

NEWSLETTER

DICIEMBRE 2022, NÚMERO 5



EN ESTE NÚMERO:

CARTA DE LA PRESIDENTA	2
THE 4D PARADIGM OF WBS.....	4
SISTEMISTAS, APRENDER ENSEÑANDO.....	7
METODOLOGÍA MBSE ISE&PPOA APLICADA A LA DEFINICIÓN DE ARQUITECTURA FUNCIONAL DE UN SISTEMA.....	9
GRUPOS DE TRABAJO TÉCNICOS.....	19
CERTIFICACIÓN SEP	23
SAVE THE DATE.....	27
CÓMO HACERSE MIEMBRO	26

CARTA DE LA PRESIDENTA:



Anabel Fraga
PRESIDENTA - AEIS INCOSE

¡FELIZ NAVIDAD Y PROSPERO AÑO NUEVO 2023! ¡QUÉ TENGAMOS UN BUEN AÑO!

¡Los años se nos pasan en un abrir y cerrar de ojos, ya son dos años que lanzamos la primera Newsletter de la Asociación!

Continuamos sumergidos en una época compleja, los virus están a la orden del día pero confío en que hemos aprendido la lección y que aplicaremos lo mejor que hemos sacado de estos años complejos.

Mantenemos nuestra creencia en que las personas son lo primero, las que dan sentido a lo que hacemos y las que forjan las empresas que nos apoyan. Gracias por seguir con nosotros, por apoyarnos y por el gran grupo de voluntariado que hemos creado este año, así como a todas las organizaciones que ya forman parte del CAB de AEIS.

En una alegría contaros que este año hemos crecido, en participación, en número de CAB, en miembros, en compromiso y la asociación ha madurado con y para la gente que nos apoya.

Hemos sido los anfitriones de diversos eventos mensuales, los organizadores principales del EMEA WSEC 2023 que se realizará en Sevilla del 24 al 26 de Abril del 2023. Si, coincidiendo con la Feria de Sevilla. Un doble de valor para disfrutar de un evento singular y una semana preciosa de la tradición Española. Y como no, en la ciudad que será la sede de la Agencia Espacial Española. ¡Esperamos veros allí!

Queremos agradecer todo vuestro apoyo y seguimiento durante este tiempo, y sobre todo este año cuando nuestros CABs han permanecido y crecido a nuestro lado, y vosotros como miembros de igual manera. Y como suelo decir todos los años: ¡Ya tenemos nuestro mayor regalo para estas fechas! ¡Vosotros!

Finalmente, queremos desearos unas Felices Fiestas, una navidad llena de paz, salud, y alegría. Que este nuevo año 2023 venga lleno de prosperidad, aprendizajes, y reflexiones ante las situaciones vividas, que nos ayuden a ser mejores personas y profesionales.

¡FELIZ NAVIDAD Y FELIZ AÑO NUEVO 2023!

Un fuerte abrazo de parte de toda la familia de INCOSE España,

THE 4D PARADIGM OF WBS

Aleksandr Turkhanov

*Product and Program Management,
Brownfield Digital Engineering of
Software-Intensive Systems
APPLIED KNOWLEDGE SYSTEMS
LIMITED*

Recently (Halligan, 2022) **has given some critics and helpful guidance on the WBS concept**. His line of argument in the article was that **WBS does not use a whole-part decomposition of the project's work**: "Let's get one thing straight. A "Work" Breakdown Structure (WBS) isn't a breakdown of work. WBS is a bad name." To address this weakness, the author proposes to use a project breakdown structure: "The PBS is a hierarchical breakdown which ... is strongly oriented towards the physical structure of the end products to be delivered by the project. Under product orientation, products, and services needed to bring ... the end and intermediate products into existence" In this paper, I analyze the proposed concept of PBS and offer some clarifications based on the four-dimensional paradigm of the total project work. I show that work and project breakdown structures are different representations of these efforts; **in the 4D viewpoint, they do not contradict each other**.

The 100% rule

WBS is a controversial and important subject. The PMI.org website alone will provide 1,300 papers on it. There were successful attempts to use WBS to manage software-intensive development (Tausworthe, 1979), and many failed ones. For the latter, we don't need to mention the sources; many practitioners have such experience to share. That's partly because discussing WBS structures and codes, and their application has come a long way (Halligan, 2022; Mueller, 2000; Okayama and Chirillo, 1980), but it is far from resolution.

One such contradiction is the decomposition of work. In project management software, we often find this implementation Halligan challenges in his article.

**LET'S GET ONE THING
STRAIGHT. A "WORK"
BREAKDOWN
STRUCTURE (WBS)
ISN'T A BREAKDOWN
OF WORK. WBS IS A
BAD NAME.**

Each task and group task will have a WBS code, and the WBS view will be constructed from that hierarchical structure of task names. Tool developers reproduce such an approach even in new products like ClickUp, leaving the critique that has existed for 40 years unanswered. Before offering the solution, consider the 100% rule as *ISO 21511:2018 Work breakdown structures for project and programme management* standard explains it. The standard provides valuable insights into the analysis of the problem with the definition of WBS:

“4.4 ... The hierarchical decomposition should include 100% of the work contained in the scope of the project or programme. Where **an element is decomposed to child elements**, the aggregate of work defined by the lower elements should represent 100% of the work in the parent element. ...

100% rule. The work breakdown structure captures the work required to achieve the project scope. The 100% rule applies to the parent and child elements. The child level of decomposition of a work breakdown structure element represents 100% of the work applicable to the parent level.”

My point is that there are two ways to follow the 100% rule, and we should not mistake one for another.

First, I want to draw your attention to the type of relationships between WBS elements the standard introduces. **It is not a structural “composition,” a whole-part breakdown, or a de-composition. It is a “parent-child” type of relationship, altogether another hierarchical structure, a causal graph** (Pearl and Mackenzie, 2018) **with a single “head” node.** I read it exactly: we need a parent (a reason) to produce a child (an outcome). A child is not a part of a parent, and thus lower-level elements of WBS are not parts of upper-level elements. And that completely changes the perception of the 100% principle. Let’s see an example that shows the difference between composition and parent-child relationships.

Two ways to follow the 100% rule

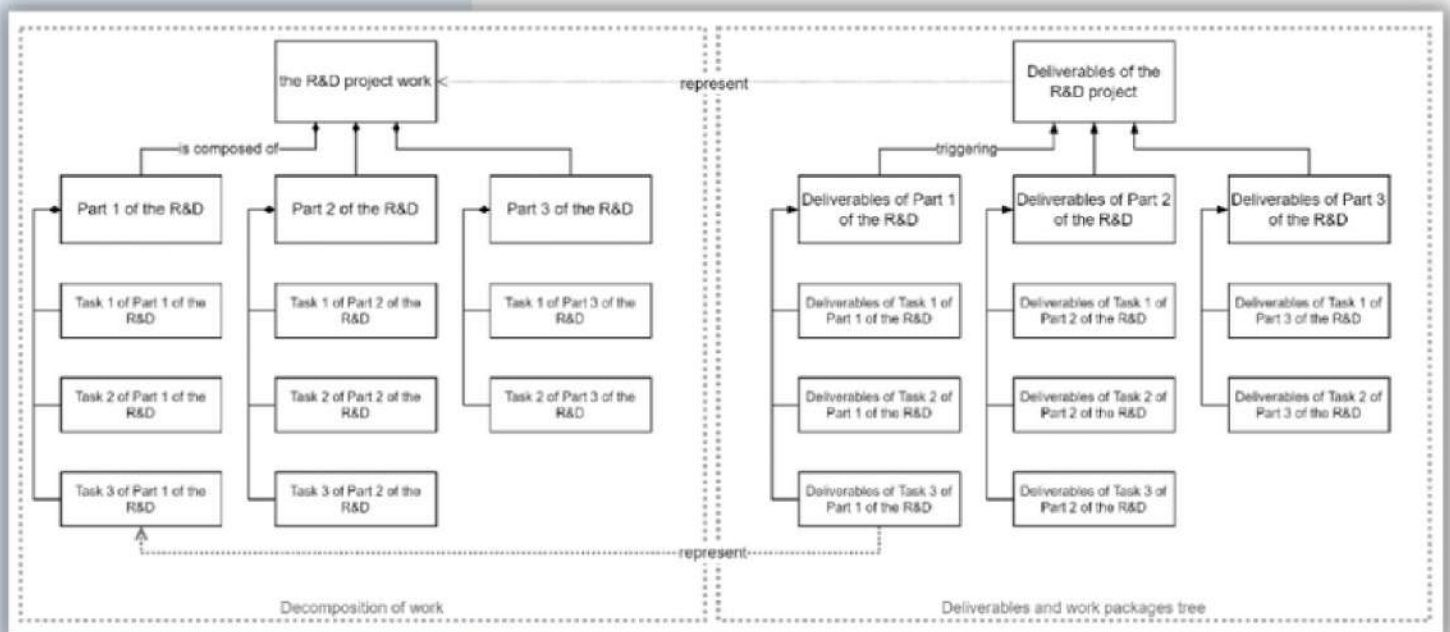
Let’s consider an example of a software-intensive system for conducting certification tests of unmanned aircraft systems. This system consists of the following:

- a plot of land with a test flight center building,
- engineering networks and facilities,
- drone surveillance towers,
- and a server room with equipment.



There is also the project as a temporal organization that creates the system and all its parts. As the project is temporal, we can consider it a four-dimensional whole-part decomposition, the same way we consider plant construction or operation in the ISO 15926 series of standards or other engineering data modeling methods (Partridge, 2021). And as we may represent the R&D project as a four-dimensional whole-part decomposition, then we can allocate each part of the whole project work to the part of the system:

The work on the purchase and laying of the cable is part of the work on the creation of the testing ground, as well as writing code and assembling the next version of the software. The selection, purchase, and installation of a blade server in a rack with the installation of Linux on it and the configuration of CI/CD is part of the work on creating the system. Without a cable, software, or server, the system will not be in complete configuration. The project work cannot be completed fully without cable laying, software development, and server installation. The whole work performed in the project and described in the scope of work also follows the 100% rule through decomposition.



