempo continuo.

- Analizar el comportamiento del sistema. Observar y analizar el comportamiento emergente que surge de la interacción de los agentes durante el tiempo de simulación. Estas observaciones proporcionan información sobre la dinámica del sistema.

6.5. Tipos de problemas de SdS adecuados para ABM

El ABM es adecuado para analizar varios tipos de problemas de SdS que implican interacciones o comportamientos complejos del sistema. La Tabla 4 proporciona una muestra de dominios y problemas de SdS explorados utilizando ABM. Se puede encontrar una revisión sistemática detallada de los problemas de SdS en [82], donde el ABM de problemas de SdS se categoriza como dominios complejos relacionados con SdS, aspectos sociales relacionados con SdS, problemas de rendimiento relacionados con SdS, enfoques de optimización relacionados con SdS, simulaciones de SdS para problemas de política, problemas relacionados con la ingeniería de SdS y aspectos teóricos de SdS.



Modelado basado en agentes: comportamiento del agente, interacción del agente, entorno

Figura 9. Elementos del Modelado Basado en Agentes.

Tipo de problema SdS	Descripción
Dominios Complejos Relacionados con SdS: Red de transporte aéreo	ABM se utiliza como una herramienta de apoyo a la toma de decisiones. Se modela una red de transporte aéreo para analizar cómo la red se comporta a lo largo del tiempo cuando los patrones de comportamiento de varios agentes, como aeropuertos y agencias gubernamentales, cambian.
Dominios Complejos Relacionados con SdS: Análisis de políticas de transporte urbano	El modelo SdS analiza el impacto de varias políticas de transporte urbano en una ciudad mediante el modelado del comportamiento de los usuarios en los sistemas de transporte urbano.
Aspectos Sociales Relacionados con SdS: Relación UAV-humano	El modelo SdS analiza el nivel de autonomía de los UAV utilizados para la vigilancia mediante el análisis del número de operadores, el nivel de autonomía de los UAV y el rendimiento del SdS.
Aspectos Sociales Relacionados con SdS: Respuesta a la demanda de la red inteligente	Se diseña un modelo SdS donde las plantas de energía, las subestaciones y los agentes consumidores trabajan juntos para equilibrar la oferta y la demanda. El modelo proporciona información sobre la dinámica de la respuesta a la demanda de la red inteligente.
Enfoques de Optimización Relacionados con SdS: Incendios forestales	Se desarrolla un modelo colaborativo de SdS para predecir el comportamiento y la efectividad de varias configuraciones de detección de incendios.

Tipo de problema SdS	Descripción
Enfoques de Optimización Relacionados con SdS: Optimización de la cartera de guerra naval [5]	Se modelan y evalúan carteras de configuraciones de arquitectura SdS en función de capacidades, costes y riesgos operativos en un escenario de guerra naval. El análisis del modelo apoya el desarrollo de la arquitectura SdS.
Problemas de Rendimiento Relacionados con SdS: Resiliencia de la red	Se modela un escenario de guerra naval en red para evaluar la resiliencia de las arquitecturas potenciales ante amenazas e interrupciones.
Simulaciones de SdS para Problemas de Política: Generación de energía	El modelo SdS explora el impacto de las políticas para mejorar la generación de energía. El modelo proporciona información para que los diseñadores de sistemas vislumbren opciones alternativas de diseño de SdS.
Problemas Relacionados con la Ingeniería de SdS: Comunicación en la ingeniería de SdS	Se desarrolla un ABM ficticio de SdS para analizar el impacto del conocimiento interno y la comunicación en el rendimiento de SdS. Los resultados revelan que el conocimiento interno adicional y la comunicación entre los sistemas constituyentes mejoran el rendimiento de SdS.
Problemas Relacionados con la Ingeniería de SdS: Desarrollo de SdS basado en el modelo de proceso de adquisición de Ondas [70]	Se modela un ABM para el desarrollo de SdS reconocido basado en el modelo de proceso de adquisición de Ondas, donde el agente SdS negocia con sistemas individuales para adquirir capacidades deseadas para la arquitectura SdS.

Tabla 4. ABM de problemas de SdS

6.6. Valor del ABM para la ingeniería de SdS

ABM es una herramienta valiosa para apoyar la ingeniería de SdS ya que proporciona un marco integral para analizar, diseñar y gestionar SdS. De alguna manera, ABM sirve como un laboratorio de pruebas para entender cómo los comportamientos individuales de los sistemas conducen a un comportamiento emergente en un SdS. Esto es importante para gestionar las complejas interacciones que no son intuitivas al estudiar el comportamiento de sus sistemas individuales. Las simulaciones pueden revelar comportamientos no intuitivos debido a las complejas interacciones entre sus sistemas constituyentes, lo cual es esencial para diseñar SdS que puedan adaptarse y responder a condiciones cambiantes. Los ingenieros también pueden usar ABM para simular varias configuraciones de la arquitectura de un SdS y evaluar el impacto de las alternativas arquitectónicas en su rendimiento general [81]. ABM también puede ser utilizado como una herramienta de apoyo a la toma de decisiones para analizar varios escenarios, incluidos escenarios raros o extremos. Este tipo de análisis de escenarios hipotéticos ayuda a evaluar la efectividad del SdS en su misión bajo diferentes condiciones, proporcionando

valiosos conocimientos para los ingenieros de sistemas. Dado que ABM proporciona la flexibilidad para modelar diferentes tipos de agentes y comportamientos, es útil en la ingeniería de SdS donde los sistemas individuales tienen intereses y motivaciones diferentes. Así, ABM proporciona la flexibilidad y adaptabilidad para modelar todos los tipos de SdS, incluidos los dirigidos, reconocidos, colaborativos y virtuales. Además, se pueden diseñar modelos escalables añadiendo nuevos agentes para modelar SdS a gran escala.

6.7. Desafíos

Como cualquier metodología de modelado, ABM tiene sus limitaciones. ABM puede ser exigente en términos de potencia de procesamiento y memoria al simular sistemas a gran escala con muchos agentes. Modelar el comportamiento de los agentes requiere datos completos y consistentes, que pueden no estar disponibles para algunos dominios de aplicación. Lo más importante es que la validación de ABM puede ser desafiante, ya que requiere validar el modelo contra datos empíricos que pueden no estar disponibles para

algunas aplicaciones. Además, los resultados del modelo son sensibles a las condiciones iniciales y parámetros del modelo, y la naturaleza estocástica de las interacciones de los agentes hace que sea difícil reproducir los resultados del modelo. A pesar de estos desafíos, ABM sigue siendo una herramienta valiosa para comprender sistemas complejos, incluidos los SdS.

7. UNA APROXIMACIÓN DE SISTEMAS DE SISTEMAS AL MODELADO

7.1. Introducción al modelado de SdS

El modelado de un SdS suele implicar la integración de múltiples modelos de sistemas con múltiples niveles de abstracción [89, 91]. A diferencia de muchos modelos más simples de sistemas individuales, los modelos de SdS pueden necesitar capturar procesos que operan a diferentes escalas (p. ej., temporal, espacial, organizacional), con impulsores exógenos de los sistemas individuales que se vuelven endógenos y con múltiples mecanismos de realimentación entre los sistemas individuales integrados en el sistema de sistemas [89, 91]. Además, los modelos de SdS generalmente necesitan integrar el conocimiento de varias disciplinas con un intercambio de información entre ellas que se produce de forma coherente y significativa. La integración del conocimiento no se limita al acoplamiento técnico de los modelos por parte de expertos en las disciplinas implicadas, sino a la integración entre las partes interesadas que pueden participar en diferentes sistemas a diferentes escalas. Como resultado, las cuestiones de escala [81, 82] son con frecuencia una consideración central del modelado de SdS.

7.2. Entender la escala

El rango de disciplinas involucradas en el modelado de SdS (por ejemplo, ver [92]) a menudo significa que se utilizan diferentes nociones de escala de diferentes maneras dependiendo del contexto [89]. La elección de la escala claramente necesita ser consistente con el propósito del modelado y con las escalas espaciales y temporales representadas en los sistemas individuales. Las características espaciales y temporales de un sistema suelen ser los aspectos principales alrededor de los cuales se considera y se enmarca la escala. Estos definen el tiempo y el espacio de interés, incluyendo la discretización y los eventos y procesos que se consideran importantes de representar [93]. Las escalas espaciales seleccionadas pueden estar influenciadas por las escalas temporales de interés y viceversa.

La resolución define la granularidad de la representación del sistema y se refiere a la unidad de escala espaciotemporal representada en cada sistema. La resolución puede ser de naturaleza espacial o temporal, pero se extiende de otras maneras, como a los sistemas sociales (por ejemplo, incluyendo individuos, grupos y comunidades) y, por tanto, puede representar una jerarquía semántica o conceptual [85]. La elección de la resolución depende en gran medida del contexto del modelado, generalmente informada por la disponibilidad de datos, las necesidades del modelo (incluyendo la estabilidad numérica, la sensibilidad y la identificabilidad del modelo) y el propósito del modelo [89]. La jerarquía y los niveles de organización se relacionan con la representación de relaciones anidadas entre sistemas [94]. Por ejemplo, los sistemas de gobernanza pueden coexistir en una variedad de escalas con unidades administrativas separadas. Las organizaciones basadas en equipos pueden tener escalas jerárquicas, con miembros que desempeñan una variedad de roles dentro de una organización que puede estar geográficamente distribuida en diferentes zonas horarias.

En el modelado de SdS, cada modelo individual puede operar en diferentes escalas espaciales/temporales, niveles jerárquicos y resoluciones para incorporar múltiples aspectos de dominios y paradigmas de modelado claramente separados (disciplinares o sectoriales) (por ejemplo, redes bayesianas, basadas en agentes y dinámica de sistemas) [95].

7.3. Consideración de la escala durante las principales fases del proceso de modelado

Como se muestra en la Figura 10, el proceso de modelado puede representarse con cinco fases principales [90]. Estas fases son iterativas, con actividades de múltiples fases que a menudo ocurren simultáneamente y con decisiones tomadas en fases anteriores que se revisitan. La práctica de modelado tiende a centrarse en la formulación y evaluación del modelo, pero las otras fases son igualmente importantes si se requiere transparencia, coherencia y equidad [90].

