

Marc Iglesias Llaberia

**Especificació i disseny de funció ADAS per integració en
vehicle prototip**

Treball Fi de Grau

dirigit per:

Sr. Carlos Olalla i Sr. Toni Escamilla

Grau d' Enginyeria en Electrònica Industrial i Automàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2023

Agraïments

M'agradaria agrair a tots aquells que m'han acompanyat en tot aquest procés, no tan sols al llarg d'aquest projecte sinó també al meu pas per la universitat. Sobretot al Xavier Boabén, company de IDIADA que sense cap motivació m'ha guiat i aconsellat de forma excel·lent.

Resum

La majoria dels accidents a les carreteres estan causats per errors dels conductors. Els vehicles autònoms busquen eliminar aquest factor humà i convertir les carreteres en un lloc més segur, tant pels conductors com pels vianants, així com també fer de la conducció una activitat molt més còmode.

Les *Advanced Driver Assistance Systems* o més comunament nombrades ADAS, ajuden a complir les tasques del conductor a la carretera de manera que redueixen el volum de treball cap al conductor i permeten una conducció més segura i amb més confort.

Aquest treball de fi de grau tracta el disseny i la integració d'una funció d'assistència a la conducció. El treball s'ha realitzat al departament d'Electrònica de l'empresa Applus+ IDIADA a Santa Oliva (Baix Penedès, Catalunya). L'objectiu és complir amb els requeriments del projecte SUNRISE per la implementació de la funció ADAS, *Highway Pilot*, per a que pugui ser integrat en un vehicle prototip. Per assolir-ho definirem totes les casuístiques, els requeriments a nivell de sistema i arquitectura.

Abstract

Most road accidents are caused by driver errors. Autonomous vehicles seek to eliminate this human factor and turn the roads into a safer location, both for drivers and for pedestrians, turning driving into a much more comfortable activity.

The "Advanced Driver Assistance Systems" or more commonly named ADAS, help to ease the severe obstacles in the road in a way that reduces the volume of workload for the driver and allows a safer and more comfortable driving.

This final bachelor's thesis deals with the design and the integration of a driving assistance function. The work has been developed in the electronics department of Applus+ IDIADA, a company based in Santa Oliva (Baix Penedès, Catalonia). The aim is to meet the requirements of the SUNRISE project for the implementation of the ADAS "Highway Pilot" function so that it can be integrated into a prototype vehicle. To achieve this, we define all the cases, the requirements at a high system and architecture level.

Resumen

La mayoría de los accidentes en las carreteras están causados por errores de los conductores. Los vehículos autónomos buscan eliminar este factor humano y convertir las carreteras en un lugar más seguro, tanto para los conductores como para los peatones, así como hacer de la conducción una actividad mucho más amena.

Las *Advanced Driver Assistance Systems* o más comúnmente llamadas *ADAS*, ayudan a cumplir las tareas del conductor en la carretera de forma que se reduce el volumen de trabajo hacia el conductor y se permite una conducción más segura y de mayor confort.

Este trabajo de fin de grado trata el diseño y la integración de una función de asistencia a la conducción. El trabajo se ha desarrollado en el departamento de Electrónica de la empresa Applus+ IDIADA en Santa Oliva (Baix Penedès, Cataluña). El objetivo es cumplir con los requisitos del proyecto SUNRISE para la implementación de la función *ADAS, Highway Pilot*, para que pueda ser integrado en un vehículo prototipo. Para ello definiremos todas las casuísticas, los requisitos a nivel de sistema y arquitectura.

Índex

1. Introducció.....	14
1.1 Motivació.....	14
1.2 Objectius.....	14
1.2.1 Abast del treball	15
1.3 Estructura.....	15
2. Estat de l'art.....	15
2.1 Vehicle autònom	16
2.1.1 Classificació i Nomenclatura dels Sistemes	16
2.2 ADAS	21
2.2.1 Infraestructura reguladora.....	22
3. Funció Highway Pilot – Procés Tècnic	25
3.1 Descripció general.....	25
3.1.1 Resum del producte	25
3.1.2 Solució real	26
3.2 Metodologia.....	26
3.2.1 Enginyeria de sistemes.....	26
3.2.2 Model-Based Systems Engineering	28
3.2.3 Metodologies.....	31
3.2.4 System Modeling Language	39
3.2.5 Polarion	47
3.2.6 Enterprise Architect.....	47
3.2.7 Meta-modelat.....	48
3.2.8 Automotive SPICE.....	50
3.3 Etapa Concepte.....	60
3.3.1 Requeriments per ADAS en el context de regulacions legals	60
3.3.2 Requeriments i necessitats dels <i>stakeholders</i>	63
3.4 Etapa de Desenvolupament.....	65

3.4.1	Definició dels requeriments de sistema	65
3.4.2	Anàlisi del disseny	67
3.4.3	Construcció de l'Arquitectura.....	75
3.4.4	Traçabilitat.....	114
4.	Conclusions i extensions.....	115
4.1	Conclusions	115
4.2	Limitacions	116
4.3	Línies futures	116
5.	Ontologia	118
6.	Referències	121

Abreviaciones

<i>ACC</i>	<i>Adaptative Cruise Control</i>
<i>ADAS</i>	<i>Advanced Driving Assistance Systems</i>
<i>AEB</i>	<i>Automatic Emergency Break</i>
<i>ASIL</i>	<i>Automotive Safety Integrity Levels</i>
<i>AUTOSAR</i>	<i>AUTOmotive Open Source ARchitecture</i>
<i>AV</i>	<i>Autonomous Vehicle</i>
<i>BSM</i>	<i>Blind-Spot Monitoring</i>
<i>CC</i>	<i>Cruise Control</i>
<i>CTG</i>	<i>Continuous Time Gap</i>
<i>DT</i>	<i>Driving Task</i>
<i>DDT</i>	<i>Dynamic Driving Task</i>
<i>DVM</i>	<i>Driver-Vigilance Monitoring</i>
<i>EA</i>	<i>Enterprise Architect</i>
<i>ECU</i>	<i>Electronic Control Unit</i>
<i>ESC</i>	<i>Electronic Stability Control</i>
<i>Euro-NCAP</i>	<i>European New Car Assessment Program</i>
<i>FCW</i>	<i>Forward Collision Warning</i>
<i>FOV</i>	<i>Field of View</i>
<i>GPS</i>	<i>Global Positioning System</i>
<i>HARA</i>	<i>Hazard Analysis and Risk Assessment</i>
<i>HMI</i>	<i>Human Machine Interface</i>
<i>HWA</i>	<i>Highway Assist</i>
<i>HWP</i>	<i>Highway Pilot</i>
<i>IEEE</i>	<i>Institute of Electrical and Electronics Engineers</i>
<i>ISO</i>	<i>International Organization for Standardization</i>
<i>LDW</i>	<i>Lane Departure Warning</i>