But what about the working products that represent these works? Is it possible to say that a cable log, measurement test report for insulation resistance, and hangers marking the laid cable are part of the working products of the overall R&D? Unlikely. Is it possible to say the same about a commit in the Git repository, a software builds log and a progress report? It is not possible. Will we include the protocol of the tender commission for the supplier nomination and an extract from the inventory list for a blade server in the final set of documents for the project termination report? I doubt it very much.

Deliverables are not in whole-part relationships. Also, without the completed "child deliverables," you cannot commit "parent deliverables." Thus, the WBS tree structure is a taxonomic index to navigate the deliverables. This index does

exist without work decomposition shown in the project statement of scope and the WBS dictionary. And the WBS consists of both – the taxonomical index and the work structure; they represent reality in different ways.

Discussion

If we take such a viewpoint, the metamodel of WBS becomes simple – objects have type “work packages,” and relationships between these objects will be “triggering,” as there are causal relationships between “parents” and “children.” And we can follow the 100% principle either by decomposing the whole work into its parts or accounting for all the required reasons (children) to produce the required outcome (parent). Both methods will produce the complete scope of work.

- We can see the signs of ongoing methodological discussions on the nature of WBS and PBS in the [ArchiMate specification](#), which allows two work packages to be in specialization, composition, flow, and triggering relationships. This viewpoint is nothing new, but the main question persists – how do we show these decompositions in the project management software in an intuitive way?

Bibliography

- Halligan, R.J., 2022. Building a Great “Work” Breakdown Structure. PPI SyEN.
- Mueller, S.E., 2000. Resolving work breakdown structure problems. AACE Int. Trans. PS2A.
- Okayama, Y., Chirillo, L.D., 1980. Product work breakdown structure. Maritime Administration Washington DC.
- Partridge, C., 2021. A 4-Dimensionalist Top Level Ontology (TLO): Mereotopology and Space-Time. Presented at the Newton Gateway to Mathematics, INI Seminar Room 1.
- Pearl, J., Mackenzie, D., 2018. The book of why: the new science of cause and effect. Basic books.
- Tausworthe, R.C., 1979. The work breakdown structure in software project management. J. Syst. Softw. 1, 181–186. [https://doi.org/10.1016/0164-1212\(79\)90018-9](https://doi.org/10.1016/0164-1212(79)90018-9)



SISTEMISTAS, APRENDER ENSEÑANDO



Luis Andrés Olmedo
Senior Systems Engineer - INDRA
Presidente electo - AEIS-INCOSE
Fundador - SISTEMISTAS

¡Hola sistemista! ¿cómo estás? Espero que 2022 haya sido un año personal y profesionalmente provechoso y deseo que en 2023 puedas conseguir las metas que con dedicación persigas. Hablando de lo personal y lo profesional, a veces dudo de que hacer esa separación sea sano. Trabajando con americanos hace un tiempo aprendí que ellos – al menos los que yo conocí - no tienen una separación real entre ambos ámbitos, son la misma persona dentro y fuera del trabajo, mezclan tareas de un ámbito en el otro y viceversa... y creo que es bueno, aporta coherencia (y flexibilidad) al conjunto de la vida de una persona. Pero no quiero filosofar, sólo divagaba un poco.

Empiezo por lo importante, sentirse agradecido, así que muchas gracias a cada uno de los invitados del podcast por dedicarme un rato y compartir su conocimiento. Gracias a los que ya habéis pasado por el podcast y a los que estáis por pasar. Y gracias a AEIS por darme la oportunidad de contar por qué empecé con #Sistemistas, qué motivos tuve para empezar, qué objetivos persigo y hasta cuándo durará. ¡Gracias!

¿Por qué?

Soy el eterno aprendiz, tengo curiosidad por muchas cosas, en algunos casos que nada tienen que ver con la tecnología, como la salud o los oficios tradicionales. En otros momentos de mi vida he tenido podcasts y blogs relacionados con esos temas... así que después de unos años (no muchos) en INCOSE, leyendo y aprendiendo de fuentes muy diversas, participando en grupos de trabajo internacionales, viendo vídeos y escuchando podcast



**EMPIEZO POR LO
IMPOTANTE,
SENTIRSE
AGRADECIDO**

de los mejores, me di cuenta de una cosa: casi todo estaba en inglés. Pero sobre todo me di cuenta de que en español había (y hay) muy poco para la cantidad de hablantes que hay en el mundo y la cantidad de grandes profesionales de la Ingeniería de Sistemas que tenemos.

Así que junté mis ganas de seguir aprendiendo de Ingeniería de Sistemas y mis experiencias anteriores con podcasts/blogs y nació SISTEMISTAS, el podcast de ingeniería de sistemas en español.

¿Para qué?

Esta me la sé: para seguir aprendiendo en primer lugar, y luego para dar visibilidad a profesionales españoles (no, mentira, a profesionales que hablen español), para dar visibilidad a empresas y proyectos y, resumiendo, para intentar que los que nos movemos en el mundo de la Ingeniería de Sistemas y compartimos el mismo idioma, podamos sentir un poco de unión y orgullo.

Hay mucho y muy bueno de esta especialidad en este idioma ¡a los invitados del podcast me remito! ¿te puedo contar un secretillo? Con alguno de ellos me he puesto nervioso, me daba cierto respeto hacer preguntas a referentes mundiales en su ámbito... pero eso también suma ¿no?

Es una lección que aprendo de cada gran experto en Ingeniería de Sistemas con el que hablo, ya sean españoles o no: su humildad es mucho más grande que su conocimiento.

¿Hasta cuándo?

Hasta que vosotros (los invitados) queráis, vosotros hacéis el podcast, yo soy el que os insiste para grabar y aprende de vosotros. No tengo un calendario de proyecto, improviso bastante, y lo hago así a propósito. Esta actividad tiene que ser, sobre todo, divertida y permitirme aprender.

Así que esto acabará el día que no tenga preguntas que hacer (es decir, nunca), el día que no tenga contenido que compartir (tengo previsto algún episodio sin invitado, en el que te traduciré contenido interesante en inglés) o el día en que nadie quiera venir a charlar conmigo.

Ya ves, depende de varios factores, hasta entonces espero que los episodios te gusten.

Y si tienes algo que contar sobre Ingeniería de sistemas ¡escíbeme! Yo quiero escucharte y aprender.

<https://www.linkedin.com/in/luisandes/>



METODOLOGÍA MBSE ISE&PPOOA APLICADA A LA DEFINICIÓN DE ARQUITECTURA FUNCIONAL DE UN SISTEMA



Alfonso García Casado
José L. Fernández Sánchez
CONSULTORES
INDEPENDIENTES

Resumen

Una de las tareas clave que los ingenieros de sistemas deben realizar es entender y definir la conducta emergente del sistema, es decir, qué debe hacer éste cuando opere en su contexto para entregar el valor esperado a sus usuarios, clientes, y otros entes interesados del sistema.

Entender esta emergencia a través de la definición de la arquitectura funcional tiene numerosos beneficios que facilitan las diferentes fases de desarrollo del sistema. En este artículo demostraremos cómo aplicar la metodología MBSE ISE&PPOOA para crear una arquitectura funcional consistente con el propósito, contexto, y operación de un Sistema de Freno de Mano Eléctrico, desarrollado en Cameo Systems Modeler® con SysML, haciendo uso de sus capacidades de simulación.

Web site:

<http://www.omgwiki.org/MBSE/doku.php?id=mbse:ppooa>

Emails:

alfgarciascasado@gmail.com, josefernandez@telefonica.net

Copyright © 2022 por A. García and J.L. Fernández.

Todos los derechos reservados.

Introducción

Consideramos un sistema como la combinación de elementos relacionados entre sí que funcionan de manera conjunta para producir un resultado que satisface una necesidad [1]. Esa funcionalidad no se produce por las contribuciones de los elementos que componen el sistema por separado de forma aislada, sino que emerge de las interacciones entre esos elementos cuando el sistema trabaja en su contexto operacional [2]. Por lo que analizar y entender esa conducta emergente resulta clave para crear sistemas que cumplan su propósito y provean del valor esperado a sus usuarios.

El resultado de este análisis es la arquitectura funcional, que representa la descomposición de las funciones del sistema, las interacciones entre esas funciones y los flujos funcionales describiendo el comportamiento del sistema para cumplir con su objetivos. Así, podemos decir que la arquitectura funcional representa la definición del problema al que los ingenieros de sistemas nos enfrentamos: ¿Qué debe hacer el sistema para entregar su valor esperado? De este modo, consideramos la definición de la arquitectura funcional como una actividad clave dentro del modelado de arquitectura de sistemas [3].

Una de las principales ventajas de la arquitectura funcional es su independencia de la solución técnica del sistema, sobre todo a los niveles superiores de su jerarquía, dándonos flexibilidad al definir y acotar el espacio de soluciones, evitando restringirlo demasiado pronto en las fases tempranas de desarrollo, así como minimizar el impacto de cambios durante la fase de diseño, permitiendo mantener opciones de solución abiertas a medida que definimos los elementos físicos del sistema e implementamos soluciones tecnológicas.

En este artículo mostramos cómo aplicar la metodología MBSE ISE&PPOOA [4] para definir la arquitectura funcional de un Sistema de Freno de Aparcamiento Eléctrico (EPBS) usando la herramienta de modelado Cameo Systems Modeler® sin perfiles específicos personalizados, solamente con el perfil estándar SysML.

Básicamente un EPBS tiene la misma responsabilidad que el clásico freno de mano: retener el vehículo una vez aparcado. Dotarlo de electrónica provee al conductor de ventajas que facilitan su labor, transfiriendo algunas responsabilidades que antes residían en el conductor al EPBS, como determinar la fuerza de retención necesaria para mantener el vehículo inmovilizado una vez aparcado; y dotándole de nuevas, como la asistencia al conductor en salidas con pendiente, o incrementar la seguridad del vehículo. El diagrama de contexto del EPBS se muestra en la Figura 1.