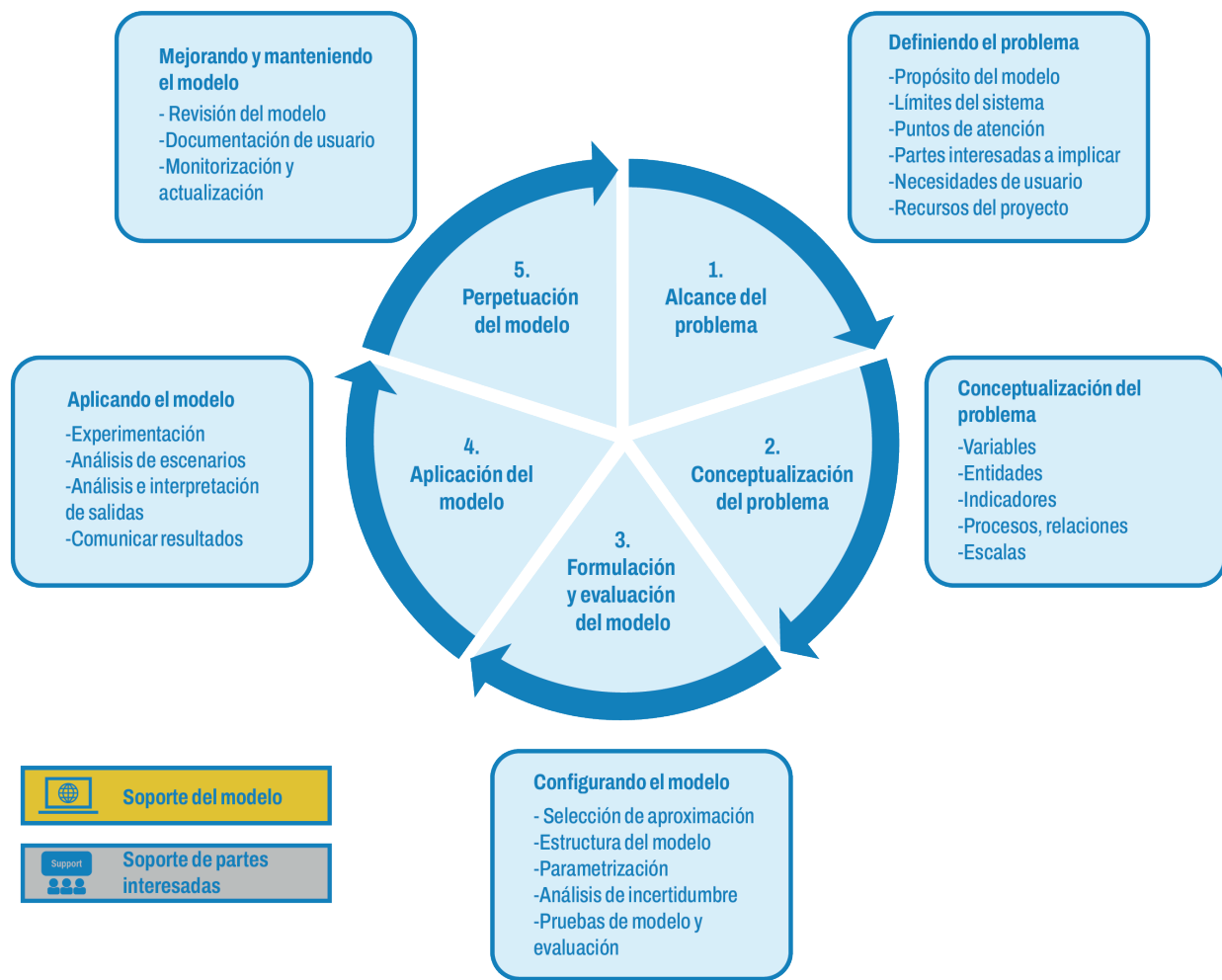


Figura 10. Las cinco fases principales del proceso de modelado, donde las flechas circulares iterativas ilustran la extensión aproximada del modelo y el apoyo de las partes interesadas en cada fase.

Cada fase requiere apoyo tanto de las herramientas de modelado como de los procesos de participación de las partes interesadas, como se indica en la Figura 10. Cabe destacar que cualquiera, o todas las fases, podrían estar motivadas políticamente para un proyecto de modelado específico, debido, por ejemplo, a un desequilibrio en la representación de las partes interesadas [96]. Esto podría llevar a sesgos en el modelo [9], variables dependientes del tiempo-lugar-financiación [98, 99] o un requisito para que los resultados del modelo se alineen con intereses específicos. Por tanto, una adecuada participación de las partes interesadas resulta crítica a lo largo de las cinco fases. La participación de las partes interesadas garantiza que el modelo sea adecuado para su propósito, represente múltiples perspectivas y se utilice y adopte según lo previsto [100].

Hay muchos puntos de decisión a lo largo del ciclo de vida del proceso de modelado, incluyendo: la selección de los límites del modelado (es decir, el propósito del modelo, la definición del problema y los límites del sistema), la base de evidencias (qué datos, y de quién es el conocimiento o la perspectiva), las características del modelo (como las variables, los resultados y las escalas), el enfoque de modelado y los métodos de prueba y evaluación del modelo [100]. Diferentes elecciones en cualquiera de estos puntos de decisión pueden dar lugar a diferentes caminos de modelado, llevando a diferentes modelos y resultados del modelado [101, 102]. Esto destaca la subjetividad inherente a las prácticas de modelado [103] y la necesidad de una reflexión continua durante el ciclo de modelado [104].

Las siguientes secciones son consistentes con las cinco fases de modelado [90]. En cada sección, el primer párrafo proporciona una breve descripción de las acciones típicas realizadas en cada fase de modelado extraídas de [90]. Los párrafos restantes proporcionan una visión general de algunas consideraciones relacionadas con los problemas de escala [89].

7.4. Delimitación del problema

La fase inicial de delimitación del problema implica definir el problema a abordar y su alcance, incluyendo la función o propósito del modelado, los límites del sistema, los temas o preguntas a abordar y las partes interesadas a involucrar. Esta fase de planificación también debe aclarar el contexto del usuario final, cubriendo tanto las necesidades del usuario como las de gestión, el contexto del problema, incluyendo la naturaleza del mismo y su correcto entendimiento, y el contexto del proyecto, que incluye los recursos disponibles como el tiempo, la financiación, las habilidades y los datos [100]. Esta es una fase crítica porque determina qué intereses se abordan y quiénes pueden participar en la formulación del problema.

Aunque el propósito general del modelo de SdS puede ser conocido, los detalles pueden ser menos claros al principio. El desarrollo de una visión consistente y compartida de las escalas a considerar implica la comunicación del alcance y las interacciones entre los sistemas individuales. Este proceso puede ayudar a identificar y abordar áreas que requieren la conciliación de diferentes puntos de vista que a menudo existen entre las partes interesadas. La conciencia de los problemas de escala probablemente evolucionará a medida que el modelado avance a través de las iteraciones. La elección de los caminos de modelado y el marco metodológico empleado está fuertemente fundada sobre este conocimiento [105].

La participación de las partes interesadas, incluidos los expertos en el dominio, a través de procesos participativos puede aportar conocimiento para hacer frente a la incertidumbre y la disponibilidad (deficiente) de datos [106, 107] con el fin de identificar las escalas relevantes. Las partes interesadas también pueden desempeñar un papel en la selección y combinación de datos y ayudar en el desarrollo del propósito del modelo.

El propósito y el uso de los modelos individuales pueden no coincidir si no se abordan las perspectivas conflictivas sobre el alcance del modelado. Los modeladores con diferentes objetivos en mente pueden considerar sólo las escalas relevantes para sus preocupaciones inmediatas (y a menudo específicas de cada disciplina), lo que lleva a una selección inadecuada de modelos individuales. Existe la posibilidad de obtener un alto grado de desajuste entre los modelos individuales, incluso si los modeladores coordinan sus esfuerzos. La concatenación de efectos inesperados a través de las escalas es común en los SdS [108].

El cambio de escala también puede ocurrir durante el proceso de modelado debido a nueva información que desencadena un cambio necesario en el contexto del modelo. La escala de las interacciones del modelo a representar también puede influir en el número y tipo de modelos individuales incluidos y en la complejidad general del sistema. Las elecciones con respecto a la escala tienen implicaciones sobre lo bien que pueden representarse las interacciones entre sistemas con respecto al propósito del modelo. La ampliación del alcance, en la que la escala del modelado se extiende continuamente para cubrir contextos no previstos originalmente, puede llegar a comprometer los esfuerzos de modelado, ya que los recursos disponibles se estiran demasiado para lograr un progreso significativo [109].

La elección de escalas se complica aún más en los casos donde los límites del sistema no pueden definirse claramente. Las zonas costeras, los sistemas atmosféricos y los sistemas de gestión de recursos naturales son ejemplos de sistemas con límites del sistema ambiguos. Los sistemas sociales y sus estructuras dinámicas son otro ejemplo que no tiene límites claros, pero que imponen importantes restricciones al comportamiento del sistema.

Generalmente, los enfoques participativos tienen como objetivo reunir los múltiples objetivos, problemas y preocupaciones de interés desde varias escalas y sistemas de gobernanza mediante el desarrollo de una relación mutuamente beneficiosa entre las partes interesadas [110]. La consideración cuidadosa de los problemas de transparencia, trazabilidad y gobernanza en los procesos de participación y compromiso [111, 112] será esencial para optimizar la relevancia, la legitimidad y la credibilidad del modelado de SdS [113].

La participación de una mayor diversidad de partes interesadas en dichos procesos permite desarrollar una representación más holística, cubriendo posibles puntos ciegos en la conceptualización del sistema y evitando el “aislamiento” del conocimiento [114, 115]. Sin embargo, incluir más perspectivas puede aumentar la complejidad del modelado y requiere una gestión cuidadosa de las expectativas y de los sesgos individuales [116]. La gestión de un SdS puede, en ocasiones, depender de la gestión efectiva de las partes interesadas y de su nivel de participación [94].

El aumento en la variedad de perspectivas también aumenta el potencial de conflicto entre equipos, miembros del equipo o partes interesadas. Por un lado, hay evidencia de que el conflicto juega un papel positivo en el aprendizaje y en el trabajo en equipo

efectivo [117]. Sin embargo, estos beneficios positivos pueden ocurrir sólo en casos donde hay altos niveles de confianza preexistente dentro del grupo y cuando el conflicto está relacionado con la tarea en lugar de ser personal. Por tanto, es necesario considerar las dinámicas de poder dentro de los equipos de modelado y las partes interesadas [118]. La identificación y el enfoque en objetivos que requieren que los participantes trabajen juntos es una base identificada hacia el éxito del proyecto y puede ayudar adicionalmente a evitar conflictos [117]. El diseño y la gestión cuidadosa de las interacciones entre equipos y partes interesadas requiere una consideración explícita de cómo los múltiples y, a veces, contradictorios objetivos podrían alinearse o conectarse. Los enfoques para la resolución y la prevención de conflictos son prometedores, pero son aún técnicas infrautilizadas. En general, los planes para la participación de las partes interesadas en el modelado de SdS deben considerar explícitamente los desafíos de escalado y diseñar estrategias para abordarlos.

7.5. Conceptualización del problema

La fase de conceptualización del problema implica construir la base de evidencia (por ejemplo, el conocimiento de expertos y de partes interesadas y la literatura, datos, modelos e hipótesis relevantes) para conceptualizar el problema o sistema, generalmente en un sentido cualitativo. Esto incluye identificar variables clave, indicadores, procesos, relaciones, entidades y escalas, así como métricas relacionadas con el rendimiento del modelo [119].