<i>LIDAR</i>	<i>Light Detection and Ranging</i>
<i>LKA</i>	<i>Lane-Keeping Assistant</i>
<i>LKS</i>	<i>Lane-Keeping Support</i>
<i>MBSE</i>	<i>Model Based System Engineering</i>
<i>MDG</i>	<i>Model Driven Generation</i>
<i>NS</i>	<i>Navigation System</i>
<i>NHTSA</i>	<i>National Highway Traffic Safety Administration</i>
<i>PAM</i>	<i>Process Assessment Model</i>
<i>ASPICE</i>	<i>Automotive Software Process Improvement and Capability dEtermination</i>
<i>SAE</i>	<i>Society of Automotive Engineers</i>
<i>SysML</i>	<i>System Modeling Language</i>
<i>OEDR</i>	<i>Object and Event Detection and Recognition</i>
<i>OEM</i>	<i>Original Equipment Manufacture</i>
<i>VRU</i>	<i>Vulnerable Road User</i>

Lista de figures

Figura 1 - Configuracions de la geometria del àrea manejable _____	18
Figura 2 - Exemple de sistema d'avís de col·lisió frontal (FCWS) _____	18
Figura 3 – Exemple de control de creuer adaptatiu (ACC) _____	19
Figura 4 – Exemple de highway assist (HWA) _____	19
Figura 5 - Resum nivells autonomia SAE _____	21
Figura 6 – Els errors són més barats d'eliminar en fases prematures _____	29
Figura 7 - correlació ús de SE i rendiment mitjà en projectes _____	31
Figura 8 - Cicle de Validació-Verificació del model V _____	32
Figura 9 - Pilars del modelat de l'Enginyeria de Sistemes _____	33
Figura 10 - Procés Harmony _____	36
Figura 11 - Procés de disseny i especificació del sistema _____	38
Figura 12 - Metodologia ARCADIA _____	39
Figura 13 - Diagrames SysML _____	40
Figura 14 - <i>Use Case Diagram</i> _____	41
Figura 15 - <i>Activity Diagram</i> _____	42
Figura 16 - <i>Sequence Diagram</i> _____	43
Figura 17 - <i>State Machine Diagram</i> _____	43
Figura 18 - <i>Block Definition Diagram</i> _____	44
Figura 19 - <i>Internal Block Diagram</i> _____	46
Figura 20 - <i>Package Diagram</i> _____	47
Figura 21 - Polarion _____	47
Figura 22 - Sparx EA i PCS _____	48
Figura 23 - Exemple Meta-model _____	49
Figura 24 - Relació del PAM _____	52
Figura 25 - Resum del PRM d'Automotive SPICE _____	53
Figura 26 - <i>Stakeholder Onion Diagram</i> _____	64
Figura 27 - Captura de Polarion mostrant requeriments de sistema _____	66

Figura 28 - <i>Stakeholder Diagram HWP</i>	68
Figura 29 - Stakeholder Diagram AEB	69
Figura 30 - Use case ADAS CAV	70
Figura 31 - Use case Avoid Collisions	71
Figura 32 - Use Case AEB	72
Figura 33 - Procés de tecnologia MDG	77
Figura 34 - Ontologia IDIADA	78
Figura 35 - Generació MDG	80
Figura 36 - MDG <i>packages</i>	80
Figura 37 - Diagrama de paquets AEB	81
Figura 38 - Artefactes A-SPICE	83
Figura 39 - Productes finals	85
Figura 40 - Captura ODD	87
Figura 41 - System Automotive Context	88
Figura 42 - System Vehicle Context	89
Figura 43 - Captura arxiu Simulink AEB	91
Figura 44 - System of Interest Structure	93
Figura 45 - System of Interest Software Structure	94
Figura 46 - Software Structure Sub-Systems	95
Figura 47 - System of Interest Definition	96
Figura 48 - Feature Functionality	97
Figura 49 - System of Interest Interfaces	98
Figura 50 - Value Types	99
Figura 51 - Feature Decomposition I	102
Figura 52 - Feature Decomposition II	103
Figura 53 - Feature Decomposition III	104
Figura 54 - AEB System Operation Modes	105
Figura 55 - Allocation of Functions to Software Components	107
Figura 56 - System elements to System Operations Mode	108

Figura 57 - Traceability DBS _____	110
Figura 58 - Traceability providing AEB _____	111
Figura 59 - Traceability user interface _____	112
Figura 60 - Matriu de traçabilitat _____	114
Figura 61 - Resum d'activitats disseny ADAS _____	117

Llista de taules

Taula 1— Primary life cycle processes – SYS process group.....	53
Taula 2 - Resum procés SYS.3	55
Taula 3 - Característiques del " <i>System architectural design</i> ".....	56
Taula 4 - Producte final d'arquitectura de sistema	58
Taula 5 - Taula resum <i>WP</i>	114

1. Introducció

1.1 Motivació

La gran majoria de vehicles al món operen en carretera, més que sobre rails, per aire o per mar. La congestió de les carreteres ha desenvolupat en un seriós problema en moltes àrees urbanes. La principal causa de la congestió és la demanda de mobilitat la qual ha excedit la capacitat actual de les principals carreteres. L'estat d'aquestes no és capaç de gestionar tot aquest gran volum d'una manera fluida el que resulta en més congestió, embussos, retards innecessaris i accidents.

Els accidents de tràfic son una de les principals causes de mort a en països desenvolupats, tant sols a Estats Units en l'any 2020 va tenir més de 38.000 morts implicats en accidents de tràfic segons dades recollides per la NHTSA [1], i la conducció autònoma té el potencial de reduir-los de forma dràstica. A més, una conducció totalment autònoma podria abaratir els costos de transport de persones i mercaderies i alliberar de temps als conductors. Les intensives emissions de gasos contaminants, degudes en gran mesura a embussos de trànsit també afecten a nivell mediambiental fins a nivells crítics i crea un creixent problema de salut pública.

Aquest tipus de sistemes prometen portar grans millores en termes de seguretat, estabilitat, connectivitat, etc. Tot, renunciant parcial o completament del factor humà. Els constants progressos en el desenvolupament de sistemes robusts de control, percepció i comunicacions, han obert un sender de evolució gairebé imparabile cap a dita autonomia.