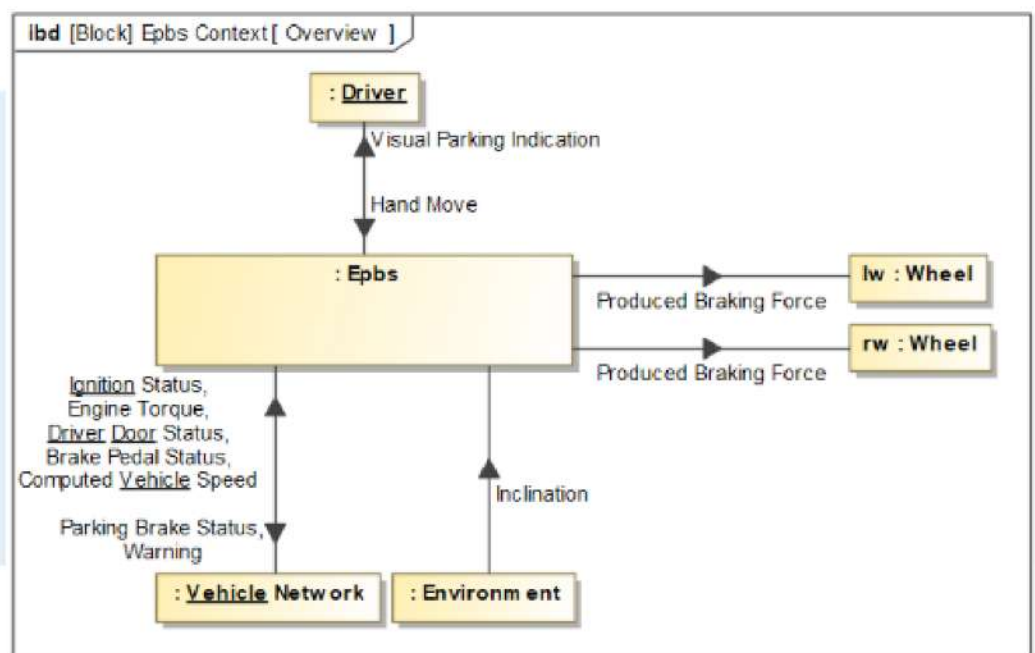


Figura 1. Diagrama de Contexto del EPBS.

El proceso

ISE&PPOOA es una metodología de Ingeniería de Sistemas requirements-driven que integra la aproximación MBSE con la Ingeniería del Software para el desarrollo de productos de alta complejidad. ISE&PPOOA aúna de forma consistente tres dimensiones: misión, sistema y software [4].

La dimensión misión aglutina el contexto del sistema, sus escenarios de operación, así como las necesidades operacionales y capacidades relacionadas con la misión.

La dimensión sistema incluye tanto los requisitos de sistema, como las arquitecturas funcional y física, identificando subsistemas, la funcionalidad asignada a los elementos físicos y las interfaces necesarias para cumplir con la misión.

La dimensión software se deriva de la dimensión sistema y define la arquitectura software para los subsistemas intensivos en software, incluyendo estructura y comportamiento.

El proceso metodológico para construir las dimensiones misión y sistema se muestran en la Figura 2 y se definen en detalle en el libro de la metodología [4].

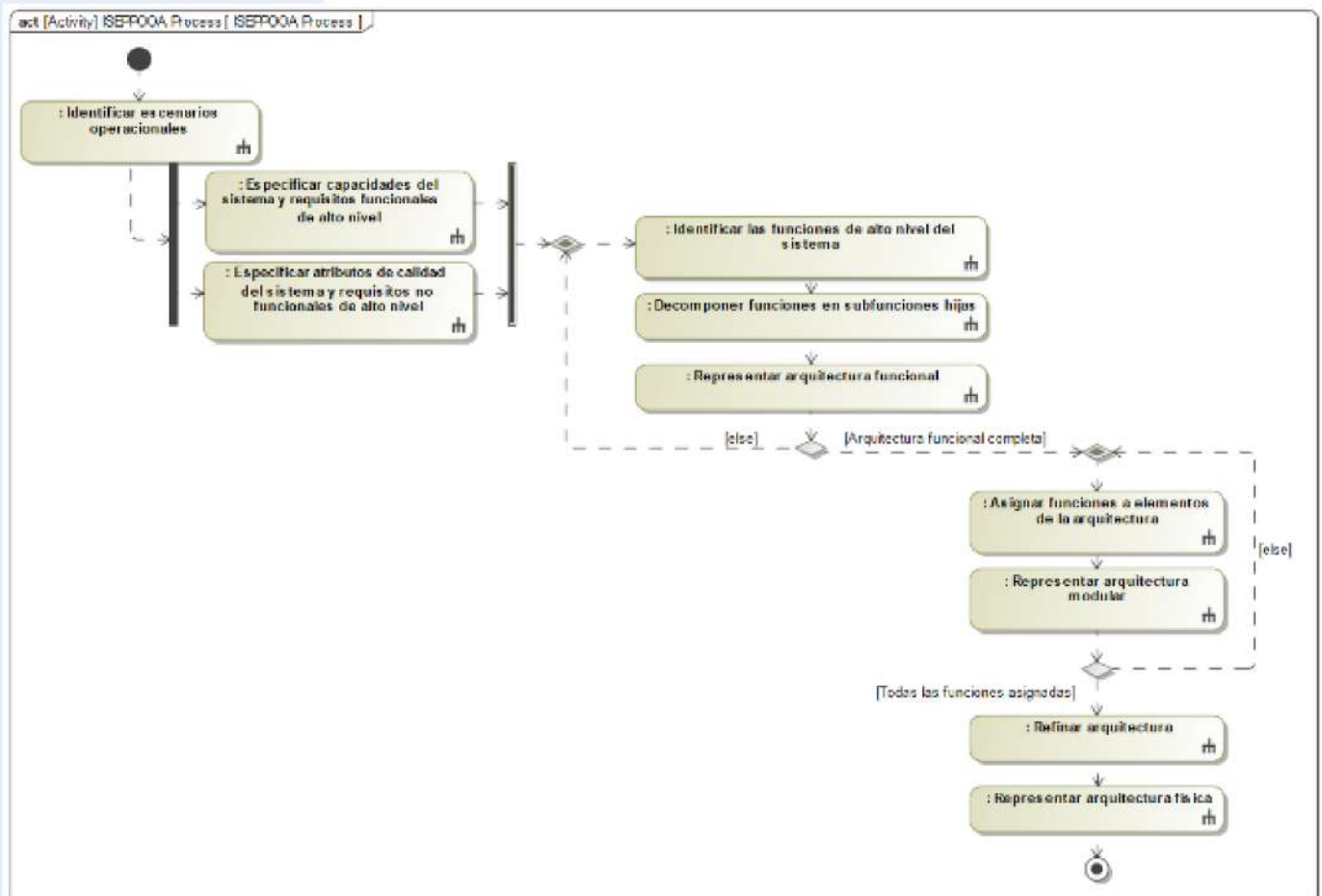


Figura 2. Proceso ISE&PPOOA (Dimensiones, Misión y Sistema)

Definición de arquitectura funcional

Una vez que hemos entendido, analizado y definido la dimensión misión (incluyendo escenarios operacionales, capacidades, atributos de calidad y requisitos de alto nivel), saltamos a la dimensión sistema para construir la arquitectura funcional del sistema, que lo haremos de manera iterativa como viene definida en la Figura 3.

Como resultado del proceso obtendremos:

1. La jerarquía funcional del sistema representada en SysML por diagramas de definición de bloques (bdd). Estas funciones satisfacen los requisitos de alto nivel del sistema.
2. Las interfaces funcionales, que identificamos y representamos usando matrices N^2 [5], mostrando intercambios de materia, energía o información.

Flujos funcionales representados en SysML por diagramas de actividad (act) [6], describiendo el comportamiento requerido del sistema a partir de las funciones e interfaces funcionales definidas en los dos pasos anteriores.

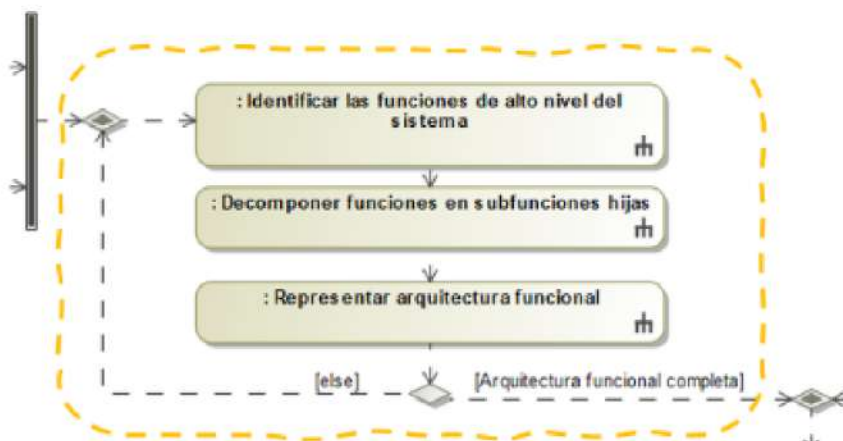


Figura 3. Definición de la Arquitectura Funcional.

1 Jerarquía Funcional

La **jerarquía funcional** es una representación de descomposición de las funciones de alto nivel del sistema en funciones hijas, hasta llegar a un nivel detalle que:

1. nos permita identificar elementos físicos que sean responsables de proveer las funciones asignadas;
2. todas las entradas al sistema se pueden transformar en salidas del sistema a su contexto de operación;
3. todos los escenarios operacionales están cubiertos por la funcionalidad definida.