Al describir y capturar la esencia del sistema, el desarrollo del modelo conceptual ayuda con el diseño del modelo computacional subsiguiente, así como a concretar el propósito del modelo. Dos aspectos específicos de la escala a considerar son el enfoque utilizado para el desarrollo del modelo conceptual y la representación formal (por ejemplo, ecuaciones y especificaciones técnicas). Los procesos que se incluyen o excluyen en función de percepciones, prioridades, creencias y valores inevitablemente influirán en los datos considerados, las propiedades del modelo computacional y, por tanto, los caminos tomados.

Si no se pueden conciliar las diferencias en la comprensión conceptual de las escalas y sus interacciones, es posible crear múltiples modelos alternativos que representen las diferentes hipótesis que se pueden probar en etapas posteriores del proceso de modelado. Este enfoque también puede ayudar a evaluar la incertidumbre arraigada en las elecciones de construcción del modelo, ya que el tratamiento de la escala puede afectar los resultados y salidas del mismo. Aunque los diagramas conceptuales se pueden desarrollar sin especificar las escalas involucradas, la consideración explícita de la escala es valiosa para evitar

una mala interpretación de la conceptualización y garantizar que se incluyan variables y procesos clave. Un ejercicio útil, que no suele declararse pero que ayuda a la transparencia, es identificar qué enfoques alternativos se consideraron, o podrían haberse considerado, y cómo estos podrían haber afectado los resultados y las salidas, si se hubieran adoptado.

7.6. Formulación del modelo y evaluación

La fase de formulación y evaluación del modelo suele ser el elemento principal del proceso de su desarrollo, ya que incluye la descripción formal del modelo, su implementación en forma de software y las pruebas del mismo. Esta fase incluye la selección del enfoque de modelado (es decir, los tipos del modelo individual utilizados [95]), la construcción de su estructura, la calibración de parámetros, el análisis de incertidumbre y las pruebas y evaluación del modelo.

La transparencia en el proceso de recopilación de datos y la aprobación de los participantes en el modelado son necesarias para garantizar que los datos recopilados sigan siendo conceptualmente relevantes en todas las escalas. Además, la transparencia en el contexto de la recopilación y el uso de datos es un factor clave para establecer confianza entre las partes interesadas y los usuarios del modelo y la futura adopción de los modelos individuales [120]. Los datos pueden necesitar ser transformados para ser completamente relevantes para el contexto de su uso previsto, como la ampliación o reducción de escala para garantizar la compatibilidad con otros procesos. Idealmente, los metadatos incluirían información sobre la recopilación de datos, la incertidumbre y el proceso de transformación, lo que ayuda a determinar la idoneidad de los mismos para el modelo de SdS. Los descriptores explícitos de los datos de entrada y salida pueden ayudar a identificar el nivel adecuado de recopilación de datos con respecto a los recursos disponibles.

El sesgo del modelador puede tener un efecto acumulativo porque la elección de los datos recopilados, así como los metadatos que los describen, influyen en cómo se perciben las interacciones del sistema y, por tanto, en cómo se conceptualizan. Lo que puede considerarse irrelevante en una disciplina puede dictar los caminos de modelado en otra. En un entorno de SdS hay muchos más participantes involucrados y hay un alto grado de incertidumbre derivada de las decisiones tomadas como resultado.

La construcción de modelos computacionales de SdS requiere la integración de la experiencia en el dominio de las diversas disciplinas involucradas con el conocimiento técnico del desarrollo de software. Si bien el contexto general puede estar bien definido dentro de la fase de delimitación, es durante su construcción cuando los modelos individuales

y las escalas que representan se desarrollan, se acoplan, se prueban y se validan. Aquí, los modelos existentes pueden ser reutilizados o se pueden desarrollar unos nuevos. Los detalles de su inicialización, interoperación, método de ejecución y gestión de los datos involucrados deben determinarse y prototiparse en esta fase.

La integración conceptual de los modelos individuales puede beneficiarse de requerir que sean mecanicistas en lugar de cajas negras. Cuando un modelo se implementa como una caja negra, se vuelve difícil de evaluar y comprender. El modelado de SdS puede hacer uso de modelos individuales preexistentes que son reutilizados, lo que potencialmente implica la transferencia de las suposiciones, limitaciones y escala del modelo a un nuevo contexto. La idoneidad del modelo dentro de su contexto original no es necesariamente aplicable al nuevo contexto. La disponibilidad de código por sí sola, por ejemplo, no implica transparencia. Lo importante es la información contextual que se requiere para evaluar la idoneidad del propósito y la funcionalidad del modelo.

La integración técnica se refiere a la corrección de las interacciones del modelo, reconociendo la distinción entre la representación conceptual o abstracta y su implementación como software. Una correcta integración técnica de los modelos computacionales requiere disponer de la experiencia en ingeniería necesaria. Resultan consideraciones cruciales el hecho de que los modelos individuales interactúan y, en consecuencia, los errores se propagarán, y que cada modelo individual puede someterse a su propio ciclo de desarrollo separado, lo que inevitablemente requiere ajustes continuos.

La calibración es el proceso de ajuste de parámetros o alteración de las formas funcionales de ecuaciones o de relaciones para lograr el comportamiento deseado del modelo. En el modelado de SdS, problemas como la no identificabilidad y la “equifinalidad”, la regla de la dimensionalidad, la carga computacional y la representatividad de los datos pueden amplificarse [90].

La calibración del modelo dentro del paradigma de SdS [81] puede tomar tres enfoques generales: (1) calibración de cada modelo individual de forma independiente antes de la integración, (2) calibración de todos los modelos juntos después de la integración, o (3) una combinación de ambos. El primer enfoque es el más simple y directo, ya que cada modelo individual se calibraría dentro de su propio dominio. Aunque es pragmático, ignora el efecto de representar diferentes escalas en el SdS representado y las interacciones entre sistemas, lo que a su vez afecta el comportamiento del modelo y el rendimiento de los modelos individuales. El segundo enfoque es aparentemente el más completo para la calibración del modelo, ya que cada posible interacción entre modelos podría estar presente en el proceso de calibración del modelo. Se aprovecha el conocimiento multidisciplinar para garantizar que los valores

calibrados sean razonables para la operacionalización ampliada. Sin embargo, el enfoque tiene las siguientes barreras principales: (1) El espacio de búsqueda para la calibración del modelo será excesivamente grande. Además, pueden surgir nuevos efectos de interacción (posiblemente erróneos) entre los parámetros de un modelo y los de otro, especialmente con diferentes escalas de información, lo que hace que la superficie de respuesta sea extremadamente compleja para la calibración del modelo. El proceso de calibración podría volverse engorroso o inviable desde un punto de vista computacional. (2) Los datos disponibles con diferentes escalas pueden no ser suficientes para restringir adecuadamente el modelo en el proceso de calibración, ya que no son identificables a partir de los datos. También existe el riesgo de sobreajuste, ya que los datos disponibles pueden ser insuficientes para producir un modelo generalizado que cubra el dominio integrado. (3) El conocimiento experto para cada modelo puede tener restricciones de escala y puede no ser fácilmente transferible al dominio completo de SdS. En el tercer enfoque, los modelos se integran de uno en uno agregando complejidad de manera incremental para que la influencia de cada modelo individual pueda atribuirse directamente y se puedan abordar los problemas derivados. Aunque este enfoque puede ser tan pragmático como el primero y quizás tan completo como el segundo, la desventaja es el tiempo y el coste computacional requeridos para realizar el acoplamiento y la calibración de forma secuencial.

Los modelos de SdS a menudo se enfocan en dominios de problemas grandes (por ejemplo, [4]), lo que requiere modelos complejos para su evaluación y, por su naturaleza, tienen un alto grado de incertidumbre. Los enfoques cuantitativos tienen como objetivo medir el efecto de la incertidumbre en un parámetro específico, entrada o suposición sobre una salida y permiten la caracterización numérica de la distribución de la salida y, por tanto, el comportamiento del modelo [121, 122]. Sin embargo, la incertidumbre cualitativa no puede caracterizarse con un valor y surge de fuentes como los sesgos y creencias subjetivas de los actores humanos [115]. La incertidumbre cualitativa también puede surgir del juicio subjetivo de los modeladores, la imprecisión lingüística y el desacuerdo entre los involucrados [124, 125].

Un enfoque comúnmente sugerido para restringir la complejidad del modelo (y posiblemente el tiempo de ejecución) es filtrar los parámetros insensibles [126]. Se dice que estos parámetros tienen una influencia insignificante en la salida del modelo y pueden “fijarse” o hacerse estáticos en análisis posteriores o eliminarse del modelo. Otro enfoque es “atar” parámetros relacionados para que puedan representarse mediante un único “hiperparámetro” [127]. Sin embargo, reducir el número de parámetros no necesariamente equivale a una reducción de la incertidumbre. Más bien, puede simplemente significar que se determina que la consideración de una fuente de incertidumbre no

es importante para un contexto o propósito dado [126] y hacerlo puede comprometer la fidelidad del modelo en condiciones nuevas nunca antes vistas.

El uso de un modelo individual dentro de un modelo de SdS en lugar de su operación individual o su modificación o simplificación mediante la selección y el atado de parámetros constituye un cambio de contexto. Por tanto, los parámetros inicialmente considerados influyentes podrían volverse inactivos y no influyentes (y viceversa), o las relaciones que llevaron a que los parámetros se ataran pueden cambiar. El cambio de contexto también cambia la relevancia de las suposiciones y objetivos y lo que constituye un análisis de incertidumbre adecuado [128]. El análisis de incertidumbre realizado en un contexto no es válido en todas las escalas. Por tanto, la simplificación prematura puede afectar en última instancia a la idoneidad del modelo de SdS para su propósito general. Un análisis de sensibilidad completo en condiciones actuales y posiblemente alternativas puede proporcionar información valiosa sobre una pregunta clave: “¿cuándo y cómo importa la incertidumbre?” [129]. Una visión alternativa es que, dada la probabilidad de recursos computacionales limitados, los esfuerzos para caracterizar y comunicar las incertidumbres a las partes interesadas pueden ser más beneficiosos que un análisis de sensibilidad exhaustivo.

Las pruebas y la evaluación pueden ayudar en la evaluación de las ramificaciones de la elección de escala. En este paso se evalúa la razonabilidad de la estructura del modelo y la interpretabilidad de las relaciones dentro de los modelos, junto con el análisis tradicional del comportamiento del modelo. No todas las salidas producidas por los modelos individuales pueden ser relevantes para el propósito del modelo del SdS y la validez de sus salidas se ve afectada debido a la naturaleza integrada del modelado de SdS. Para que cualquier evaluación sea efectiva, las salidas específicas del modelo de interés que son relevantes para el propósito del modelo del SdS deben ser bien entendidas. Las salidas pueden estar en una escala espaciotemporal particular, por ejemplo, un promedio a largo plazo de una salida del modelo sobre un dominio espacial grande o un evento extremo en un punto específico. Los problemas también pueden surgir de la idoneidad conceptual de los modelos individuales, ya que la incertidumbre puede propagarse y acumularse a medida que se integran más modelos [131]. Por tanto, el primer paso en las pruebas y la evaluación implica intentar refutar aspectos de la estructura del modelo de SdS y las relaciones funcionales dentro del modelo basándose en su falta de correspondencia con el sistema representado y las salidas del modelo. Se podría involucrar a las partes interesadas para evaluar la alineación conceptual y la idoneidad de la representación de SdS en las escalas seleccionadas.