1.2 Objectius

L'objectiu principal del treball de fi de grau es especificar una funció *Highway Pilot (HWP)* per la finalment intentar integrar-la en un cotxe prototip.

Els objectius específics del projecte son:

- Anàlisis del requeriments del les parts interessades (*stakeholders*).
- Definició dels *Operational Design Domain (ODD)* [11] per a la funció HWP.
- Especificació dels requeriments a nivell de sistema.

- Avaluació de possibles arquitectures de sistema.

1.2.1 Abast del treball

La intenció del treball és definir una Arquitectura de Sistema tenint com a exemple el projecte en possessió de Applus+ Idiada, que es pugui reutilitzar a posteriori. Aquest projecte es una funció ADAS *Highway Pilot* provinent d'un projecte europeu d'Idiada. Aquesta definició del procés es crearà per tal d'obtenir un procés estandarditzat el qual seguirà el model Automotive SPICE. La definició de l'Arquitectura de Sistema es crearà seguint el model de llenguatge *System Modeling Language (SysML)* i en base a les directrius *Automotive SPICE*.

Gràcies a l'aplicació dels conceptes relatius a l'Arquitectura de Sistema, la comprensió del nostre sistema i la seva traçabilitat milloraran, incrementant alhora el nivell d'ASPICE del mateix.

1.3 Estructura

Bàsicament l'estructura del treball determinarà els passos a seguir. Primer de tot hem de saber tots els detalls del sistema amb el que treballarem, això implica revisar el projecte a *Polarion* i veure quins requisits tenim i d'on provenen (regulacions europees i d'estandardització), entendre el sistema i a partir d'aquí abordar l'enginyeria de sistemes juntament amb el *MBSE*. També hem d'aprendre els bàsics del *SysML* per saber de quines opcions té i quina manera podrem modelar el sistema.

Per últim hem d'instal·lar l'eina que farem servir en la majoria del projecte, que és l'*Enterprise Architect* i també fer una recerca de com unim el nostre propòsit amb el que podem fer realment dintre l'eina sent conscient que el nostre coneixement és un limitant. Llavors fent un seguiment de la guia *A-SPICE* definirem una ontologia per poder desenvolupar finalment els diagrames que ens serviran com arquitectura del sistema.

2. Estat de l'art

Abans de començar a explicar el contingut d'aquest treball, es necessari posar-lo en context mitjançant el seu estat de l'art. En aquest capítol s'intentarà exposar la situació actual del vehicle autònom com de tots els detalls que amaga aquesta tecnologia.

2.1 Vehicle autònom

Un vehicle autònom (AV) és un vehicle que té la capacitat de conduir sense el control físic actiu o vigilància per un operador humà [5].

Els cotxes autònoms detecten al seu voltant mitjançant l'ús de radars, làsers, sistema de posicionament global i visió computeritzada. La informació recollida pels sensors es interpreta per sistemes avançats de control per identificar rutes de navegació apropiades, així com obstacles i senyals de trànsit [6].

2.1.1 Classificació i Nomenclatura dels Sistemes

Actualment existeixen dos sistemes proposats de classificació pels cotxe autònoms: el de la *National Highway Traffic Safety Administration* (NHTSA) als Estats Units i el la *Society of Automotive Engineers* (SAE). La SAE com a societat de d'enginyers d'automoció va néixer el 1905 a EEUU [8] amb la intenció de que els tècnics poguessin difondre idees i conèixer novetats, així com per promoure estàndards en la indústria.

Avui en dia s'està estenent i acceptant cada vegada més l'estàndard creat per la SAE, de fet la NHTSA ha adoptat aquest sistema de classificació al setembre de 2016, abandonant el seu propi.

2.1.1.1 Estàndard SAE J316

La SAE distingeix entre sis nivells d'autonomia que seran descrits a continuació: SAE Level 0, SAE Level 1, SAE Level 2, SAE Level 3, SAE Level 4 i SAE Level 5. A continuació podem veure a la figura 1 un resum de diferències entre nivells.

L'estandardització d'aquests 6 nivells té uns objectius principals:

- Identificar sis nivells de conducció autònoma, des de "sense automatització" fins a "automatització completa".
- Establir una base de nivells i definicions sobre aspectes funcionals de la tecnologia.
- Descriure les diferències de forma categòrica per una progressió gradual a través dels diferents nivells d'automatització.
- Ser consistents amb les pràctiques actuals de la indústria.

- Aclarir les possibles confusions i ser útil en nombroses disciplines (enginyeria, legal, medis i discurs públic).
- Educar a una comunitat més ampla, aclarint per cada nivell quin es el paper (si n'hi ha) dels conductors en la realització de la tasca de la conducció dinàmica mentre es fa servir un sistema de conducció automàtica.

D'acord amb l'estàndard *SAE J3016* de nivells de conducció autònoma, per classificar qualsevol tipus de vehicle hi ha quatre aspectes fonamentals per determinar el nivells d'automatització del mateix.

-Qui s'encarrega del **moviment** del vehicle (l'humà o la màquina). El moviment es diferencia entre longitudinal (es a dir, accelerar i frenar) i lateral (es a dir, la direcció).

-Qui s'encarrega de la **detecció i resposta** davant objectes i eventualitats (OEDR, per les seves sigles en anglès), mitjançant sistemes que monitoritzen l'entorn del vehicle durant la conducció (l'humà o la màquina).

-Qui s'encarrega del **suport de la conducció**, es a dir, qui actua en cas de error dels sistemes automatitzats, o davant la pèrdua de les condicions pel seu funcionament (de nou, l'humà o la màquina).

-**Condicions específiques** pel funcionament del sistema (d'horari, climatològiques, geogràfiques, tipus de carretera, quantitat de trànsit, velocitat, etc.) Es a dir, el sistema de conducció automatitzada funciona en tots els casos, o només en certes condicions que el limiten.

Explicuem una mica millor el moviment: en el funcionament del vehicle es distingeixen dos moviments principals.

El control de **moviment longitudinal** : això es refereix a que el sistema de conducció automatitzada mantingui una velocitat , es detecti el vehicle que ens precedeix en el carril, es mantingui la distancia de seguretat apropiada amb ell i s'actui sobre l'accelerador o el fre segons sigui pertinent. Això seria per exemple la que fa uns sistema de control de velocitat de creuer adaptatiu (ACC).

Control de **moviment lateral**: això es refereix a que el sistema de conducció automatitzada detecti la posició del vehicle respecte als límits del carril i sigui capaç d'actuar sobre la direcció (o els frens) per mantenir-se dintre del mateix apropiadament, inclús per prendre corbes. Això seria per exemple, les funcions d'un *Lane Keeping System (LKS)*.