Es esencial recalcar que una función la consideramos en ISE&PPOOA como una transformación de flujos de entrada en flujos de salida, siendo éstos materia, energía o información. Por ejemplo, la función de un motor eléctrico es convertir (transformación) energía eléctrica (flujo de entrada) en energía mecánica (flujo de salida) capaz de realizar trabajo; un ascensor transporta (transformación) personas (flujo de entrada y salida) verticalmente de una planta a otra del edificio.

A la hora de definir la jerarquía funcional, recomendamos usar tanto la aproximación “top-down” como la “bottom-up”.

Aproximación “top-down”: partimos de las capacidades de sistema [7] y sus requisitos funcionales de alto nivel para identificar las funciones alto nivel que formarán la parte superior de la jerarquía funcional.

#	Name	Text
1	54 EPBS Capabilities	
2	54.6 Comfortable Control	The system allows the <u>driver</u> to easily engage or release the parking brake, by just requesting the intended action through the push/pull of a button. The system will evaluate the engage request, calculate the needed braking force to retain the <u>vehicle</u> stationary and immobilize the <u>vehicle</u> safely. In case of a release command, the system will evaluate whether it is safe to disengage the parking brake, and proceed if there is no hazard.
3	54.2 Safe Automatic Release	The system allows the <u>driver</u> to drive away from parking position, or when starting on a slope, without direct manual interaction from the <u>driver</u> . The system will automatically detect when it is safe to release the <u>vehicle</u> to avoid rollback. Additionally, it will prevent releasing the <u>vehicle</u> if the <u>driver</u> 's seat belt is not buckled and engine is on (safe child lock).

Figura 4. Ejemplo de capacidades de sistema para el EPBS.

Aproximación *bottom-up*: partimos de los escenarios operacionales para identificar funciones de bajo nivel, que iremos agrupando para construir funciones de mayor nivel cumpliendo los principios de cohesión [8].

#	Name	Pre Condition	Basic Flow of Events	Post Condition
1	○ Immobilize <u>Vehicle</u> Automatically	<p><u>Vehicle</u> speed is less than 7 km/h.</p> <p>Ignition is On.</p> <p>EPBS in not retaining the <u>Vehicle</u>.</p>	<p>The <u>Driver</u> opens his door.</p> <p>EPBS detects the <u>door</u> is open and immobilizes the <u>Vehicle</u> to avoid the <u>Driver</u> gets injured when he leaves the <u>Vehicle</u> while it is moving in an uncontrolled manner.</p> <p>EPBS provides a warning to the <u>Driver</u> to indicate the hazardous situation.</p> <p>EPBS indicates the <u>Driver</u> that the Parking Brake is engaged.</p>	<p>EPBS is retaining the <u>Vehicle</u>.</p> <p><u>Vehicle</u> speed is 0 km/h.</p>

Figura 5. Ejemplo de Escenario Operacional en formato texto.

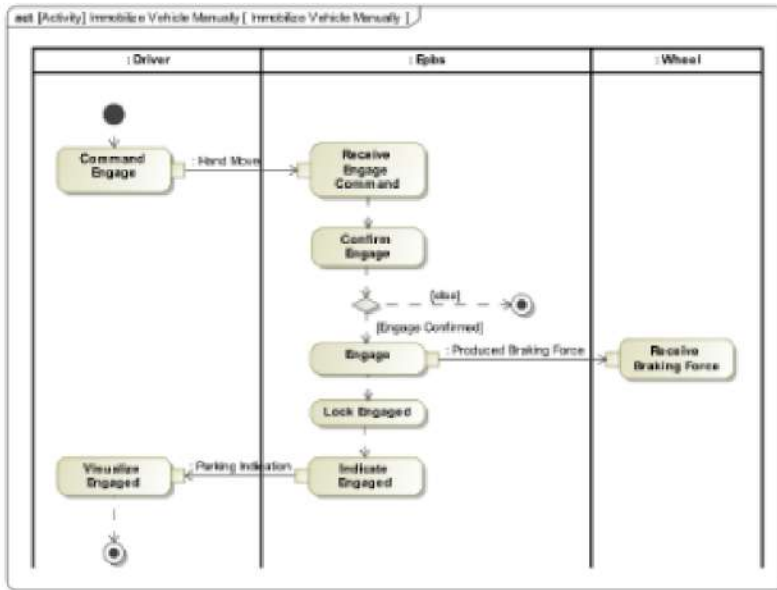


Figura 6. Ejemplo de Escenario Operacional en formato diagrama de actividad.

La identificación de las funciones de alto nivel y su descomposición no es una tarea trivial y, se necesita de un gran dominio de la abstracción y un conocimiento profundo del dominio del sistema.

Por otro lado, podemos usar taxonomías funcionales definidas en dominios de aplicación similares al de nuestro sistema en desarrollo, como la taxonomía de funciones de un avión comercial publicada por INCOSE [9], la taxonomía para procesos industriales del NIST [10]. También podemos usar plantillas y patrones funcionales, como la definida por Hatley y Pirbhai [11] para sistemas de control de tiempo real (Figura 7) o el clásico patrón de control de lazo cerrado, tal y como propone Buede [12].

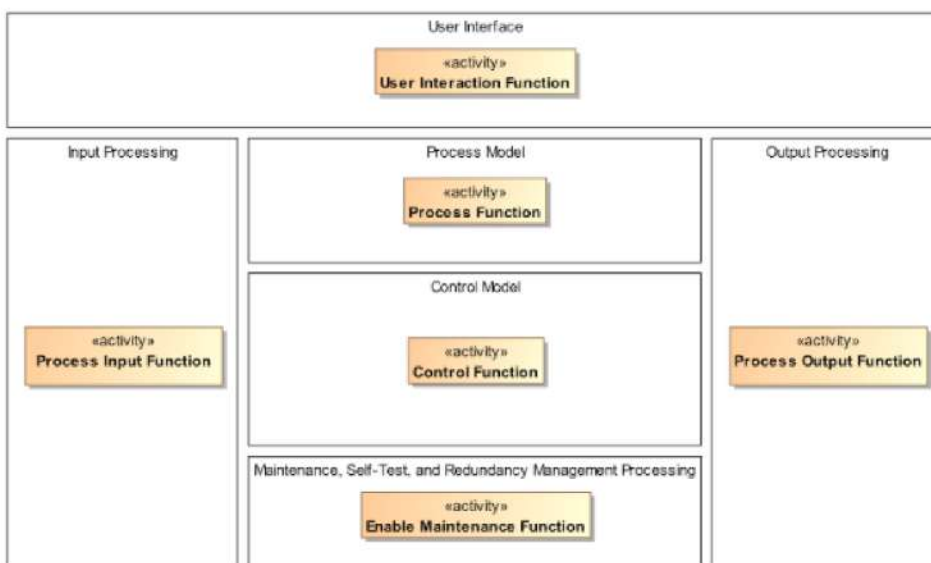


Figura 7. Plantilla de Hatley-Pirbhai para sistemas de tiempo real.

Un ejemplo de jerarquía funcional para el EPBS lo encontramos en la Figura 8.

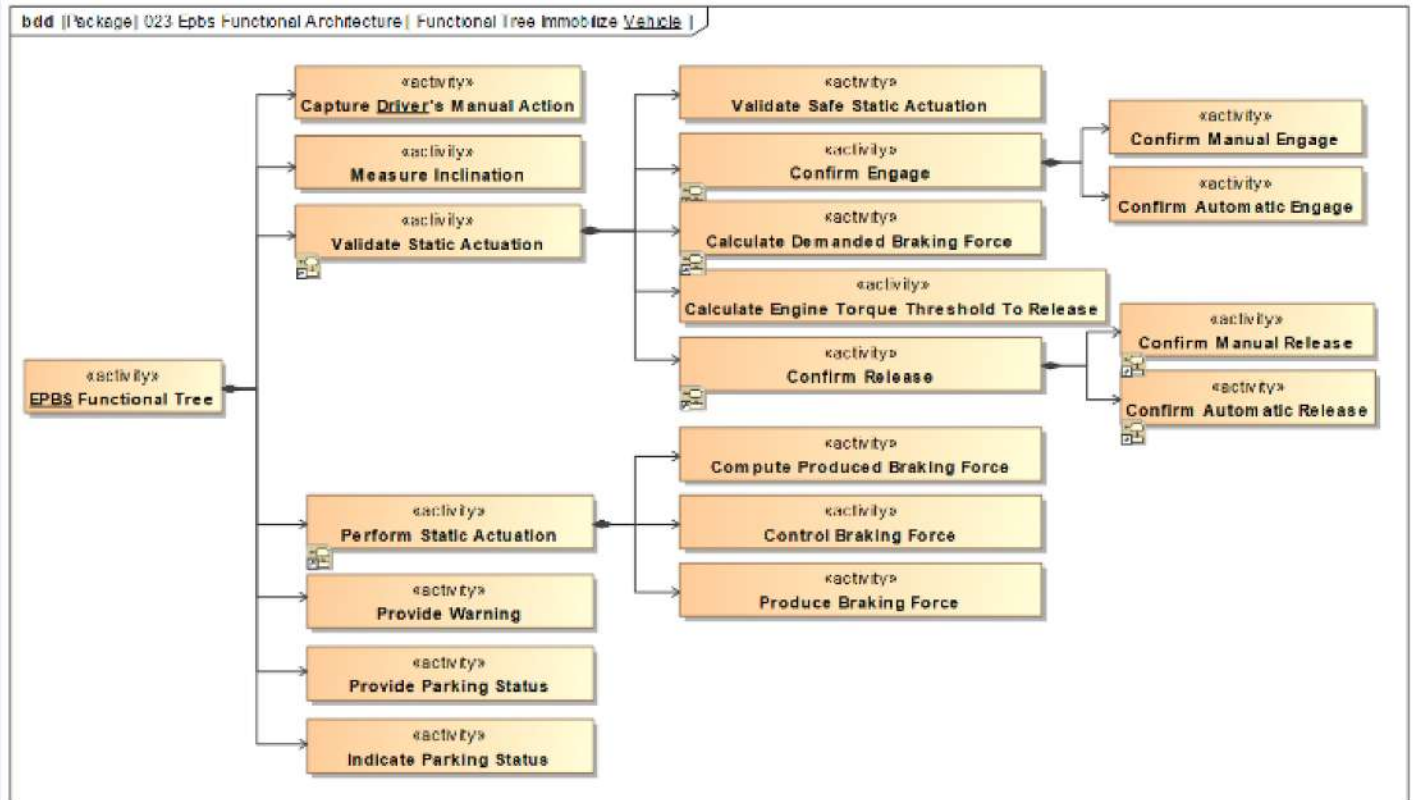


Figura 8. Ejemplo de jerarquía funcional para el EPBS.