La evaluación de las relaciones de comportamiento a nivel integrado de SdS es similar a la prueba de hipótesis científicas o “pruebas conceptuales” [132], en las que se examinan las relaciones funcionales dentro del modelo de SdS. Estas pruebas pueden ser especialmente útiles en casos donde el funcionamiento interno de un modelo es inaccesible o desconocido, pero el comportamiento esperado del modelo individual en el contexto integrado puede caracterizarse [132]. Estos enfoques pueden utilizarse para identificar aspectos imposibles o implausibles de la salida del modelo de SdS. Si se puede demostrar que algún aspecto de la estructura del modelo o alguna relación funcional dentro del mismo es una representación inadecuada de los aspectos correspondientes del sistema real, entonces esa parte del modelo se refuta [133].

El siguiente paso se centra más específicamente en la correspondencia entre las proyecciones del modelo y los datos observados. Estrictamente hablando, los datos utilizados en las pruebas y evaluación del modelo deben ser independientes de los datos utilizados para su desarrollo [127]. Una variedad de métodos visuales, estadísticos y de aprendizaje automático se utilizan ampliamente para evaluar los modelos de SdS. Sin embargo, la elección del método debe basarse en las preguntas fundamentales sobre qué escenarios y qué observaciones utilizar en la evaluación. La evaluación de modelos bajo el rango de condiciones similares a las de interés puede ayudar a identificar las limitaciones del modelo.

El análisis de sensibilidad ahora se considera una práctica estándar en el modelado [126, 134, 135]. El objetivo es evaluar la sensibilidad del comportamiento del modelo del SdS a los cambios en sus modelos individuales y sus interacciones. Un problema derivado de la probable sobreparametrización de los modelos individuales es la equifinalidad y la falta de identificabilidad. La equifinalidad se refiere al fenómeno de diferentes implementaciones o combinaciones de la estructura del modelo, de los valores de parámetros y de sus interacciones que producen resultados igualmente aceptables [90]. La identificabilidad se refiere a la capacidad de atribuir la influencia en las salidas del modelo a parámetros o estructuras de modelo únicos [136]. Por tanto, cuanto mayor sea el número de parámetros, menos identificable se vuelve el modelo.

Las sensibilidades se evalúan como parte del análisis de identificabilidad, generalmente clasificando los parámetros según su influencia en las salidas, lo que puede ayudar a determinar qué parámetros requieren esfuerzos específicos para reducir la incertidumbre o mejorar la identificabilidad. La información del análisis de sensibilidad e identificabilidad puede ayudar a simplificar el modelo. Aplicar ingenuamente el análisis de sensibilidad e identificabilidad sin considerar el contexto del SdS puede afectar negativamente los resultados del modelado.

La evaluación de sensibilidades idealmente se basa en análisis globales, en lugar de locales. El uso de análisis de sensibilidad global en la evaluación de modelos ha tenido un uso creciente, a pesar de la falta de adopción o uso con conocimiento de las herramientas software disponibles para realizar dichos análisis [137].

7.7. Aplicación del modelo

La fase de aplicación del modelo implica experimentar o ejecutar el modelo utilizando, por ejemplo, escenarios de interés, seguido del análisis de las salidas y resultados del modelo. Esta fase también incluye la comunicación e interpretación de las ideas del modelo a los usuarios finales.

Un aspecto crítico en la aplicación de los modelos de SdS es que los modelos individuales típicamente evolucionan de forma independiente. El desarrollo de cada modelo individual, por necesidad, es liderado por expertos en la disciplina y pasa por ciclos de desarrollo separados y asíncronos. Dado que cada modelo puede provenir de diferentes paradigmas de modelado y fuentes de conocimiento, la implementación puede ajustarse con el tiempo o incluso ser reemplazada en respuesta a nuevos conocimientos adquiridos. Avanzar hacia aplicaciones de prueba del modelo utilizando el tipo y volumen de datos esperados lo antes posible, rápida y frecuentemente, permite a los modeladores encontrar problemas en la aplicación del modelo en etapas más tempranas del proceso. La experiencia adquirida con cada iteración sirve posteriormente para rectificar y protegerse frente a futuros desafíos de la aplicación. Por tanto, la aplicación del modelo requiere ser monitorizada y escrutada para asegurar que los modelos subyacentes (incluyendo sus metadatos, conocimiento representado y contexto de aplicación) se mantengan actualizados y resulten apropiados.

En casos de tiempo de ejecución prolongados, reemplazar los modelos individuales con más coste computacional por metamodelos puede ser una opción viable. Los metamodelos aproximan el comportamiento de entrada-salida del modelo original [138-140] y, por tanto, proporcionan representaciones simplificadas de modelos más complejos [141]. Los metamodelos aprovechan la simplicidad emergente de los sistemas complejos y, aunque hay una variedad de métodos disponibles para lograr esto, generalmente requieren que los modelos complejos (es decir, los modelos individuales originales) estén disponibles de antemano. Los metamodelos, al ser aproximaciones de la superficie de respuesta de un modelo original, son más relevantes para las condiciones existentes en los conjuntos de datos sobre los que se ajustan, por lo que se debe tener cuidado si se utilizan bajo condiciones que trascienden los márgenes de los datos. Forzar los sistemas con datos más allá de su capacidad son una preocupación particular. Si es posible,

la alternativa más pragmática y eficiente en recursos puede ser simplemente asignar más recursos computacionales (por ejemplo, supercomputadoras), especialmente considerando el tiempo necesario para investigar e implementar las opciones.

En el contexto de la gestión, donde los modelos de SdS se aplican típicamente, es necesario describir adecuadamente el nivel de incertidumbres en el modelo de SdS y sus predicciones [103]. Las partes interesadas individuales pueden reaccionar de manera diferente a las incertidumbres y a los niveles de incertidumbre. Presentar los resultados de los escenarios en relación con la línea base modelada reduce de manera ordenada los sesgos inherentes que surgen al depender de las preferencias de las partes interesadas para conformar los umbrales deseables, como suele ocurrir en los enfoques de análisis multicriterio o multiobjetivo [142, 143]. Con este enfoque, la aceptabilidad de un (posible) cambio relativo máximo o mínimo se convierte en el foco de la discusión de las partes interesadas.

Un requisito común compartido con las herramientas para realizar análisis (por ejemplo, para análisis de sensibilidad e incertidumbre y modelado exploratorio) es la provisión y definición de valores de parámetros. Estos pueden consistir en un valor "predeterminado", un rango dentro del cual los valores pueden variar, si estos valores son categóricos, escalares o considerados como constantes (se pueden encontrar ejemplos en [129, 144, 145]). Los valores categóricos pueden indicar la sustitución con otros tipos de datos o una colección de tipos de datos. Dicha información puede ser lo mínimo necesario para realizar análisis apropiados, reproducir y replicar resultados y apoyar la automatización posterior de estas actividades. Los valores de los parámetros, en efecto, representan dimensiones de escala y la selección inapropiada de sus valores y rangos puede dar lugar a resultados engañosos [146, 147].

7.8. Perpetuación del modelo

La fase final de perpetuación del modelo es relevante para los modelos que se utilizarán para apoyar la toma de decisiones continua o los procesos operativos para garantizar la mejora continua y su adopción a largo plazo. Implica proporcionar documentación a los usuarios para ejecutar el modelo e interpretar sus resultados, así como garantizar que existan planes y mecanismos para monitorear, mantener y actualizar adecuadamente el modelo. La revisión iterativa del modelo se logra a través de la colaboración continua entre los modeladores y los usuarios finales.

Cuando los modelos de SdS son utilizados por partes interesadas externas, se espera cierto nivel de soporte

técnico. Sin esto, el uso del modelo y, por tanto, su impacto probablemente será mínimo. Los modelos computacionales son software en el sentido de que están hechos de código, por lo que el uso continuo conlleva un coste básico para cubrir su mantenimiento, las mejoras y la actualización de la documentación. Dicha capacidad es crucial en contextos donde la gestión a largo plazo y el apoyo a la toma de decisiones son un requisito reconocido. En tales casos, el diseño, la implementación y la documentación del modelo deben planificar estas actividades a largo plazo desde el principio. En el contexto de SdS, esto implica retener el conocimiento multidisciplinar dentro de un equipo u organización (por ejemplo, [89, 103]).

La documentación es un conducto a través del cual se propaga la información y el conocimiento y proporciona el contexto necesario para la evaluación del modelo [111]. Sin una documentación suficiente, es difícil entender el contexto que llevó a cualquier problema específico, incluidos los desajustes entre los modelos individuales. La falta de contexto afecta a la validez percibida de la conceptualización del modelo y restringe el uso del mismo, lo que lo hace inapropiado o inválido para su propósito.

El acto de documentar permite, en sí mismo, una comunicación reflexiva y transparente y la obtención de nuevas ideas. Las suposiciones no documentadas con respecto a la escala y su influencia pueden comprometer otros modelos individuales, por lo que la falta de documentación puede bloquear la conciencia holística de los problemas del SdS. El mantenimiento y uso a largo plazo del modelo también pueden verse obstaculizados. Ningún individuo posee el conocimiento y la conciencia de los detalles del modelado en su totalidad y, mucho menos, de los efectos de las interacciones entre modelos. Por tanto, es importante reconocer que la redacción y el mantenimiento de la documentación deben ser un esfuerzo en equipo y se debe fomentar una cultura que apoye esto. En la práctica, hay pocos incentivos para documentar los modelos en tal medida. Un problema clave en la documentación del modelo de SdS es que los detalles de los modelos individuales importantes para el equipo del SdS pueden considerarse innecesarios para los equipos que desarrollan los modelos individuales. Una vez más, esto se debe a posibles desconexiones entre el propósito del modelo de SdS y los objetivos originales de cada modelo individual.

La evaluación del proceso en un SdS se centra en dos aspectos: el logro de los objetivos y la longevidad de los modelos. En términos de consecución de objetivos, la evaluación del proceso considera si los objetivos del modelo del SdS fueron respaldados por sus modelos individuales y, cuando corresponda, si los modelos individuales lograron sus propios objetivos. Aunque satisfacer los objetivos de los modelos individuales puede parecer un camino indirecto para satisfacer los objetivos

del modelo de SdS, esta interpretación es engañosa. Un enfoque de SdS para el modelado, en lugar de simplemente un enfoque de modelado múltiple, aprovecha la autonomía e independencia de los modelos individuales. Los modelos individuales deben ser capaces de producir sus propios resultados, independientemente de cómo se utilicen esos modelos en el contexto del modelo del SdS [148].