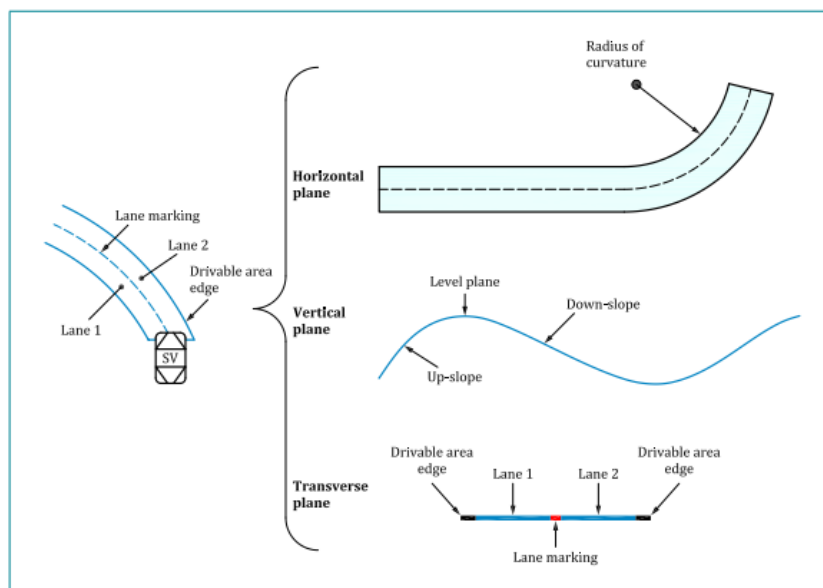


Figura 1 - Configuracions de la geometria del àrea manejable

Font: [10]

Nivells:

- **SAE Level 0:** Sense automatització de la conducció, control total manual del conductor. Funcions limitades a alertar al conductor.

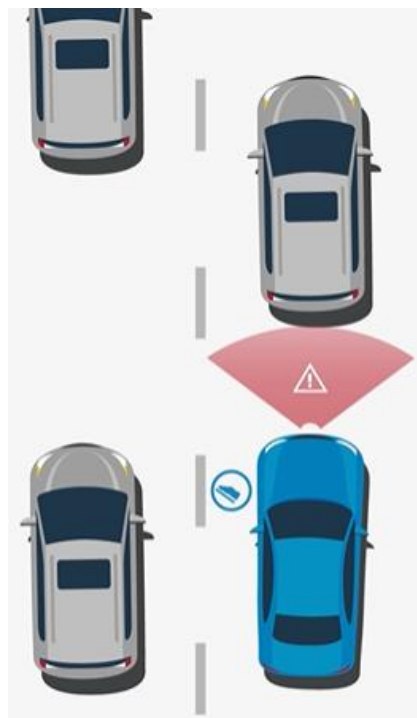


Figura 2 - Exemple de sistema d'avís de col·lisió frontal (FCWS)

- **SAE Level 1:** Son les comunament conegudes com ADAS, on el sistema realitza un control longitudinal o lateral del vehicle. Aquestes funcions estan disponibles en alguns modes de conducció, mentre els conductors humans son responsables de monitoritzar l'entorn i son el *fallback* (últim recurs) en les *Dynamic Driving Task (DDT)*. Com a exemple trobem els sistemes *Adaptive Cruise Control (ACC)* o *Lane Keeping Assist (LKS)*.

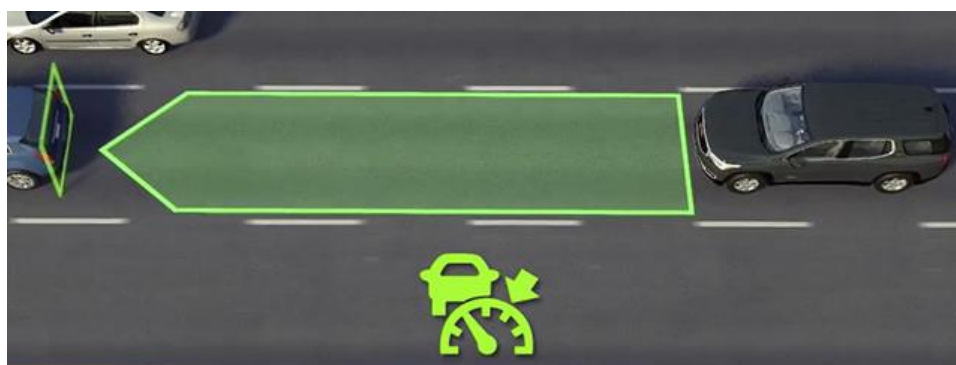


Figura 3 – Exemple de control de creuer adaptatiu (ACC)

- **SAE Level 2:** Aquests sistemes controlen longitudinalment i lateralment el moviment del vehicle. Es a dir, el nivell 2 es refereix a la habilitat del vehicle per girar, frenar o accelerar. Ho es, tot i així, només es possible en una condicions de carretera particulars i amb la interacció del conductor, es a dir, el conductor humà ha de monitoritzar el sistema i és el *fallback* pels esdeveniments dinàmics de conducció. El sistema *Highway Assist (HWA)* és un exemple de tal tecnologia que ja està disponible al mercat.

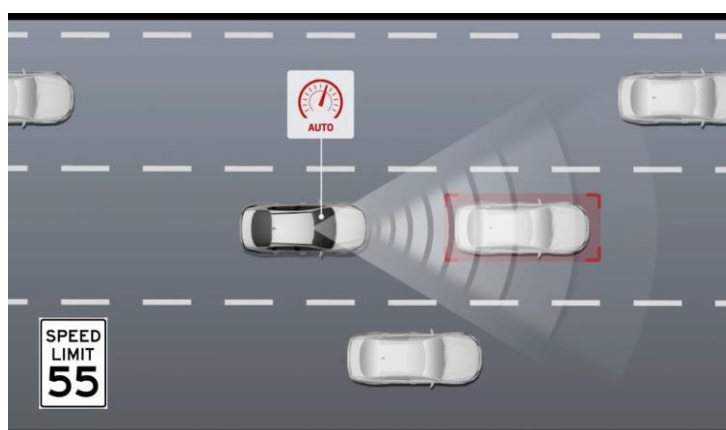


Figura 4 – Exemple de highway assist (HWA)

- **SAE Level 3:** Del nivell 3 cap amunt, el sistema per si mateix és responsable de controlar de l'entorn. Això implica requeriments molt alts pels sensors de reconeixement de l'entorn. El conductor humà és encara l'últim recurs (*fallback*) en cas d'emergència pels esdeveniments dinàmics de conducció. Una última regulació UN Regulation No. 157 [9] ha entrat en vigor el gener del 2023 on el conductor ja pot deixar de tenir les mans sobre el volant, tot i que per altra banda ha de controlar que aquest està llest per realitzar un *fallback* a través d'una monitorització activa de l'estat del conductor.
- **SAE Level 4:** És més o menys el mateix que el nivell 3 amb la diferència que el conductor humà ja NO és el responsable de *l'últim recurs* per les DDT. El sistema ha de ser capaç de manejar aquestes situacions per ell mateix. Un exemple d'aquest tipus serien les xarxes de taxi autònoms locals.
- **SAE Level 5:** Automatització de la conducció completa. El vehicle pot ser conduït de manera autònoma sota qualsevol circumstància i en qualsevol lloc.