Durante todo este proceso es esencial mantener la consistencia de flujos entrada/salida entre funciones padre e hijas (Figura 9) en todas las capas de la jerarquía (Figura 8).

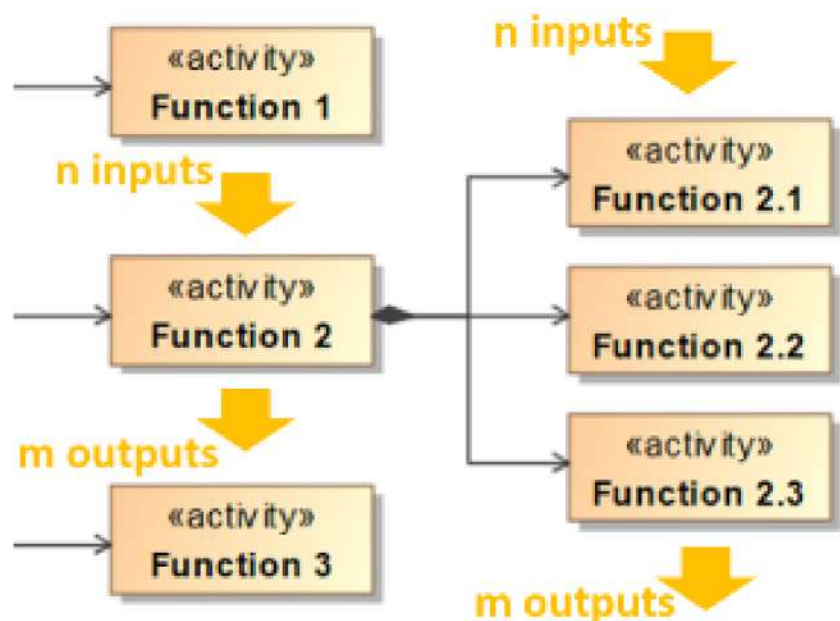


Figura 9. Mantener flujos entrada/salida entre capas.

2 Interfaces Funcionales

Para definir las interfaces funcionales, seleccionamos un grupo de funciones de la jerarquía funcional y creamos una matriz N^2 con esas funciones, definiendo los intercambios de materia, energía e información entre ellas. Una de las grandes ventajas de la matriz N^2 es que muestra de manera muy compacta y simple la complejidad funcional del sistema, permitiendo analizar la calidad de la jerarquía funcional elegida respecto a sus dependencias “out-in”, así como la modularidad de la arquitectura física a partir de la identificación de grupos funcionales altamente acoplados, que nos minimizará futuras interfaces físicas entre componentes [13].

Para nuestro ejemplo, la funcionalidad elegida es la relacionada con la inmovilización del vehículo (Figura 10). La matriz N^2 resultante (Figura 11) es triangular superior sin lazos de realimentación, mostrando una secuencia de transformación clara entre flujos de entrada y de salida externas, consistente con el contexto del sistema (Figura 1).

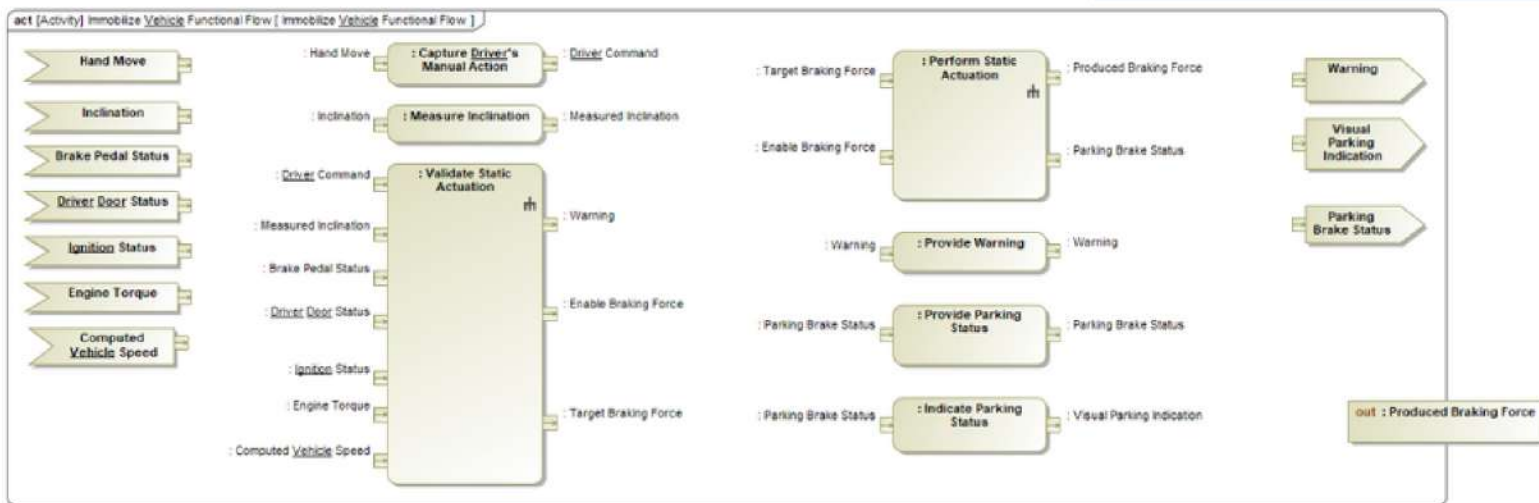


Figura 10. Funciones seleccionadas para el EPBS con entradas y salidas.

Para representar los flujos funcionales mostramos los flujos identificados en la matriz N^2 en el diagrama de actividad de la Figura 10, añadiendo los flujos de control necesarios para regular la ejecución funcional, de modo que el diagrama refleje el comportamiento esperado, usando simulación para validar el comportamiento del diagrama.

El flujo funcional resultante para el EPBS se muestra en la Figura 12.

Legend		Immobilize Vehicle Functional Flow												
↗ Object Flow														
↗ ObjectFlowThroughNode														
		:Capture Driver's Manual Action	:Measure Inclination	:Validate Static Actuation	:Perform Static Actuation	:Indicate Parking Status	:Provide Parking Status	:Provide Warning	:Produced Braking Force	Visual Parking Indication	Parking Brake Status	Warning		
Immobilize Vehicle Functional Flow														
Hand Move	1													
Inclination		1												
Brake Pedal Status			1											
Driver Door Status				1										
Ignition Status					1									
Engine Torque						1								
Computed Vehicle Speed							1							
:Capture Driver's Manual Action								1						
:Measure Inclination									1					
:Validate Static Actuation										2				
:Perform Static Actuation											1			
:Indicate Parking Status												1		
:Provide Parking Status													1	
:Provide Warning														1

Figura 11. Matriz N² de la principal funcionalidad del EPBS.

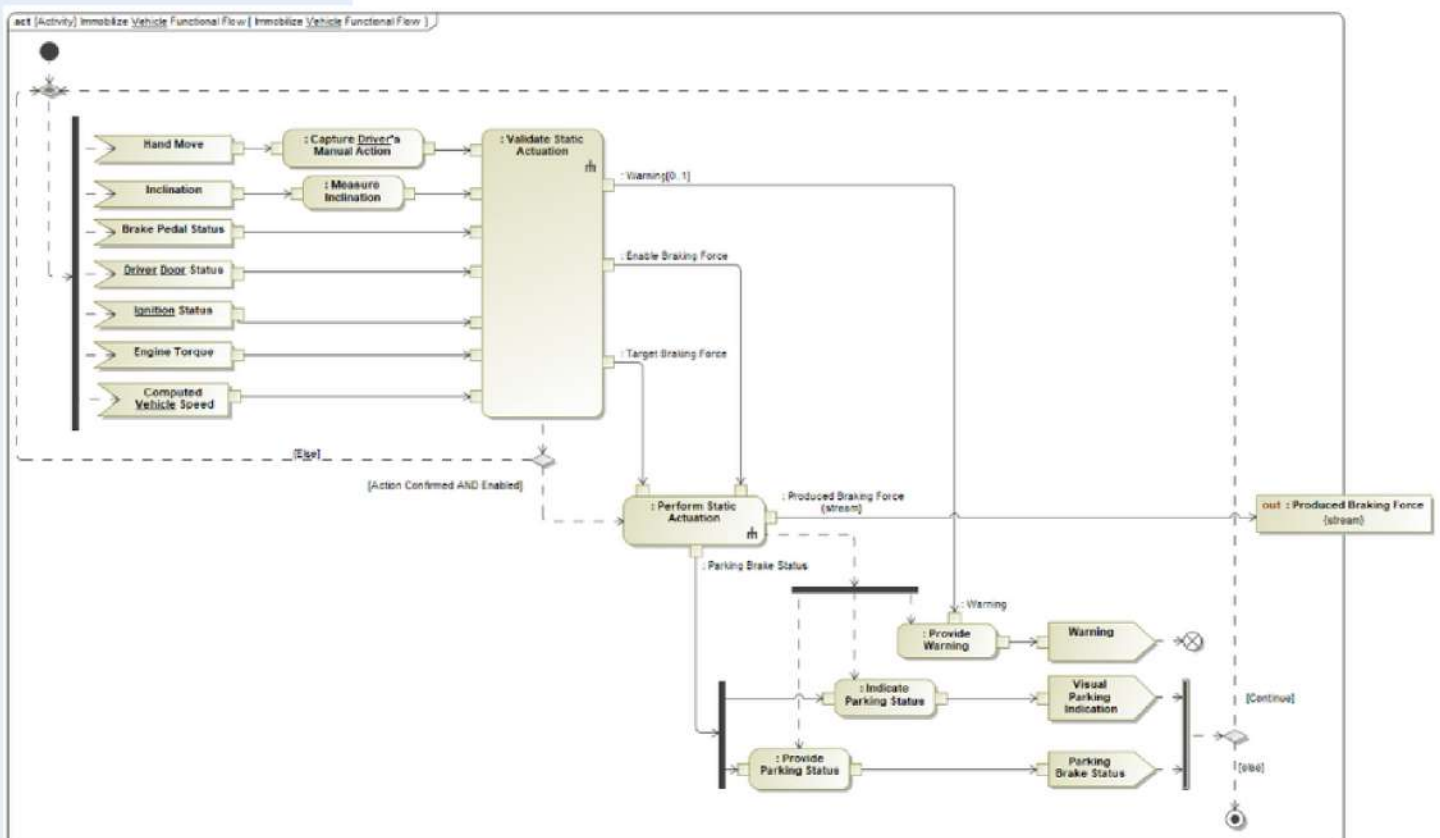


Figura 12. Flujo funcional principal del EPBS para la inmovilización del vehículo.