La evaluación de la longevidad del modelo de SdS, refiriéndose a la capacidad de aprovechar o reutilizar el modelo a lo largo del tiempo, requiere el desarrollo y la evaluación de un plan específico para su mantenimiento que incluya: (1) monitorizar la evolución de los modelos individuales; (2) identificar alternativas para los modelos que puedan perder su validez, disponibilidad o accesibilidad durante la vida útil del modelo del SdS; (3) establecer una estrategia para la evolución continua del modelo del SdS, incluyendo el desarrollo de posibles marcos de transformación e implementaciones; e (4) identificar oportunidades para facilitar el mantenimiento de los sistemas individuales alineados con el mantenimiento del modelo de SdS.

8. CONCLUSIONES

Este capítulo ha presentado una variedad de metodologías para alcanzar los objetivos de investigación. El análisis cualitativo proporcionó profundos conocimientos sobre los comportamientos y preferencias de los usuarios, permitiendo una comprensión integral de los factores subyacentes que influyen en los procesos de toma de decisiones. Se utilizó el modelado cuantitativo para predecir resultados con alta precisión, ofreciendo un marco robusto para analizar conjuntos de datos complejos. Además, se realizaron casos de estudio para ilustrar aplicaciones prácticas y validar modelos teóricos, cerrando la brecha entre la teoría y la práctica.

La combinación de estas metodologías permitió un enfoque holístico de la investigación, asegurando que los hallazgos fueran tanto fiables como aplicables en escenarios del mundo real. El análisis cualitativo destacó la importancia del contexto y las experiencias individuales, mientras que los modelos cuantitativos ofrecieron resultados generalizables que podrían aplicarse en diferentes entornos. Los casos de estudio proporcionaron ejemplos concretos que demostraron las implicaciones prácticas de la investigación, reforzando la validez de los constructos teóricos.

REFERENCIAS

1. A.M. Madni and M. Sievers (2018). "Model-based systems engineering: Motivation, current status, and research opportunities." *Systems Engineering*, vol. 21, no. 3, pp. 172-190. doi: 10.1002/sys.21438
2. M.W. Maier (1998). "Architecting principles for Systems of Systems." *Systems Engineering*, vol. 1, no. 4, pp. 267-284. doi: 10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D
3. S.C-Y. Lu, W., Elmaraghy, G. Schuh, and R. Wilhelm (2007). "A Scientific Foundation of Collaborative Engineering," *CIRP Annals*, vol. 56, no. 2, pp. 605-634. doi: 10.1016/j.cirp.2007.10.010
4. P.D. Collopy and P.M. Hollingsworth (2011). "Value-Driven Design." *Journal of Aircraft*, vol. 48, no. 3, pp. 749-759. doi: 10.2514/1.C000311
5. P.T. Grogan and A. Valencia-Romero (2019). "Strategic Risk Dominance in Collective Systems Design," *Design Science*, vol. 5, no. e24. doi: 10.1017/dsj.2019.23
6. M. Maier, "Architecting Principles for Systems of Systems," *Systems Engineering*, vol. 1, no. 4, p. 267, 1999.
7. A. Golkar and I. Lluch i Cruz, "The Federated Satellite Systems paradigm: Concept and business case evaluation," *Acta Astronautica*, vol. 111, no. 0, pp. 230-248, 6// 2015, doi: <http://dx.doi.org/10.1016/j.actaastro.2015.02.009>.
8. U. Pica and A. Golkar, "Sealed-Bid Reverse Auction Pricing Mechanisms for Federated Satellite Systems," *Systems Engineering*, vol. 20, no. 5, pp. 432-446, 2017, doi: 10.1002/sys.21395.
9. I. Lluch and A. Golkar, "Architecting federations of systems: A framework for capturing synergy," *Systems Engineering*, vol. 22, no. 4, pp. 295-312, 2019/07/01 2019, doi: <https://doi.org/10.1002/sys.21482>.
10. I. Lluch, "A framework for architecting federations of engineering systems," PhD, Space Center, Skolkovo Institute of Science and Technology, Moscow, Russia, 2017.
11. A. M. Ross and D. E. Hastings, "The Tradespace Exploration Paradigm," in *INCOSE 2005 International Symposium*, Rochester, NY, July, 2005 2005.
12. A. M. Ross, D. E. Hastings, J. M. Warmkessel, and N. P. Diller, "Multi-Attribute Tradespace Exploration as Front End for Effective Space System Design," *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 20-28, 2004/01/01 2004, doi: 10.2514/1.9204.
13. M. Kasunic and W. Anderson, "Measuring systems interoperability: Challenges and opportunities," *Software engineering measurement and analysis initiative*, 2004.
14. R. Akhtyamov, R. Vingerhoeds, and A. Golkar, "Identifying Retrofitting Opportunities for Federated Satellite Systems," *Journal of Spacecraft and Rockets*, vol. 56, no. 3, pp. 620-629, 2019, doi: 10.2514/1.A34196.
15. E. Stoll et al., "On-orbit servicing," *IEEE Robotics & Automation Magazine*, vol. 16, no. 4, pp. 29-33, 2009, doi: 10.1109/MRA.2009.934819.
16. M. E. Sosa, S. D. Eppinger, and C. M. Rowles, "Designing Modular and Integrative Systems," 2000. [Online]. Available: <https://doi.org/10.1115/DETC2000/DTM-14571>.
17. D. E. Hastings and C. Joppin, "On-Orbit Upgrade and Repair: The Hubble Space Telescope Example," *Journal of Spacecraft and Rockets*, vol. 43, no. 3, pp. 614-625, 2006/05/01 2006, doi: 10.2514/1.15496.
18. B. Ma, Z. Jiang, Y. Liu, and Z. Xie, "Advances in Space Robots for On-Orbit Servicing: A Comprehensive Review," *Advanced Intelligent Systems*, vol. 5, no. 8, p. 2200397, 2023/08/01 2023, doi: <https://doi.org/10.1002/aisy.202200397>.
19. Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: a survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352-375, 2018/01/01 2018, doi: 10.1504/IJWGS.2018.095647.
20. O. von Maurich and A. Golkar, "Data authentication, integrity and confidentiality mechanisms for federated satellite systems," *Acta Astronautica*, vol. 149, pp. 61-76, 2018/08/01/ 2018, doi: <https://doi.org/10.1016/j.actaastro.2018.05.003>.
21. F. Garcia and E. Rachelson, "Markov Decision Processes," in *Markov Decision Processes in Artificial Intelligence*, 2013, pp. 1-38.
22. D. Selva, A. Golkar, O. Korobova, I. L. i. Cruz, P. Collopy, and O. L. de Weck, "Distributed Earth Satellite Systems: What Is Needed to Move Forward?," *Journal of Aerospace Information Systems*, vol. 14, no. 8, pp. 412-438, 2017/08/01 2017, doi: 10.2514/1.1010497.
23. S. Briatore, N. Garzaniti, and A. Golkar, "Towards the Internet for Space: bringing cloud computing to space systems," 36th International Satellite Communications Systems Conference (ICSSC 2018), doi: doi:10.1049/cp.2018.1719

24. Schoonenberg, W.C.H., I.S. Khayal, and A.M. Farid, A Hetero-functional Graph Theory for Modeling Interdependent Smart City Infrastructure. 2019, Berlin, Heidelberg: Springer. 196.
25. Farid, A.M., D.J. Thompson, and W. Schoonenberg, A tensor-based formulation of hetero-functional graph theory. *Scientific Reports*, 2022. 12(1): p. 18805.
26. Little, J.C., R.O. Kaaronen, J.I. Hukkinen, S. Xiao, T. Sharpee, A.M. Farid, R. Nilchiani, and C.M. Barton, Earth Systems to Anthropocene Systems: An Evolutionary, System-of-Systems, Convergence Paradigm for Interdependent Societal Challenges. *Environmental Science & Technology*, 2023. 57(14): p. 5504-5520.
27. Brown, F.T., *Engineering System Dynamics*. 2nd ed. 2007, Boca Raton, FL: CRC Press, Taylor & Francis Group.
28. Karnopp, D., D.L. Margolis, and R.C. Rosenberg, *System dynamics: a unified approach*. 2nd ed. 1990, New York, NY: Wiley.
29. Paynter, H.M., *Analysis and design of engineering systems*. 1961: MIT Press.
30. Koenig, H.E., Y. Tokad, and H.K. Kesavan, *Analysis of Discrete Physical Systems*. 1967: McGraw-Hill.
31. Kuo, B.C., *Linear networks and systems*. 1967: McGraw-Hill.
32. Blackwell, W.A., *Mathematical modeling of physical networks*. 1968: Collier-Macmillan.
33. Shearer, J.L., A.T. Murphy, and H.H. Richardson, *Introduction to system dynamics*. 1967, Reading, UK: Addison-Wesley.
34. Chan, S.-P., S.-Y. Chan, and S.-G. Chan, *Analysis of linear Networks and Systems*. 1972: Addison-Wesley.
35. Forrester, J.W., *Industrial Dynamics – A Major Breakthrough for Decision Makers*. *Harvard Business Review*, 1958. 36(July-August): p. 37-66.
36. Sterman, J.D., *Business Dynamics: Systems Thinking and Modeling for A Complex World*. Vol. 19. 2000, Boston, MA, USA: Irwin/McGraw-Hill.
37. Newman, M., *Networks: An Introduction*. 2009, Oxford, UK: Oxford University Press.
38. Van Steen, M., *Graph Theory and Complex Networks: An Introduction*. 2010: Maarten van Steen.
39. Ghorbanichemazkati, E. and A.M. Farid, Generalizing Linear Graphs and Bond Graph Models with Hetero-functional Graphs for System-of-Systems Engineering Applications. 2025 In review.
40. Naderi, M.M., M.S. Harris, E. Ghorbanichemazkati, J.C. Little, and A.M. Farid, Convergent Anthropocene Systems-of-Systems: Overcoming the Limitations of System Dynamics with Hetero-functional Graph Theory. *Journal of Cleaner Production*, 2025 In review.
41. Delligatti, L., *SysML Distilled - A Brief Guide to the Systems Modeling Language*. 2014, Upper Saddle River, NJ: Addison-Wesley.
42. Friedenthal, S., A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. 2nd ed. 2011, Burlington, MA: Morgan Kaufmann.
43. Weillkiens, T., *Systems engineering with SysML/UML modeling, analysis, design*. 2007, Burlington, MA: Morgan Kaufmann.
44. Group, S.H.W., *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. 2015, International Council on Systems Engineering (INCOSE).
45. Hoyle, D., *ISO 9000 pocket guide*. 1998, Boston, MA: Butterworth-Heinemann.
46. Girault, C. and R. Valk, *Petri nets for systems engineering: a guide to modeling, verification, and applications*. 2013: Springer Science & Business Media.
47. Farid, A.M., Reconfigurability Measurement in Automated Manufacturing Systems, in *Engineering Department Institute for Manufacturing*. 2007, University of Cambridge.
48. Khayal, I.S. and A.M. Farid, Architecting a System Model for Personalized Healthcare Delivery and Managed Individual Health Outcomes. *Complexity*, 2018. 1(1): p. 1-25.
49. Schoonenberg, W.C.H. and A.M. Farid, A Dynamic Model for the Energy Management of Microgrid-Enabled Production Systems. *Journal of Cleaner Production*, 2017. 1(1): p. 1-10.
50. Farid, A.M., A Hybrid Dynamic System Model for Multi-Modal Transportation Electrification. *IEEE Transactions on Control System Technology*, 2016. PP(99): p. 1-12.
51. Farid, A.M., Multi-Agent System Design Principles for Resilient Coordination & Control of Future Power Systems. *Intelligent Industrial Systems*, 2015. 1(3): p. 255-269.
52. Farid, A.M., An engineering systems introduction to axiomatic design, in *Axiomatic Design in Large Systems: Complex Products, Buildings & Manufacturing Systems*, A.M. Farid and N.P. Suh, Editors. 2016, Springer: Berlin, Heidelberg. p. 1-47.