Hi ha molta recerca per aconseguir nivells més alts d'automoció, però tals tecnologies actualment només estan disponibles en simulacions o prototips. Això podria canviar en els pròxims anys.

Level	Name	Narrative definition	DDT		DDT fallback	ODD
			Sustained lateral and longitudinal vehicle motion control	OEDR		
Driver performs part or all of the DDT						
0	No Driving Automation	The performance by the <i>driver</i> of the entire DDT, even when enhanced by <i>active safety systems</i> .	<i>Driver</i>	<i>Driver</i>	<i>Driver</i>	n/a
1	Driver Assistance	The <i>sustained</i> and ODD-specific execution by a <i>driving automation system</i> of either the <i>lateral</i> or the <i>longitudinal vehicle motion control</i> subtask of the DDT (but not both simultaneously) with the expectation that the <i>driver</i> performs the remainder of the DDT.	<i>Driver and System</i>	<i>Driver</i>	<i>Driver</i>	Limited
2	Partial Driving Automation	The <i>sustained</i> and ODD-specific execution by a <i>driving automation system</i> of both the <i>lateral</i> and <i>longitudinal vehicle motion control</i> subtasks of the DDT with the expectation that the <i>driver</i> completes the OEDR subtask and <i>supervises</i> the <i>driving automation system</i> .	System	<i>Driver</i>	<i>Driver</i>	Limited
ADS ("System") performs the entire DDT (while engaged)						
3	Conditional Driving Automation	The <i>sustained</i> and ODD-specific performance by an ADS of the entire DDT with the expectation that the <i>DDT fallback-ready user</i> is <i>receptive</i> to ADS-issued requests to <i>intervene</i> , as well as to DDT performance-relevant system failures in other vehicle systems, and will respond appropriately.	<i>System</i>	System	<i>Fallback-ready user (becomes the driver during fallback)</i>	Limited
4	High Driving Automation	The <i>sustained</i> and ODD-specific performance by an ADS of the entire DDT and DDT fallback without any expectation that a <i>user</i> will respond to a request to <i>intervene</i> .	<i>System</i>	<i>System</i>	System	Limited
5	Full Driving Automation	The <i>sustained</i> and unconditional (i.e., not ODD-specific) performance by an ADS of the entire DDT and DDT fallback without any expectation that a <i>user</i> will respond to a request to <i>intervene</i> .	<i>System</i>	<i>System</i>	<i>System</i>	Unlimited

Figura 5 - Resum nivells autonomia SAE

Font: [7]

2.2 ADAS

El terme *Advanced Driver Assistance System (ADAS)* descriu sistemes en un vehicle que donen suport a totes les tasques que desenvolupa conduint. *Donges* [10] ens ensenya que les tasques de conducció es poden dividir en tres nivells:

- navegació,
- orientació i
- estabilització.

Per cada *Driving Task (DT)*, ADAS especials estan disponibles per ajudar al conductor. Per exemple, un *Navigation System (NS)* dona suport al conductor en aspectes de navegació,

un *Adaptative Cruise Control* (ACC) en aspectes d' orientació, i el *Electronic Stability Control* (ESC) a nivells d'estabilització.

Les ajudes al conductor poden provenir via

-avisos òptics, acústics o hàptics i/o

-intervencions de les ADAS automàtiques o iniciades pel conductor.

Una *Lane-Departure Warning* (LDW) és un avís del sistema que informa el conductor que està apunt de sortir del carril sense voler. Per altra banda, un *Lane-Keeping Assistant* (LKA) ajuda al conductor a mantenir el vehicle dintre del carril a partir de aplicar intervencions al vehicle.

Una altra manera de categoritzar les ADAS és distingint entre sistemes orientats al confort i orientats a la seguretat. La principal funció de les ADAS orientades al confort és assistir al conductor encarregant-se de *DTs*. En la direcció longitudinal, els sistemes *ACC* ajuden al conductor, mentre que un sistema *LKA* dona suport en la direcció lateral. Un *Automatic Emergency Brake* (AEB) és un sistema orientat a la seguretat que hauria de prevenir col·lisions entre vehicles.

Tot això sobre el paper podria tenir aquesta primera classificació, no obstant, les ADAS no poden ser assignades exclusivament a una de les categories descrites anteriorment. Tal com hem dit, un *ACC* és un sistema orientat a la comoditat. *Eichberger* va demostrar [12] que un *ACC* també incrementa la seguretat utilitzant un anàlisi detallat de 217 accidents fatídics de la base de dades ZEDATU (Àustria) de l'any 2003. Aquesta recerca va mostrar que un *ACC* hauria previngut 8 accidents i hauria reduït la severitat d'uns altres 8 accidents, representant al voltant d'un sis per cent del total d'accidents. En comparació, un *AEB*, el que es classifica com a sistema orientat a la seguretat te un potencial d'un vint-i-un per cent. Aquest exemple ens mostra perquè els sistemes no poden ser classificats clarament.

2.2.1 Infraestructura reguladora

El marc regulador per les ADAS es pot dividir en diferents nivells, els quals estan explicats en les properes seccions.

2.2.1.1 Tractats Internacionals

L'objectiu dels tractats internacionals es harmonitzar legislacions nacionals. D'aquí, la *Convention on Road Traffic* va ser presentat l'any 1968. Per definir tals factors com estàndards

de la normativa de conducció i l'homologació per vehicles i requisits del conductor [14]. Aquest tractat va ser aprovat per gairebé tots els països europeus i en nombrosos països arreu del món, que es van comprometre a adoptar tals continguts a les seves lleis nacionals.