Conclusión

Hemos presentado cómo aplicar la metodología MBSE ISE&PPOOA usando la herramienta Cameo Systems Modeler® para definir la arquitectura funcional de un sistema, con un resultado elegante y efectivo, validado por simulación, y consistente con la misión del sistema, su contexto, capacidades y escenarios operacionales para entregar valor a sus stakeholders.

El uso de la jerarquía funcional (Figura 8) y el análisis de las interfaces funcionales con la matriz N^2 (Figura 11) nos permiten generar flujos funcionales en diagramas de actividad (Figura 12) para describir el comportamiento del sistema de forma clara, sencilla y consistente.

Con una buena definición del problema (jerarquía funcional, matriz N^2 y flujos funcionales), podremos definir una solución física para el sistema que sea modular y flexible, con interfaces mínimas entre componentes, y que satisfice las expectativas de sus stakeholders al ser consistente con la misión del sistema.

Referencias

1. NASA *Systems Engineering Handbook*
2. E. Crawley, B. Cameron, and D. Silva. *System Architecture: Strategy and Product Development for Complex Systems*. Pearson Higher Education 2016.
3. R.S. Carson and B.J. Sheeley. *Functional Architecture as the Core of Model-Based Systems Engineering*. Proc. INCOSE International Symposium, Vol. 23, 2013, pages 29-45.
4. J.L. Fernandez and C. Hernandez. *Practical Model-Based Systems Engineering*. Artech House 2019.
5. D.K. Hitchins. *Advanced Systems Thinking, Engineering and Management*. Artech House, 2003.
6. C. Bock. *SysML and UML 2 Support for Activity Modeling*. Systems Engineering, Vol. 9, No. 2, 2006.
7. DoD Deputy Chief Information Officer. *DoDAF Architecture Framework Version 2.02*. Department of Defense, 2010.
8. R. Stevens, et al. *Systems Engineering. Coping with Complexity*. Prentice Hall Europe, 1998.
9. INCOSE. *Framework for the Application of Systems Engineering in the Commercial Aircraft Domain*. International Council on Systems Engineering (INCOSE), 2000.
10. J.M. Hirtz, et al. *Evolving Functional Basis for Engineering Design*. Proc. ASME Design Engineering Technical Conference, Pittsburgh, PA, September 9–12, 2001.
11. D.J. Hatley, and I. A. Pirbhay, *Strategies for Real-Time System Specification*, New York: Dorset House, 1988.
12. Dennis M. Buede, *The Engineering Design of Systems: Models and Methods*. John Wiley & Sons, 2009.
13. D.A. Gebala, and S.D. Eppinger. *Methods for Analyzing Design Procedures*. De-Vol. 31, Design Theory and Methodology ASME, 1991)

Autores



Alfonso García es Ingeniero Industrial por la Universidad Politécnica de Madrid y trabaja como Consultor Independiente en Ingeniería de Sistemas; certificado CSEP (Certified Systems Engineering Professional) desde 2014. Tiene más de 15 años de experiencia como ingeniero de sistemas, arquitecto y consultor en diferentes industrias y proyectos, incluyendo sistemas de gestión de tráfico aéreo, aviónica, automoción, y sistemas de elevación, usando tanto la aproximación tradicional a la Ingeniería de Sistemas como en su vertiente MBSE. Es miembro de INCOSE desde 2008.



José L. Fernández Sánchez es Doctor en Informática e Ingeniero Aeronáutico, ambos por la Universidad Politécnica de Madrid. Tiene más de 30 años de experiencia en la industria. Estando involucrado en proyectos relacionados con el desarrollo de software y mantenimiento de grandes sistemas, específicamente sistemas en tiempo real para control de tráfico aéreo, SCADAs en generación de energía, aviónica y aplicaciones de telefonía móvil. Fue profesor titular en la E.T.S. Ingenieros Industriales, Universidad Politécnica de Madrid (UPM). Es el autor de ISE&PPOOA, una metodología MBSE aplicable a sistemas intensivos en software. Es miembro sénior del IEEE y miembro de INCOSE, participando en el cuerpo de conocimiento de ingeniería de software, el cuerpo de conocimiento de ingeniería de sistemas y los grupos de trabajo de ingeniería de requisitos de estas asociaciones. Es miembro del PMI participando como revisor de PMBoK 6th Edition, 2017, y de Requirements Management, Practice Guide, 2016.

GRUPOS DE TRABAJO TÉCNICOS DE LA AEIS



Demetrio Zorita
Senior Systems Engineer - SENER
Aeroespacial
Director Técnico - AEIS INCOSE

Los Working Groups (WG) son una de las formas más directas que tiene INCOSE para lograr impacto en la forma en que se hace Ingeniería de Sistemas (IS). Existen desde hace mucho tiempo en los homólogos extranjeros de la AEIS, y su aportación es muy valiosa y su valoración muy positiva.

Los WG de la AEIS tratarán de:

- Investigar cómo se hace IS en España en sus respectivas áreas
- Promover buenas prácticas y mejorar la forma de hacer IS en estas áreas
- Servir de punto de encuentro de profesionales donde enriquecerse, compartir experiencias y divertirse con ello

Cada país tiene una serie de WG distintos, respondiendo a las inquietudes de sus miembros. En la AEIS hemos empezado por un WG genérico de Voluntariado, cuyas iniciativas internas han derivado en la creación de tres primeros WG técnicos en verano de 2022:

WG de
MBSE

WG de
Requisitos

WG de Complex
Systems, Ontologies,
and System Thinking

Estos WG, y los que se puedan crear en el futuro, están formados por compañeros con conocimientos o intereses en estos dominios. Tiene un carácter completamente voluntario y no retribuido, siendo la principal motivación el compromiso con la difusión y mejora de la IS nacional, y las ganas de ver cómo se hace su trabajo en otros sitios y tener un rico foro de debate al respecto.

Los WG tienen plena libertad para proponer las acciones que más interesen y motiven a sus participantes, y para organizarse logísticamente. La Dirección Técnica de la AEIS los apoya, confirma que haya consistencia entre ellos y garantiza su alineamiento con el mandato general de la AEIS.

Los WG están abiertos a cualquier profesional español o trabajando en España, con ganas de contribuir y relacionarse, sea miembro de AEIS o no. En breve los abriremos también a compañeros de INCOSE Hispanoamérica.

Te invitamos a participar y aportar tus ganas y conocimientos para conseguir entre todos una mejor IS en España e Hispanoamérica. Te proporcionará:

- La posibilidad de tener impacto con tu paso por ellos
- La oportunidad de saber de primera mano el estado del arte, el quién-es-quié, el quién-hace-qué, y el cómo-lo-hace, en tu área de trabajo
- La ocasión de conocer profesionales con gran bagaje en tu dominio, debatir y disfrutar con ello
- Puntos PDU para la renovación del certificado CSEP de INCOSE, mediante la participación en las reuniones y otras actividades del WG

Para apuntarte a alguno, no tienes más que entrar en la web de la AEIS e inscribirte, ¡se tardan 30 segundos! (<https://www.aeis-incose.org/recursos-para-miembros/working-groups/>). Una vez dado de alta, se te invitará a las reuniones.

El estado de los WG técnicos actuales es el siguiente:

WG de MBSE

En este WG se busca debatir y promover la adopción del enfoque MBSE, incluyendo metodologías, lenguajes y herramientas. Estudiar su uso actual en España y fomentar su adopción como método de mejora de las prácticas actuales de IS. Explicitar sus bondades, detectar sus dificultades de penetración y retos. Analizar sus especificidades para distintos sistemas y sectores. Estudiar su estandarización. Promover la formación en MBSE, tanto a nivel profesional como universitario. Recopilar buenas prácticas, asesorar a miembros de AEIS sobre MBSE.

Su primer proyecto es una encuesta de penetración de MBSE en la ingeniería española y la prospección al respecto. Además están estudiando editar una guía de iniciación a MBSE (guía "Don't panic"), y la organización de un congreso temático sobre MBSE en España el año que viene.

Se trata de un WG que ha arrancado con mucho impulso. Lo forman unas 30 personas con 4 co-líderes de la ESA, Mathworks, Indra y Akkodis. Se reúne por videoconferencia mensualmente durante 1 hora, desde este Septiembre.



WG de REQUISITOS

En este WG se busca agrupar y consolidar los Requisitos en España, creando un espacio de trabajo a nivel nacional de interesados que trabajan con IS y Requisitos, que desean continuar avanzando en este tema y mejorar las prácticas en España.