53. Viswanath, A., E.E.S. Baca, and A.M. Farid, An Axiomatic Design Approach to Passenger Itinerary Enumeration in Reconfigurable Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 2014. 15(3): p. 915-924.
54. Thompson, D.J. and A.M. Farid. Reconciling Formal, Multi-Layer, and Hetero-functional Graphs with the Hetero-functional Incidence Tensor. in 2022 17th Annual System of Systems Engineering Conference (SOSE). 2022.
55. Kivelä, M., A. Arenas, M. Barthelemy, J.P. Gleeson, Y. Moreno, and M.A. Porter, Multilayer networks. *Journal of complex networks*, 2014. 2(3): p. 203-271.
56. Park, G.-J. and A.M. Farid, Design of Large Engineering Systems, in *Design Engineering and Science*, N.P. Suh, M. Cavique, and J. Foley, Editors. 2021, Springer: Berlin, Heidelberg. p. 367-415.
57. Thompson, D., W.C.H. Schoonenberg, and A.M. Farid, A Hetero-functional Graph Resilience Analysis of the Future American Electric Power System. *IEEE Access*, 2021. 9: p. 68837-68848.
58. Thompson, D.J. and A.M. Farid, A hetero-functional graph structural analysis of the American Multi-Modal Energy System. *Sustainable Energy, Grids and Networks*, 2024. 38: p. 101254.
59. Khayal, I.S. and A.M. Farid, A Dynamic System Model for Personalized Healthcare Delivery and Managed Individual Health Outcomes. *IEEE Access*, 2021. 9: p. 1-16.
60. Schoonenberg, W.C.H. and A.M. Farid, Hetero-functional network minimum cost flow optimization: A hydrogen–natural gas network example. *Sustainable Energy, Grids and Networks*, 2022. 31: p. 100749.
61. Farid, A.M., Measures of reconfigurability and its key characteristics in intelligent manufacturing systems. *Journal of Intelligent Manufacturing*, 2017. 28(2): p. 353-369.
62. Farid, A.M. and D.C. McFarlane. A Development of Degrees of Freedom for Manufacturing Systems. in *IMS 2006 5th International Symposium on Intelligent Manufacturing Systems: Agents and Virtual Worlds*. 2006. Sakarya, Turkey.
63. Farid, A.M. and D.C. McFarlane, Production degrees of freedom as manufacturing system reconfiguration potential measures. *Proceedings of the Institution of Mechanical Engineers, Part B*, 2008. 222(10): p. 1301-1314.
64. Farid, A.M. and L. Ribeiro, An Axiomatic Design of a Multiagent Reconfigurable Mechatronic System Architecture. *IEEE Transactions on Industrial Informatics*, 2015. 11(5): p. 1142-1155.
65. Farid, A.M., Electrified transportation system performance: Conventional vs. online electric vehicles, in *The On-line Electric Vehicle: Wireless Electric Ground Transportation Systems*, N.P. Suh and D.H. Cho, Editors. 2017, Springer: Berlin, Heidelberg. p. 279-313.
66. Van der Wardt, T.J.T. and A.M. Farid A Hybrid Dynamic System Assessment Methodology for Multi-Modal Transportation-Electrification. *Energies*, 2017. 10, DOI: 10.3390/en10050653.
67. Khayal, I.S. and A.M. Farid, Axiomatic Design Based Volatility Assessment of the Abu Dhabi Healthcare Labor Market. *Journal of Enterprise Transformation*, 2015. 5(3): p. 162-191.
68. Farid, A.M., A Hetero-functional Graph Resilience Analysis for Convergent Systems-of-Systems. 2025 In review.
69. J Venkateswaran, YJ Son, *International Journal of Industrial Engineering*. 2004. 11, 151-159
70. Rathore, A., Balaraman, B., Zhao, X., Venkateswaran, J., Son, Y., and Wysk, R. (2005), "Development and Benchmarking of an Epoch Time Synchronization Method for Distributed Simulation," *Journal of Manufacturing Systems*, Vol. 24, No. 2, pp. 69–78.
71. Salado A, Kannan H., Szajnarfarber Z., Rouse W., Son Y., Nirav Merchant, A Reference Architecture for a Policy Test Laboratory.
72. IEEE. (2010). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules. *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000)*, 1-38. doi:10.1109/IEEESTD.2010.5553440
73. Rouse, W. B. (2019). *Computing Possible Futures: Model Based Explorations of "What if?"*. Oxford, UK: Oxford University Press.
74. Rouse, W. B. (2022). *Transforming Public-Private Ecosystems: Understanding and Enabling Innovation in Complex Systems*. Oxford, UK: Oxford University Press.
75. Singh, R., & Mathirajan, M. (2014, 9-12 Dec. 2014). A conceptual simulation framework for the performance assessment of lot release policies. Paper presented at the 2014 IEEE International Conference on Industrial Engineering and Engineering Management.
76. Hurt, T., McDonnell, J., & McKelvy, T. (2006, 3-6 Dec. 2006). *The Modeling Architecture for Technology, Research, and Experimentation*. Paper presented at the Proceedings of the 2006 Winter Simulation Conference.

77. Iwanaga, T., H.-H. Wang, S.H. Hamilton, V. Grimm, T.E. Koralewski, A. Salado, S. ElSawah, S. Razavi, J. Yang, P. Glynn, J. Badham, A. Voinov, M. Chen, W.E. Grant, T.R. Peterson, K. Frank, G. Shenk, C.M. Barton, A.J. Jakeman, and J.C. Little, Socio-technical scales in socio-environmental modeling: Managing a system-of-systems modeling approach. *Environmental Modelling & Software*, 2021. 135: p. 104885.
78. Jakeman, A.J., S. ElSawah, H.-H. Wang, S.H. Hamilton, L. Melsen, and V. Grimm, Towards normalizing good practice across the whole modeling cycle: its instrumentation and future research topics. *Socio-Environmental Systems Modelling*, 2024. 6: p. 18755.
79. Little, J.C., E.T. Hester, S. ElSawah, G.M. Filz, A. Sandu, C.C. Carey, T. Iwanaga, and A.J. Jakeman, A tiered, system-of-systems modeling framework for resolving complex socio-environmental policy issues. *Environmental Modelling & Software*, 2019. 112: p. 82-94.
80. Little, J.C., R.O. Kaaronen, J.I. Hukkinen, S. Xiao, T. Sharpee, A.M. Farid, R. Nilchiani, and C.M. Barton, Earth Systems to Anthropocene Systems: An Evolutionary, System-of-Systems, Convergence Paradigm for Interdependent Societal Challenges. *Environmental Science & Technology*, 2023. 57(14): p. 5504-5520.
81. Cash, D.W., W.N. Adger, F. Berkes, P. Garden, L. Lebel, P. Olsson, L. Pritchard, and O. Young, Scale and Cross-Scale Dynamics: Governance and Information in a Multilevel World. *Ecology and Society*, 2006. 11(2).
82. Ostrom, E., A diagnostic approach for going beyond panaceas. *Proceedings of the National Academy of Sciences*, 2007. 104(39): p. 15181-15187.
83. Kelly, R.A., A.J. Jakeman, O. Barreteau, M.E. Borsuk, S. ElSawah, S.H. Hamilton, H.J. Henriksen, S. Kuikka, H.R. Maier, A.E. Rizzoli, H. van Delden, and A.A. Voinov, Selecting among five common modelling approaches for integrated environmental assessment and management. *Environmental Modelling & Software*, 2013. 47(0): p. 159-181.
84. Macpherson, E., R.I. Cuppari, A. Kagawa-Viviani, H. Brause, W.A. Brewer, W.E. Grant, N.M. Herman-Mercer, B. Livneh, K.R. Neupane, T.N. Petach, C.N. Peters, H.-H. Wang, C. Pahl-Wostl, and H. Wheeler, Setting a pluralist agenda for water governance: Why power and scale matter. *WIREs Water*, 2024. 11(5).
85. Packett, E., N.J. Grigg, J. Wu, S.M. Cuddy, P.J. Wallbrink, and A.J. Jakeman, Mainstreaming gender into water management modelling processes. *Environmental Modelling & Software*, 2020. 127: p. 104683.
86. Melsen, L.A., It Takes a Village to Run a Model—The Social Practices of Hydrological Modeling. *Water Resources Research*, 2022. 58(2): p. e2021WR030600.
87. Sanz, D., J. Vos, F. Rambags, J. Hoogesteger, E. Cassiraga, and J.J. Gómez-Alday, The social construction and consequences of groundwater modelling: insight from the Mancha Oriental aquifer, Spain. *International Journal of Water Resources Development*, 2019. 35(5): p. 808-829.
88. Hamilton, S.H., C.A. Pollino, D.S. Stratford, B. Fu, and A.J. Jakeman, Fit-for-purpose environmental modeling: Targeting the intersection of usability, reliability and feasibility. *Environmental Modelling & Software*, 2022. 148: p. 105278.
89. Grimm, V. and U. Berger, Robustness analysis: Deconstructing computational models for ecological theory and applications. *Ecological Modelling*, 2016. 326: p. 162-167.
90. Lahtinen, T.J., J.H.A. Guillaume, and R.P. Hämäläinen, Why pay attention to paths in the practice of environmental modelling? *Environmental Modelling & Software*, 2017. 92: p. 74-81.
91. Voinov, A., R. Seppelt, S. Reis, J.E.M.S. Nabel, and S. Shokravi, Values in socio-environmental modelling: Persuasion for action or excuse for inaction. *Environmental Modelling & Software*, 2014. 53: p. 207-212.
92. Zare, F., J.H.A. Guillaume, A.J. Jakeman, and O. Torabi, Reflective communication to improve problem-solving pathways: Key issues illustrated for an integrated environmental modelling case study. *Environmental Modelling & Software*, 2020. 126: p. 104645.
93. MacLeod, M. and M. Nagatsu, What does interdisciplinarity look like in practice: Mapping interdisciplinarity and its limits in the environmental sciences. *Studies in History and Philosophy of Science Part A*, 2018. 67: p. 74-84.
94. Hamilton, S.H., S. ElSawah, J.H.A. Guillaume, A.J. Jakeman, and S.A. Pierce, Integrated assessment and modelling: Overview and synthesis of salient dimensions. *Environmental Modelling & Software*, 2015. 64: p. 215-229.
95. Kragt, M.E., B.J. Robson, and C.J.A. Macleod, Modellers' roles in structuring integrative research projects. *Environmental Modelling & Software*, 2013. 39: p. 322-330.
96. Tranquillo, J., *An Introduction to Complex Systems: Making Sense of a Changing World*. 2019: Springer Nature.
97. Sarosa, S. and A. Tatnall, Failure to Launch: Scope Creep and Other Causes of Failure from an Actor-Network Theory Perspective. *International Journal of Actor-Network Theory and Technological Innovation (IJANTTI)*, 2015. 7(4): p. 1-13.