El cinquè paràgraf de l'article vuit esmenta:

"Tot conductor tindrà el control en tot moment del seu vehicle o de guiar els seus animals"

Del primer paràgraf de l'article 13 esmenta:

"Tot conductor de vehicle ha de trobar-se en condicions i la posició de poder exercir control del seu vehicle en qualsevol circumstància i totes aquelles maniobres que se'l requereixin"

D'aquestes declaracions es podria deduir que només les ADAS que poden ser anul·lades pel conductor serien permeses. Tot i així, aquesta interpretació és incompatible amb la finalitat d'introduir vehicles més altament automatitzats però s'hi han afegit nous canvis en la Convenció. Aquests afegeixen que mentre aquets sistemes que influencien en la conducció han de ser considerats en conformitat amb les dues premisses abans esmentades, com també amb les regulacions internacionals de construcció, implementació i d'ús. En el nostre cas del HWP, la recent regulació es la UN R.157 [15].

2.2.1.2 Lleis nacionals i homologació

Per l'homologació, els vehicles han de complir amb les regulacions, com ara la *ECE Regulations* a Europa. Per ESC, es descriu un procediment per un test especial [16]. El mateix procediment es troba a *United States Law* [17]. A Europa, per la funció de frenada de un ACC o AEB, el sistema ha d'ajustar-se a la normativa d'acord als requeriments de l'annex vuit de ECE-R 13-H [16], que només defineix els requisits pel procés de disseny de sistemes de control complexos d'electrònica de vehicle. No hi ha límits en quant a característiques físiques, com l'acceleració.

2.2.1.3 Estàndards i test de consumidor

Tot i que els requeriments derivats dels estàndards i els tests de consumidor no son obligatoris, encara son molt importants pels fabricants de vehicles. Els estàndards descriuen les últimes tecnologies disponibles per a la majoria d'ADAS. Per un sistema ACC, la *International Organization for Standardization (ISO)* i la *SAE* han definit estàndards [18][19].

Allí, els sistemes ACC estan dividits en diferents classes, depenent del seu rendiment en sensors de reconeixement de l'entorn.

A Europa, el *European New Car Assessment Program* (Euro-NCAP) inclou algunes ADAS en la seva avaluació, inclou *Seat Belt Reminders*, *Speed Assist Systems*, AEB i *Lane Support Systems* [20]. Des que la avaluació Euro-NCAP és un punt important de venda, els fabricants de cotxes es veuen forçats a introduir aquests sistemes de seguretat en els seus nous vehicles. Euro-NCAP [21] ja ha anunciat que tests més complexes es definirien en els propers anys, ja que s'espera un millor rendiment dels sistemes degut a la extensa recerca dels fabricants de vehicles.

Als Estats Units la *Insurance Institute of Highway Safety* (IIHS) estableix estàndards pels tests a sistemes AEB. Per altra banda a *United States New Car Assessment Program* (US-NCAP) avalua ESC, LDW i LKS, FCW i AEB.

Les condicions frontera que hem descrit només ens serveixen per fer-nos una idea aproximada de com es dissenyen les ADAS, no obstant, han de ser complertes per assegurar una conformitat amb les lleis i la homologació encara que, el rendiment de les ADAS depèn de l'estratègia del fabricant en qüestió, i del disseny i validació dels enginyers de sistema.

3. Funció Highway Pilot – Procés Tècnic

3.1 Descripció general

3.1.1 Resum del producte

La funció principal de la funció *Highway Pilot* és controlar el moviment longitudinal i lateral del vehicle. Això s'aconsegueix controlant les entrades de l'accelerador, el frenat i la direcció al vehicle subjecte per tal d'adaptar la velocitat a la velocitat desitjada. Això també té en compte els vehicles circumdants, la geometria i les condicions de la carretera, i opcionalment les limitacions legals, i mantenir el vehicle subjecte dins dels límits del carril.

Això es recolza mitjançant l'ús d'informació sobre la distància per avançar els vehicles, la informació del carril, el moviment del vehicle subjecte i els paràmetres que es poden configurar per part del conductor. A partir d'aquesta informació, la funció envia ordres als actuadors i mostra informació d'estat al conductor. Aquesta funció és capaç de donar suport al conductor tant quan condueix per carreteres obertes com quan es troba un trànsit més lent.

Opcionalment, el conductor pot triar que la funció *Highway Pilot* rebi els consells de velocitat establerts d'altres funcionalitats ADAS automàticament o amb una confirmació prèvia del conductor (per exemple, reconeixement de senyals de trànsit, Assistència de velocitat intel·ligent, etc.).

La funció *Highway Pilot* s'introdueix per augmentar la comoditat de conducció en la tasca del conductor, però el conductor sempre és responsable de la vigilància i la consciència a l'hora d'ajustar la velocitat, la frenada, els obstacles al carril o controlar el vehicle d'una altra manera.

La funció *Highway Pilot* està de manera predeterminada en un estat DESACTIVAT a l'engegada del sistema del vehicle. Mentre es troba en un estat DESACTIVAT, la funció no proporciona un indicador d'estat al controlador i no realitza cap acció. L'estat ENABLED és un estat d'alt nivell que engloba els estats *STANDBY* i *ACTIVE*. En un estat *STANDBY*, la funció ofereix una indicació d'estat, es supervisa l'entorn i, si es compleixen les condicions, el conductor pot activar la funció HWP. Quan està en un estat *ACTIVE*, la funció proporciona un indicador d'estat al conductor i controla la velocitat del vehicle i la posició lateral del vehicle subjecte al carril.

Si es detecta una fallada, mentre la funció HWP es troba en un estat *ACTIVE*, la funció passarà a un estat FALLA, o si la funció HWP no estava en estat *ACTIVE* anteriorment (és a

dir, en un estat *OFF*), el conductor no podrà habilitar la funció i en conseqüència s'indicarà com a tal en un indicador d'estat.

3.1.2 Solució real

Tal com he assenyalat, la motivació del projecte es el desenvolupament d'una funció ADAS com és el HWP, tot i així, un cop avaluant la complexitat del sistema que afrontem he decidit que dividirem el sistema en un banc de funcions de ADAS, que intrínsecament estan sota l'arbre que és un sistema de tanta autonomia i de nivell tant alt d'estàndard *SAE J316*. Més en concret ficarem el focus en la funció AEB, que està dos nivells per sota del HWP. Això servirà com a "base de dades" a IDIADA, ja que serà el primer pas per crear aquest banc de ADAS que quedaran construïdes pels seus subsistemes que en un treball posterior ens servirà per proveir al client de unes funcions més personalitzades com ara la del HWP.