El WG ha impartido tres Webinar en Septiembre, Octubre y Noviembre sobre tres recientes guías INCOSE sobre requisitos: INCOSE Guide to Needs and Requirements, INCOSE Guide for Writing Requirements, e INCOSE Guide to Verification and Validation, que están disponibles en el Canal youtube "[AEIS INCOSE Chapter](#)". Se encuentran ahora definiendo su próximo proyecto, que posiblemente sea una encuesta y prospección sobre la Ingeniería de Requisitos en España, además de nuevos Webinar.

Se trata de un WG en proceso de adhesión de participantes. Lo forman actualmente unas 5 personas y además 3 co-líderes de Reuse e Indra. Están definiendo la logística de sus iteraciones.

WG de Complex Systems, Ontologies & System Thinking

En este WG se busca agrupar y consolidar el Complex Systems, Ontologies, and System Thinking en España, creando un espacio de trabajo a nivel nacional de interesados que trabajan con Ingeniería de Sistemas, Complex Systems, Ontologies y System Thinking, que deseen continuar avanzando en estos temas y mejorar sus prácticas en España.

Se trata de un WG incipiente que aún está definiéndose y decidiendo su hoja de ruta, liderado por 2 co-líderes de la UC3M y Balandra SW.

Otros WG futuros

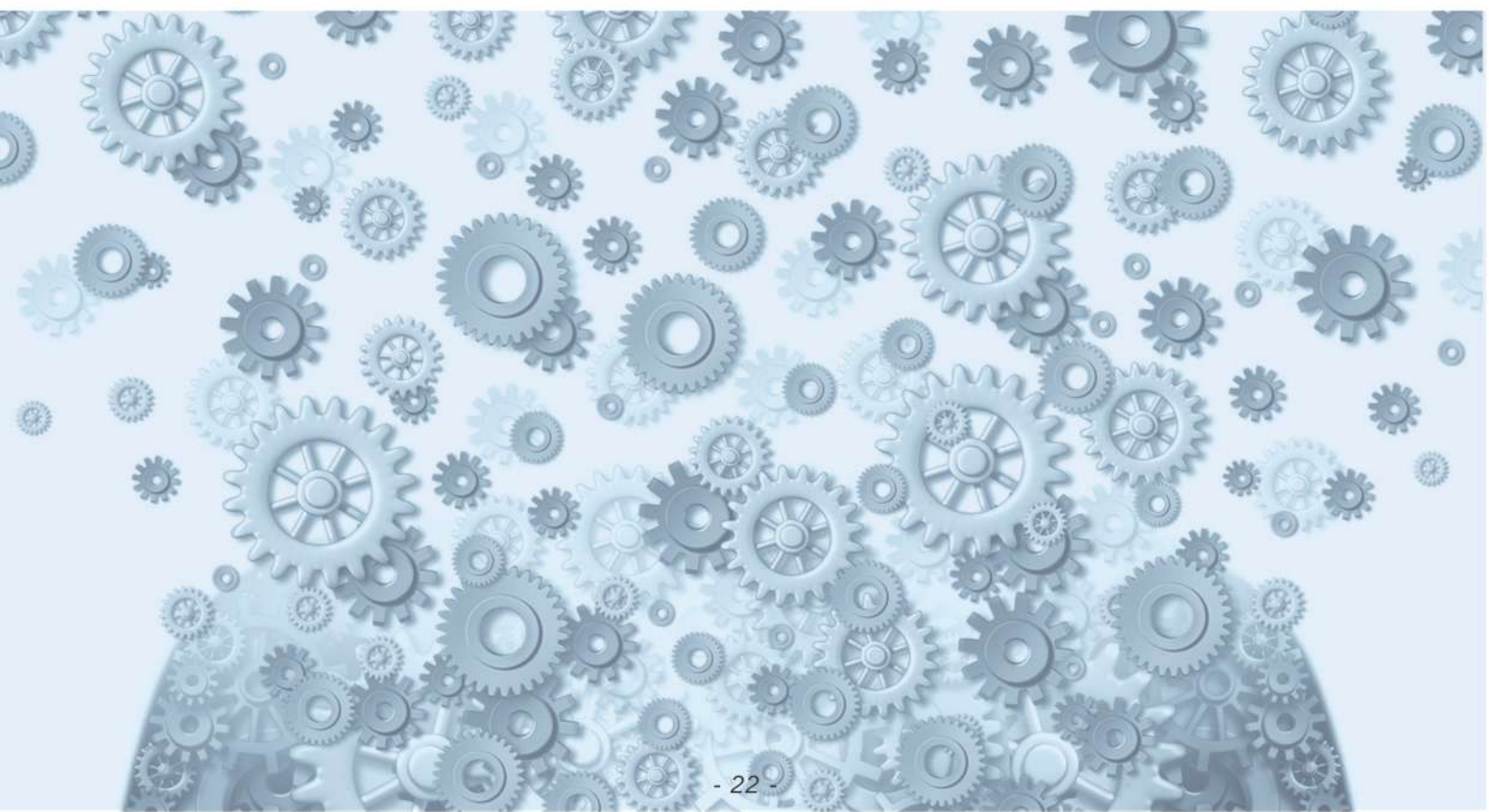
AEIS está abierta a la creación de nuevos WG en base a la motivación e intereses de sus miembros.

Ejemplos podrían ser: Formación y competencias / Métodos y procesos de IS / Herramientas de IS / Innovación en IS / Systems Engineering Management / Gestión del conocimiento / Integración, Verificación y Validación / Seguridad, Mantenimiento y Operaciones / IS Agile o Lean / Arquitecturas / Gestión del riesgo / Factores Humanos / Gestión de configuración / Sistemas de Sistemas / WG sectoriales.

Si tienes ganas y ánimos para ello, no dudes en decírnos.

Estos WG podrían llevar a cabo acciones como:

<p>Prospección para ver cómo se hace IS en su rama en España y dificultades de penetración</p> 	<p>Promoción de buenas prácticas: Webinar, etc a definir por el WG</p> 	<p>Asesoramiento a AEIS y sus miembros en su materia</p> 	<p>Investigación y avance en el estado del arte</p> 	<p>Análisis de especificidades sectoriales</p> 
<p>Estudio de estandarización</p> 	<p>Catalogación de formaciones y herramientas disponibles en su área</p> 	<p>Punto de encuentro, enriquecimiento y compartición de experiencias de los miembros del WG</p> 	<p>Organización de un workshop temático periódico</p> 	<p>Otros proyectos autodefinidos</p> 



CERTIFICACIÓN SEP

PROGRAMA DE CERTIFICACIÓN PROFESIONAL DE INGENIERÍA DE SISTEMAS

Cristina Saiz Valverde

*Systems Engineering Consultant for
Defense Acquisition Programs
ISDEFE*

¿El Programa de Certificación Profesional de Ingeniería de Sistemas INCOSE ofrece una evaluación independiente de los profesionales en Ingeniería de Sistemas, impulsando:

- Una comunidad de Ingeniería de Sistemas:
- El reconocimiento de profesionales de la Ingeniería de Sistemas:
- Un respaldo del conocimiento de los profesionales, reconocido por las organizaciones e instituciones:

El objetivo del Programa de Certificación INCOSE es proporcionar un método formal para reconocer el conocimiento, la experiencia, la formación y el liderazgo en Ingeniería de Sistemas.

El Programa de Certificación reconoce tres niveles de conocimiento y experiencia en Ingeniería de Sistemas.



ASEP (Associate Systems Engineering Professional)

Nivel de certificación dirigido a personas al inicio de su carrera como ingenieros de sistemas, recién graduados universitarios y con una experiencia laboral entre 0 y 5 años



CSEP (Certified Systems Engineering Professional)

Nivel de certificación dirigido a ingenieros de sistemas en ejercicio, con más de 5 años de experiencia profesional en Ingeniería de Sistemas



ESEP (Expert Systems Engineering Professional)

Nivel de certificación dirigido a líderes senior en Ingeniería de Sistemas, con experiencia profesional mayor de 20 años y logros reconocidos en Ingeniería de Sistemas.

La obtención de cualquiera de los niveles ASEP/ CSEP/ ESEP requiere ser miembro de AEIS-INCOSE y mantener anualmente la membresía.