98. Thompson, J.L., Building Collective Communication Competence in Interdisciplinary Research Teams. *Journal of Applied Communication Research*, 2009. 37(3): p. 278-297.
99. Cockerill, K., P. Glynn, I. Chabay, M. Farooque, R.P. Hämmäläinen, B. Miyamoto, and P. McKay, Records of engagement and decision making for environmental and socio-ecological challenges. *EURO Journal on decision processes*, 2019. 7(3-4): p. 243-265.
100. Glynn, P.D., A.A. Voinov, C.D. Shapiro, and P.A. White, From data to decisions: Processing information, biases, and beliefs for improved management of natural resources and environments. *Earth's Future*, 2017. 5(4): p. 356-378.
101. Cash, D.W., W.C. Clark, F. Alcock, N.M. Dickson, N. Eckley, D.H. Guston, J. Jäger, and R.B. Mitchell, Knowledge systems for sustainable development. *Proceedings of the National Academy of Sciences*, 2003. 100(14): p. 8086-8091.
102. Hoekstra, A., B. Chopard, and P. Coveney, Multiscale modelling and simulation: a position paper. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2014. 372(2021).
103. Hoekstra, A.G., S. Portegies Zwart, and P.V. Coveney, Multiscale modelling, simulation and computing: from the desktop to the exascale. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2019. 377(2142): p. 20180355.
104. Martin, D.M., S.J. Powell, J.A. Webb, S.J. Nichols, and N.L. Poff, An Objective Method to Prioritize Socio-Environmental Water Management Tradeoffs Using Multi-Criteria Decision Analysis. *River Research and Applications*, 2017. 33(4): p. 586-596.
105. Tjosvold, D., C. Hui, D.Z. Ding, and J. Hu, Conflict values and team relationships: conflict's contribution to team effectiveness and citizenship in China. *Journal of Organizational Behavior*, 2003. 24(1): p. 69-88.
106. De Dreu, C.K.W., The virtue and vice of workplace conflict: food for (pessimistic) thought. *Journal of Organizational Behavior*, 2008. 29(1): p. 5-18.
107. Bennett, N.D., B.F.W. Croke, G. Guariso, J.H.A. Guillaume, S.H. Hamilton, A.J. Jakeman, S. Marsili-Libelli, L.T.H. Newham, J.P. Norton, C. Perrin, S.A. Pierce, B. Robson, R. Seppelt, A.A. Voinov, B.D. Fath, and V. Andreassian, Characterising performance of environmental models. *Environmental Modelling & Software*, 2013. 40(Supplement C): p. 1-20.
108. Barba, L.A., Praxis of Reproducible Computational Science. *Computing in Science & Engineering*, 2019. 21(1): p. 73-78.
109. Saltelli, A., K. Aleksankina, W. Becker, P. Fennell, F. Ferretti, N. Holst, S. Li, and Q. Wu, Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices. *Environmental Modelling & Software*, 2019. 114: p. 29-39.
110. Zimmermann, H.J., An application-oriented view of modeling uncertainty. *European Journal of Operational Research*, 2000. 122(2): p. 190-198.
111. Chen, C.-F., H.-w. Ma, and K.H. Reckhow, Assessment of water quality management with a systematic qualitative uncertainty analysis. *Science of The Total Environment*, 2007. 374(1): p. 13-25.
112. Linkov, I. and D. Burmistrov, Model Uncertainty and Choices Made by Modelers: Lessons Learned from the International Atomic Energy Agency Model Intercomparisons. *Risk Analysis*, 2003. 23(6): p. 1297-1308.
113. Refsgaard, J.C., J.P. van der Sluijs, A.L. Højberg, and P.A. Vanrolleghem, Uncertainty in the environmental modelling process – A framework and guidance. *Environmental Modelling & Software*, 2007. 22(11): p. 1543-1556.
114. Pianosi, F., K. Beven, J. Freer, J.W. Hall, J. Rougier, D.B. Stephenson, and T. Wagener, Sensitivity analysis of environmental models: A systematic review with practical workflow. *Environmental Modelling & Software*, 2016. 79: p. 214-232.
115. Raick, C., K. Soetaert, and M. Grégoire, Model complexity and performance: How far can we simplify? *Progress in Oceanography*, 2006. 70(1): p. 27-57.
116. Song, X., J. Zhang, C. Zhan, Y. Xuan, M. Ye, and C. Xu, Global sensitivity analysis in hydrological modeling: Review of concepts, methods, theoretical framework, and applications. *Journal of Hydrology*, 2015. 523: p. 739-757.
117. Razavi, S., R. Sheikholeslami, H.V. Gupta, and A. Haghnegahdar, VARS-TOOL: A toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis. *Environmental Modelling & Software*, 2019. 112: p. 95-107.
118. Reichert, P., Towards a comprehensive uncertainty assessment in environmental research and decision support. *Water Science and Technology*, 2020. 81(8): p. 1588-1596.

- 119.** Dunford, R., P.A. Harrison, and M.D.A. Rounsevell, Exploring scenario and model uncertainty in cross-sectoral integrated assessment approaches to climate change impacts. *Climatic Change*, 2015. 132(3): p. 417-432.
- 120.** Iwanaga, T., D. Partington, J. Ticehurst, B.F.W. Croke, and A.J. Jakeman, A socio-environmental model for exploring sustainable water management futures: Participatory and collaborative modelling in the Lower Campaspe catchment. *Journal of Hydrology: Regional Studies*, 2020. 28: p. 100669.
- 121.** Li, G., Z. Bie, Y. Kou, J. Jiang, and M. Bettinelli, Reliability evaluation of integrated energy systems based on smart agent communication. *Applied Energy*, 2016. 167: p. 397-406.
- 122.** Norton, J., An introduction to sensitivity assessment of simulation models. *Environmental Modelling & Software*, 2015. 69: p. 166-174.
- 123.** Razavi, S. and H.V. Gupta, What do we mean by sensitivity analysis? The need for comprehensive characterization of “global” sensitivity in Earth and Environmental systems models. *Water Resources Research*, 2015. 51(5): p. 3070-3092.
- 124.** Guillaume, J.H.A., J.D. Jakeman, S. Marsili-Libelli, M. Asher, P. Brunner, B. Croke, M.C. Hill, A.J. Jakeman, K.J. Keesman, S. Razavi, and J.D. Stigter, Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environmental Modelling & Software*, 2019. 119: p. 418-432.
- 125.** Douglas-Smith, D., T. Iwanaga, B.F.W. Croke, and A.J. Jakeman, Certain trends in uncertainty and sensitivity analysis: An overview of software tools and techniques. *Environmental Modelling & Software*, 2020. 124: p. 104588.
- 126.** Castelletti, A., S. Galelli, M. Ratto, R. Soncini-Sessa, and P.C. Young, A general framework for Dynamic Emulation Modelling in environmental problems. *Environmental Modelling & Software*, 2012. 34: p. 5-18.
- 127.** Christelis, V. and A.G. Hughes, Metamodel-assisted analysis of an integrated model composition: An example using linked surface water – groundwater models. *Environmental Modelling & Software*, 2018. 107: p. 298-306.
- 128.** Pietzsch, B., S. Fiedler, K.G. Mertens, M. Richter, C. Scherer, eacute, dric, K. Widyastuti, M.-C. Wimmler, L. Zakharova, and U. Berger, Metamodels for Evaluating, Calibrating and Applying Agent-Based Models: A Review. *Journal of Artificial Societies and Social Simulation*, 2020. 23(2): p. 9.
- 129.** Asher, M.J., B.F.W. Croke, A.J. Jakeman, and L.J.M. Peeters, A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 2015. 51(8): p. 5957-5973.
- 130.** Maier, H.R., J.H.A. Guillaume, H. van Delden, G.A. Riddell, M. Haasnoot, and J.H. Kwakkel, An uncertain future, deep uncertainty, scenarios, robustness and adaptation: How do they fit together? *Environmental Modelling & Software*, 2016. 81: p. 154-164.
- 131.** Reichert, P. and M.E. Borsuk, Does high forecast uncertainty preclude effective decision support? *Environmental Modelling & Software*, 2005. 20(8): p. 991-1001.
- 132.** Kwakkel, J.H., The Exploratory Modeling Workbench: An open source toolkit for exploratory modeling, scenario discovery, and (multi-objective) robust decision making. *Environmental Modelling & Software*, 2017. 96: p. 239-250.
- 133.** Pianosi, F., F. Sarrazin, and T. Wagener, A Matlab toolbox for Global Sensitivity Analysis. *Environmental Modelling & Software*, 2015. 70: p. 80-85.
- 134.** Shin, M.-J., J.H.A. Guillaume, B.F.W. Croke, and A.J. Jakeman, Addressing ten questions about conceptual rainfall-runoff models with global sensitivity analyses in R. *Journal of Hydrology*, 2013. 503: p. 135-152.
- 135.** Wagener, T. and F. Pianosi, What has Global Sensitivity Analysis ever done for us? A systematic review to support scientific advancement and to inform policy-making in earth system modelling. *Earth-Science Reviews*, 2019. 194: p. 1-18.
- 136.** Salado, A. Abandonment: A natural consequence of autonomy and belonging in Systems of Systems. in 2015 10th System of Systems Engineering Conference (SoSE). 2015.
- 137.** Baldwin C., Sauser B., and Cloutier R., “Simulation Approaches for System of Systems: Event-based versus Agent Based Modeling, *Procedia Computer Science*, 44, pp. 363-372, 2015.
- 138.** Acheson P., Dagli C., and Kilicay-Ergin N., “Model Based Systems Engineering for System of Systems Using Agent-Based Modeling” *Procedia Computer Science*, 16, pp. 11-19, 2013.
- 139.** Wei Y., and Madey G., “Agent-based Simulation for UAV Swarm Mission Planning and Execution”, *Proceedings of the Agent-Directed Simulation Symposium*, Society for Computer Simulation International, 2013.