3.2 Metodologia

3.2.1 Enginyeria de sistemes

Tal com diu la NASA [29], "*Systems Engineering*" (enginyeria de sistemes) es descriu com un enfocament metòdic i multi-disciplinar pel disseny, realització, direcció tècnica operacions, i retirament d'un sistema. També entendre les interaccions del sistema amb l'exterior i les pròpies dintre del sistema. Però la finalitat és guiar el procés d'enginyeria per tal que el sistema pugui ser implementat i que resulti exitós.

Així en poques paraules ho podríem definir com una manera de desenvolupar projectes complexes que satisfacin objectius d'una manera eficient. Què implica fer-ho d'una manera eficient? Hi ha tres conceptes clau per poder respondre. Minimitzar que s'hagi de refer, descobrir errors en una fase prematura i comunicar-ho de manera eficient.

Els integrants més determinants d'un sistema són els següents:

- **Stakeholders** (o parts interessades). Tenen el control sobre la gestió del projecte i els requisits del sistema
- **Actors**. Representa totes aquelles parts que interactuen directament amb el sistema sota estudi. Per exemple, en el nostre cas per desenvolupar un projecte d'automoció, tindrem el conductor. Són aquests els que realment donaran lloc a

les diferents relacions i els que permetran el disseny del sistema, ja que de les seves interaccions en depèn que se satisfan els requisits o no.

- **Relacions.** Molt en la línia amb la definició d'actors, aquestes relacions representen com n'és la interacció amb el sistema. Així, es permet la descripció de diferents escenaris i capacitats per a situacions reals que permeten entendre en més profunditat el comportament del sistema i ajudar a identificar les restriccions
- **Entorn.** La definició del límit del sistema es essencial. Representant els diferents escenaris sota els que puguem dissenyar i avaluar el nostre sistema, ja que, encara que es doni el cas en que un sistema obté grans rendiments per a un entorn i una situació concreta, no sempre es complirà en totes les circumstàncies, sent primordial per tant identificar l'entorn en el qual ha de desenvolupar el nostre disseny, per exemple, en aquest cas una carretera determinada.

3.2.1.1 Sistema, Element de Sistema i Funció del Sistema

En aquesta part es descriurà breument el coneixement bàsic requerit per la correcta comprensió del treball. Alguns conceptes poden ser abstractes però més tard en aquest projecte, alguns d'ells, es mostraran com un exemple per tal que siguin més fàcil d'entendre.

Per entendre de manera correcta el projecte, es necessari començar per definir els conceptes més bàsics, com serà el concepte de **sistema**.

Un sistema és un grup d'entitats que treballen conjuntament per complir una tasca. La tasca pot ser portada a terme per un, o més sistemes. Un sistema pot ser un sub-sistema d'un sistema superior. Aquests sistemes superiors comunament anomenats Sistema de Sistemes [28].

La definició de sistema és altament complex degut a totes les formes que pot adoptar. Per posar-li solució a aquest problema s'han fet una sèrie de definicions per alguns experts que exposarem aquí:

- "Un sistema és un objecte que entrega valor" [31]
- "Un sistema es defineix com una sèrie de conceptes i/o elements usats per satisfer una necessitat o requeriment"[32]

- “Un sistema és un conjunt de components dissenyats per complir amb un particular objectiu seguint un pla”[33]

Els “conceptes”, “elements” or “components”, que aquestes definicions descriuen com a parts que conformen un sistema, també es coneixen com **elements de sistema**. Aquest serà el terme per referir-nos a ells en aquest projecte. Aquests elements de sistema són realment importants pel disseny de l'arquitectura del sistema perquè ens ajuden a dividir un sistema molt extens i complicat en múltiples parts més reduïdes que ens permetrà analitzar-lo més fàcilment. Aquest conceptes doncs serà àmpliament utilitzat en la definició del sistema.

Retornant a les definicions del sistema que hem fet anteriorment, “usat per satisfer una necessitat o requeriment”, per tant té un “objectiu particular”. Per aconseguir aquest objectiu el sistema té, el que es coneix com, **funcions de sistema**. Es possible definir una funció de sistema com accions o un conjunt d'accions executades pel sistema al qual pertanyen per arribar al objectiu. Aquest conceptes s'utilitzarà amb freqüència en la vista/definició dinàmica del sistema.

3.2.1.2 Arquitectura de sistema

Per definir exactament què significa Arquitectura de Sistema, es va fer una definició per l'organització INCOSE (*International Council on System Engineering*) [34]. Defineixen l'arquitectura del sistema com “el conglomerat de funcions d'un sistema descompost en elements que interactuen entre ells, amb uns requeriments que inclouen aquesta associació amb les funcions de sistema agregades i els seus requeriments/definició d'interfícies”. L'organització de la INCOSE aclareix aquesta descripció quan diu, “Quan s'utilitza com un nom, el Disseny de Sistema és el mateix que l'Arquitectura de Sistema”. Aquesta descripció de l'arquitectura de sistema es molt complexa però, ahora, molt precisa.

Per descriure què és una arquitectura de sistema d'una forma senzilla, es podria dir que es un model conceptual que defineix diferents maneres de veure un mateix sistema. Sent l'estructura (referint-se als elements de sistema i les interfícies) i el comportament (referint-se a les funcions de sistema) son el més important.

3.2.2 Model-Based Systems Engineering

Model Based Systems Engineering (MBSE) [35] representa la metodologia per representar i donar suport a les activitats de l'enginyeria de sistemes, com ara requeriments de sistema, disseny anàlisis, verificació i validació. *Systems Modeling Language (SysML)* i MBSE

pot adreçar els reptes de manejar sistemes complexes, fent un manteniment i mapejar les arquitectures de sistema i software, i canviant els requeriments de comportament.