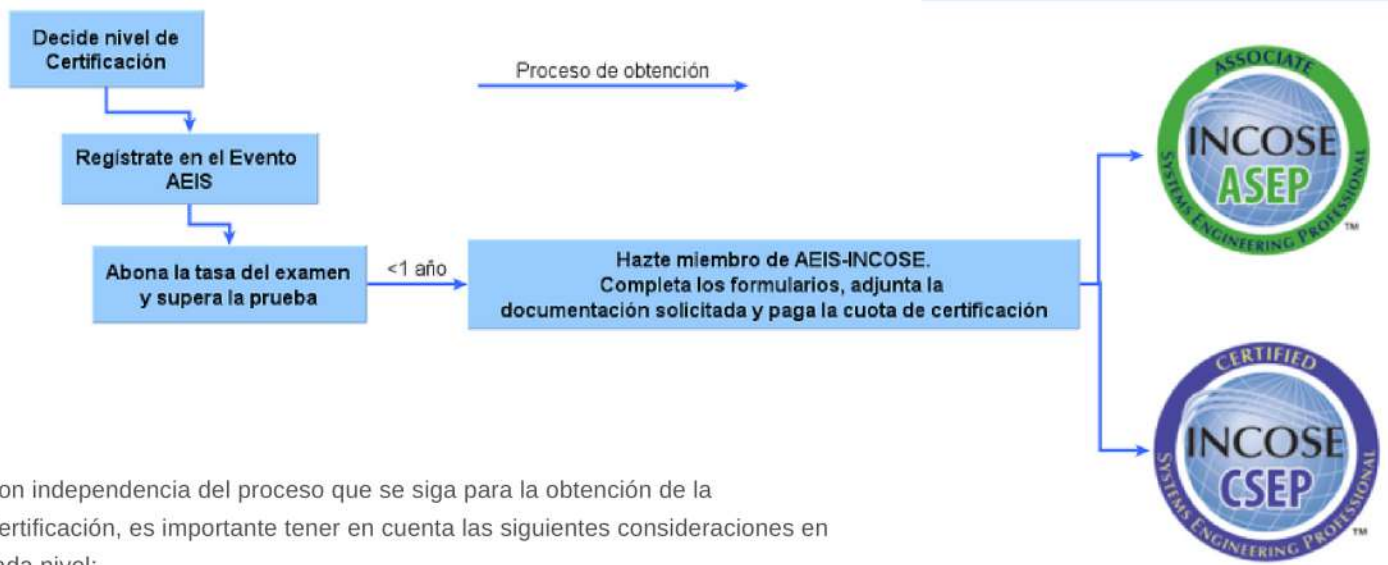
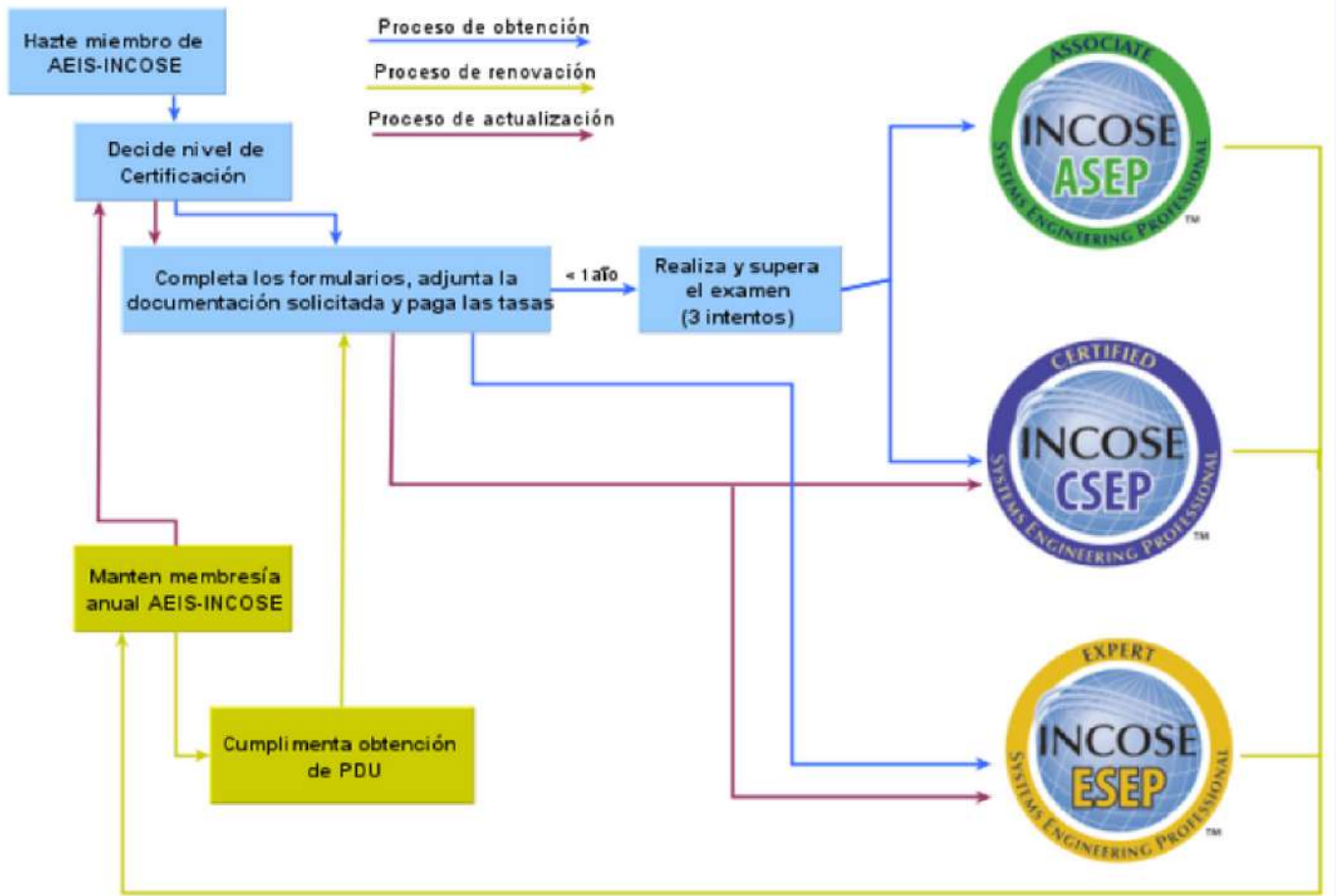
Actualmente, se identifican 2 vías para obtener la Certificación:

Proceso principal, facilitado directamente por INCOSE.

Proceso facilitado a través de Eventos patrocinados por AEIS.

La certificación de los niveles ASEP y CSEP, canalizada a través de los Eventos de AEIS, implica que todos los procesos de membresía, envío de documentación, justificaciones y pago de la cuota de certificación, pueden iniciarse tras haber superado el examen. Se dispone de 1 año, desde la fecha en la que se confirma el resultado, para formalizar los trámites.

LA OBTENCIÓN DE CUALQUIERA DE LOS NIVELES ASEP/ CSEP/ ESEP REQUIERE SER MIEMBRO DE AEIS-INCOSE Y MANTENER ANUALMENTE LA MEMBRESÍA.



Con independencia del proceso que se siga para la obtención de la Certificación, es importante tener en cuenta las siguientes consideraciones en cada nivel:

ASEP

- Se debe demostrar el conocimiento sobre Ingeniería de Sistemas a través de un examen que se basa en el Manual INCOSE SE (<https://www.incose.org/systems-engineering-certification/about-the-exam>).
- La certificación ASEP se renueva cada 5 años, con una duración máxima de 15 años, justificando el desarrollo profesional continuo en Ingeniería de Sistemas mediante la obtención de PDU (<https://www.incose.org/systems-engineering-certification/the-certification-process/how-do-i-renew#Activities>).

CON INDEPENDENCIA DEL PROCESO QUE SE SIGA PARA LA OBTENCIÓN DE LA CERTIFICACIÓN, ES IMPORTANTE TENER EN CUENTA ALGUNAS CONSIDERACIONES EN CADA NIVEL

CSEP

- Se debe demostrar el conocimiento sobre Ingeniería de Sistemas a través de un examen que se basa en el Manual INCOSE SE (<https://www.incose.org/systems-engineering-certification/about-the-exam>).
- Se debe justificar la experiencia profesional en Ingeniería de Sistemas (<https://www.incose.org/systems-engineering-certification/certification-forms>).
- La certificación CSEP se renueva cada 3 años, justificando el desarrollo profesional continuo en Ingeniería de Sistemas mediante la obtención de PDU (<https://www.incose.org/systems-engineering-certification/the-certification-process/how-do-i-renew#Activities>).

Aquellos que hayan logrado el reconocimiento de ASEP pueden hacer la transición a CSEP completando los requisitos de experiencia de CSEP y presentando la solicitud de CSEP junto con la cuota requerida para la transición a CSEP. No se requiere ningún examen adicional para este reconocimiento.

ESEP

Se debe demostrar el conocimiento, el liderazgo profesional, los logros y la experiencia en Ingeniería de Sistemas. Para ello:

- Se debe superar una entrevista telefónica por parte del panel de evaluación de ESEP.
- Se debe justificar la experiencia profesional en Ingeniería de Sistemas y la experiencia como líder (<https://www.incose.org/systems-engineering-certification/certification-forms>).
- La certificación ESEP no exige renovación. Sólo requiere mantener anualmente la membresía AEIS-INCOSE.

Por último, se refleja la información relativa a los costes requeridos en los procesos de obtención y mantenimiento de las Certificaciones.

	Tasa examen	Cuota certificación	Cuota renovación	Membresía anual
ASEP	80 USD examen por ordenador*. 30 USD examen en papel, a través de Eventos.	180 USD	100 USD (cada 5 años)	160** €
Transición ASEP a CSEP	-	200 USD	-	
CSEP	80 USD examen por ordenador*. 30 USD examen en papel, a través de Eventos.	350 USD	100 USD (cada 3 años)	
ESEP	-	630 USD	-	

* Actualmente no se realizan los exámenes a través de Prometric.

** Precio estándar anual. Para otras opciones de membresía, consultar: <https://www.aeis-incose.org/opciones-de-membresia/>

CÓMO HACERSE MIEMBRO DE AEIS

Puede darse de alta directamente en el apartado información de nuestra web* , en cualquiera de las siguientes opciones:

- **Membresía individual estándar:** 160€/año (Disponible para cualquier profesional involucrado en la ingeniería de sistemas con interés en mejorar sus conocimientos técnicos y tener acceso a una red de profesionales del sector).
- **Membresía individual estándar tres años:** 445€/ 3 años (Disponible para cualquier profesional involucrado en la ingeniería de sistemas con interés en mejorar sus conocimientos técnicos y tener acceso a una red de profesionales del sector).
- **Membresía individual estudiantes:** 50€/año (Disponible para todos aquéllos estudiantes de un área técnica, teniendo que poder demostrarlo como su ocupación principal. No se dispone de derecho a voto en las elecciones de INCOSE y AEIS).

<https://www.aeis-incose.org/opciones-de-membresia/>



SAVE THE DATES



[HTTPS://WWW.INCOSE.ORG/IW2023](https://www.incose.org/iw2023)

MANTENTE ACTUALIZADO:

<https://www.aeis-incose.org/eventos/>

<https://www.incose.org/>



[HTTPS://WWW.INCOSE.ORG/EMEAWSEC2023](https://www.incose.org/emeawsec2023)



ADVISORY BOARD (CAB)



uc3m

indra



Capgemini engineering



PATROCINADOR



ACUERDOS DE COLABORACIÓN



AEIS - INCOSE ESPAÑA

C/ Don Pedro, 10 28005 Madrid
(Real Academia de Ingeniería)
contact@aeis-incose.org
<https://www.aeis-incose.org/>