140. Macal C., and North M., "Agent-based Modeling and Simulation" Proceedings of the 2009 Winter Simulation Conference, pp. 86-98, 2009.
141. Mour A., Kenley R., Davendralingam N., and DeLaurentis D., "Agent-based Modeling for Systems of Systems" INCOSE International Symposium, pp. 973-976
142. Silva R., and Braga R. "Simulating System-of-Systems with Agent-based Modeling: A Systematic Literature Review" IEEE Systems Journal, Vol. 14, No. 3, pp. 3609-3617, 2020.
143. Von Neumann, John. Theory of Self-Reproducing Automata. Ed. Arthur W. Burks. Urbana: University of Illinois Press, 1966.
144. Games, M., (1970) The fantastic combinations of John Conway's new solitaire game "life" by Martin Gardner. Scientific American, 223, pp.120-123.
145. Schelling, T. C. (1971). "Dynamic models of segregation". The Journal of Mathematical Sociology. 1 (2), pp. 143- 186.
146. Epstein J., and Axtell R., Growing Artificial Societies: Social Science from the Bottom Up, Brookings Institution Press, 1996.
147. Axelrod R., The Evolution of Cooperation, New York: Basic Books, 1984.
148. Borshchev A. and Filippov A., "From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools" The 22nd International Conference of the System Dynamics Society, July 25-29, 2004, Oxford, England.

JOSÉ LUIS DE ROSARIO SÁNCHEZ-SIMÓN

José Luis de Rosario Sánchez-Simón es Ingeniero de Minas especializado en Energía y Combustibles por la Universidad Politécnica de Madrid. También tiene un Executive MBA del Instituto de Empresa. Actualmente, es profesor en la Universidad Europea de Madrid, donde imparte clases de Gestión de Operaciones en los programas de Máster en Ingeniería Industrial y Organización de la Ingeniería, Gestión de Proyectos y Negocios.



También trabaja en Isdefe como Ingeniero de Sistemas especializado en estrategia, mejora de procesos y sistemas de aseguramiento de la calidad. Fue el ganador del concurso de I+D+i por su proyecto de cuadro de mando integral. Anteriormente, trabajó en ING como analista de negocios responsable de los informes de operaciones en España. En IBM, fue controller financiero para Francia e Italia, gestionando el análisis de ingresos, gastos y beneficios para cada línea de negocio. En la Fundación ONCE, lideró el aseguramiento de la calidad. Implementó sistemas de gestión de almacenes basados en RFID en LTR. En STEF, fue especialista en aseguramiento de la calidad y logística para Procter & Gamble en las plataformas de Madrid, Barcelona, Lisboa y Canarias. También está comprometido con el apoyo a personas con discapacidades, compartiendo su experiencia a través de charlas en hospitales, universidades y asociaciones.

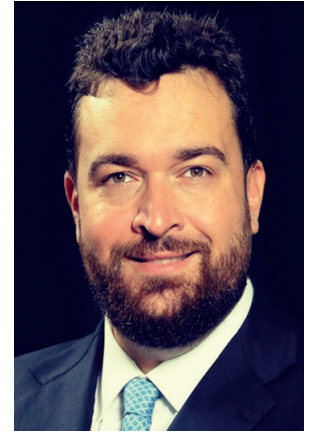
DR. PAUL GROGAN

El Dr. Paul Grogan es profesor titular en la Escuela de Computación e Inteligencia Aumentada de la Universidad Estatal de Arizona. Tiene un doctorado en sistemas de ingeniería y un máster en aeronáutica y astronáutica del Instituto Tecnológico de Massachusetts, así como un grado en mecánica de ingeniería y astronáutica de la Universidad de Wisconsin. Dirige el Laboratorio de Diseño Colectivo, que desarrolla y estudia el uso de métodos y herramientas basados en información para el diseño de sistemas terrestres y espaciales con arquitecturas distribuidas o descentralizadas.



DR. ALESSANDRO GOLKAR

El Dr. Alessandro Golkar es Catedrático en la Universidad Técnica de Múnich, Cátedra de Sistemas de Naves Espaciales, dentro del Departamento de Ingeniería Aeroespacial y Geodesia. El Dr. Golkar se centra en la ingeniería de sistemas espaciales, la gestión de tecnología y el emprendimiento espacial. Anteriormente, ocupó cargos en la industria y la universidad, incluyendo Vicepresidente en la organización de Planificación y Desarrollo Tecnológico del CTO de AIRBUS (Toulouse, Francia) y Director del Centro Espacial en Skoltech (Moscú, Rusia). Es el autor principal del curso en línea sobre la Nueva Economía Espacial de MIT Professional Education en los EE.UU., donde trabaja como Instructor Senior Visitante. El Dr. Golkar tiene un doctorado en Aeronáutica y Astronáutica del MIT, un MBA Ejecutivo del Instituto de Tecnología Qantic, y títulos de Máster y Licenciatura en ingeniería aeroespacial de la Universidad de Roma “La Sapienza”.



DR. AMRO M. FARID

El Dr. Amro M. Farid es el Catedrático de la Cátedra Alexander Crombie Humphreys en Economía de la Ingeniería en el Departamento de Sistemas y Empresas del Instituto de Tecnología Stevens. Es el Científico Principal de Sistemas en el Centro Nacional de Análisis Energético de CSIRO, la Agencia Nacional de Ciencia de Australia. También es Científico Visitante en el Departamento de Ingeniería Mecánica del MIT y CEO de Engineering Systems Analytics LLC. En Stevens, lidera el Laboratorio de Redes Inteligentes Integradas de Sistemas de Ingeniería (LIINES) y ha publicado más de 165 artículos revisados por pares en Redes Eléctricas Inteligentes, Nexus de Hidrógeno-Energía-Agua, Sistemas de Transporte Electrificados, Gestión Energética de la Producción Industrial y la Cadena de Suministro, Ciudades Inteligentes, Regiones y Naciones.



DR. YOUNG-JUN SON

El Dr. Young-Jun Son es el James J. Solberg Head y Catedrático Ransburg de la Edwardson School of Industrial Engineering en la Universidad de Purdue. Antes de este puesto, fue Jefe de Departamento y Catedrático de Ingeniería de Sistemas e Industrial en la Universidad de Arizona. Su investigación se centra en un modelo de simulación y decisión basado en datos y a múltiples escalas, necesario para el diseño y control en diversas aplicaciones, incluyendo la empresa manufacturera extendida, la red de energía renovable y almacenamiento, la seguridad nacional, el transporte y la red social. Ha sido autor/coautor de más de 110 artículos en revistas y 100 artículos en conferencias. Es Fellow del Instituto de Ingenieros Industriales y de Sistemas (IISE) y ha recibido el Premio al Joven Ingeniero Destacado de la Sociedad de Ingenieros de Manufactura (SME) en 2004, el Premio al Joven Ingeniero Destacado del IIE en 2005, los Premios al Mejor Artículo en la Reunión Anual del IISE (2005, 2008, 2009, 2016, 2018, 2019), y el Premio al Mejor Artículo del Año en 2007 de la International Journal of IE. Sus actividades de investigación han sido financiadas por NSF, AFOSR, DOT/FHWA, el Departamento de Energía de EE.UU./Autoridad de Comercio de Arizona, USDA, NIST, Sandia National Lab, la Fundación Científica de Arizona, Boeing, Samsung, Motorola, Raytheon, Tucson Electric Power, Microsoft y varias empresas de software de aplicaciones. Es Editor de Departamento para IISE Transactions, en el consejo editorial de siete revistas adicionales. Fue vicepresidente y secretario del Grupo de Desarrollo de Productos de Datos de Simulación de Manufactura Central (CMSD) de SISO. Actualmente sirve en 1) el consejo de la Winter Simulation Conference, y ha servido en 2) el consejo del IISE como Vicepresidente de Educación Continua y 3) como miembro del Comité de Reuniones de INFORMS. Ha sido co-Presidente del Programa para ISERC 2007, Presidente General de la Reunión Anual de INFORMS 2018, y Presidente General de la Winter Simulation Conference 2019.



DR. NIL H. ERGIN

La Dra. Nil H. Ergin es la Profesora a cargo de los Programas de Ingeniería de Sistemas y Gestión de Ingeniería en la Universidad Estatal de Pensilvania y Profesora Titular de Ingeniería de Sistemas en la Escuela de Estudios Profesionales de Posgrado de Penn State Great Valley. Obtuvo su doctorado en ingeniería de sistemas y su maestría en gestión de ingeniería en la Universidad de Missouri-Rolla (actualmente conocida como Universidad de Ciencia y Tecnología de Missouri) y su grado en ingeniería ambiental en la Universidad Técnica de Estambul. Antes de unirse a la Universidad Estatal de Pensilvania, la Dra. Ergin trabajó en el Instituto de Investigación de Sistemas de Manufactura e Ingeniería (RIMES) en la Universidad de Texas en El Paso, donde participó en contratos de investigación financiados por la industria y enseñó en el programa de posgrado en ingeniería de sistemas. La investigación de la Dra. Ergin integra principios clave de la ciencia de sistemas, teorías de resolución de problemas, modelos de múltiples agentes y modelos de conocimiento para el análisis de problemas complejos en aplicaciones de sistemas de sistemas y sistemas adaptativos complejos. Es miembro senior de IEEE y miembro de INCOSE.



DR. JOHN LITTLE

El Dr. John Little obtuvo una licenciatura en Ingeniería Química de la Universidad de Ciudad del Cabo y una maestría y un doctorado en Ingeniería Ambiental de la Universidad de California, Berkeley. Después de completar un postdoctorado en el Laboratorio Nacional Lawrence Berkeley, se unió al Departamento de Ingeniería Civil y Ambiental de Virginia Tech, y actualmente es Catedrático Charles E. Via, Jr. La investigación de John se centró anteriormente en la dinámica de procesos en sistemas ambientales (edificios y ambientes interiores, y calidad del agua en lagos y embalses), pero ahora se ha ampliado a la dinámica de procesos en sistemas del Antropoceno. Debido a que los sistemas del Antropoceno son altamente interdependientes y evolucionan dinámicamente, a menudo con tasas aceleradas de evolución cultural y tecnológica, la familia resultante de desafíos sociales (por ejemplo, cambio climático, energía renovable, infraestructura adaptativa, desastres, pandemias, inseguridad alimentaria, pérdida de biodiversidad, desarrollo sostenible, resiliencia y equidad) también son altamente interdependientes y necesitan ser enmarcados y abordados de manera holística. Para catalizar las transformaciones sociales necesarias a escala local, urbana, regional y global, se necesita un paradigma de convergencia evolutiva de sistemas de sistemas. El Dr. Little recibió un Premio de Carrera de la Fundación Nacional de Ciencias en 1996, fue elegido miembro de la Academia de la Sociedad Internacional de Calidad del Aire Interior y Clima en 2008, recibió el Premio a la Mejor Tesis Doctoral de la Asociación de Profesores de Ingeniería y Ciencias Ambientales en 2011, y el Premio al Mérito Técnico de la Sociedad de Gestión de Lagos de América del Norte en 2014. El Dr. Little ha sido profesor visitante en la Universidad de Sydney, Australia; el Instituto Federal Suizo de Ciencia y Tecnología Acuática (Eawag), Suiza; la Universidad de Tsinghua, China; la Universidad Nacional Cheng Kung, Taiwán; la Universidad de Granada, España; el Centro Científico y Técnico del Edificio, Francia; y la Universidad de La Rochelle, Francia.





Isdefe



Isdefe
C/ Beatriz de Bobadilla, 3
28040 Madrid
Tel.: +34 91 411 50 11
Email: general@isdefe.es
www.isdefe.es