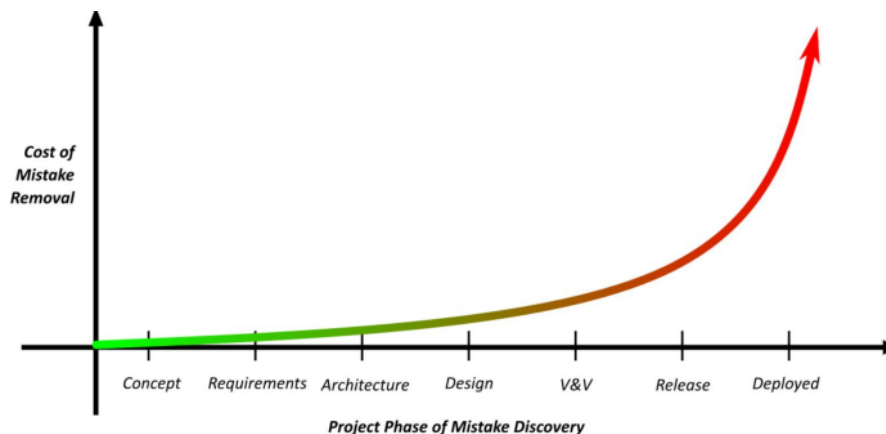


Figura 6 – Els errors són més barats d'eliminar en fases prematures

[27]

Tradicionalment els mètodes utilitzats en l'enginyeria de sistemes són basats en la documentació i això pot portar a problemes com en la comprensió completa de requeriments textuais i subjecte a interpretacions individuals, falta de consistència entre els elements del sistema, una traçabilitat incompleta en tots els sentits dels artefactes. Un model de sistema creat amb la capacitat de simular el comportament del sistema pot ajudar incrementant el nivell de comprensió del comportament funcional del sistema, reduint en fases de desenvolupament mitjançant la detecció d'anomalies en requeriments en fases prematures del *life cycle*. També és útil definint les interfícies adequades, límits funcionals com els *inputs* per la implementació i d'aquesta manera ens permetrà definir l'arquitectura de sistema per complet pel sistema que considerem.

3.2.2.1 Justificació de l'exercici de l'Enginyeria de Sistemes basada en Models

L'objectiu d'aquesta revisió bibliogràfica [19] va ser la cerca de casos reals industrials que poguessin informar sobre la decisió de recolzar o no el procés de canvi, la inversió, la formació i les eines necessàries per implantar un enfocament MBSE a l'empresa. La pregunta que es va plantejar va ser, com es justifica un canvi d'un enfocament d'enginyeria de sistemes basat en documents (*DBSE*) a un enfocament d'enginyeria de sistemes basat en models (*MBSE*)?

Per a la realització d'aquest estudi es van fer servir documents publicats per autors dels sectors de defensa, espai i l'enginyeria de productes de sistemes complexos, on alguns van

utilitzar mètriques per tractar d'analitzar els resultats i altres no. Els 67 estudis de casos sense mètrica van atribuir l'èxit principalment a la integritat, la coherència i la comunicació dels requisits mentre que els 21 estudis de casos amb mètriques sobre el cost i el calendari van atribuir l'èxit principalment a la capacitat d'un enfocament MBSE per millorar les estratègies de prevenció d'errors.

Els estudis de casos que no incloïen mètriques quantificables abordaven principalment el procés pel qual un enfocament MBSE podia integrar-se al desenvolupament d'un sistema complex, sobretot per a l'enginyeria de requisits. Els principals beneficis que es van obtenir van ser una anàlisi més meticulosa per trobar un conjunt més complet de requisits, consistència i millores en la comunicació entre equips de disseny i enginyers, que tenien una millor visió i comprensió dels requisits i relacions entre ells. Altres àrees a les que també es van obtenir beneficis van ser a la Validació i Verificació, l'explotació del concepte, el reús dels dissenys i l'anàlisi de marges del sistema. Alguns d'aquests estudis també van abordar els desafiaments experimentats en la transició de DSBE a MBSE, sent la manca d'enginyers qualificats un de els principals problemes detectats, donant lloc a barreres en l'adopció d'eines de modelatge d'Enginyeria de Sistemes, els processos de desenvolupament de models i les diferències entre els equips de disseny i les parts interessades.

Pel que fa als casos d'estudi en què es van fer servir mètriques, es van analitzar segons aquells que documentaven una guia amb canvis en processos d'enginyeria i aquells que justificaven la implementació d'aquesta nova metodologia a partir de reduccions en costos i temps, i d'altres basats en la detecció primerenca d'errors. S'enumeren a continuació alguns dels resultats principals obtinguts dels estudis.

En primer lloc, Goldenson i Elm [44], al seu estudi "*The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey,*" va poder documentar una gran correlació entre l'ús de SE i el rendiment mitjà en projectes (Figura 7).

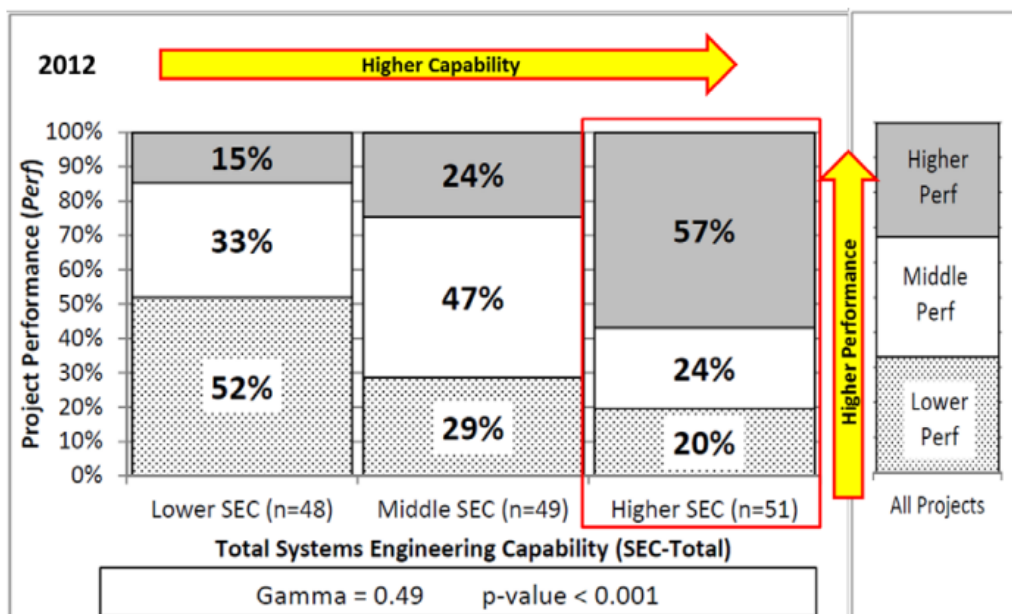


Figura 7 - correlació ús de SE i rendiment mitjà en projectes

[41]

3.2.3 Methodologies

3.2.3.1 Model en V

És interessant dedicar una secció al model en V, ja que es tracta del model de cicle de vida de desenvolupament més estès en l'indústria relacionada amb l'enginyeria de sistemes. El model cobreix des de que identifiquem les necessitats, trobem les activitats que es portaran a terme, el pla que es durà a terme, l'execució detallada de les activitats, així com la responsabilitat que se'n extreu, i el registre que tots aquestes parts comporten. Aquests processos són únicament aplicables a requisits tècnics productes que tenen impacte directe en la forma o funció del producte final.

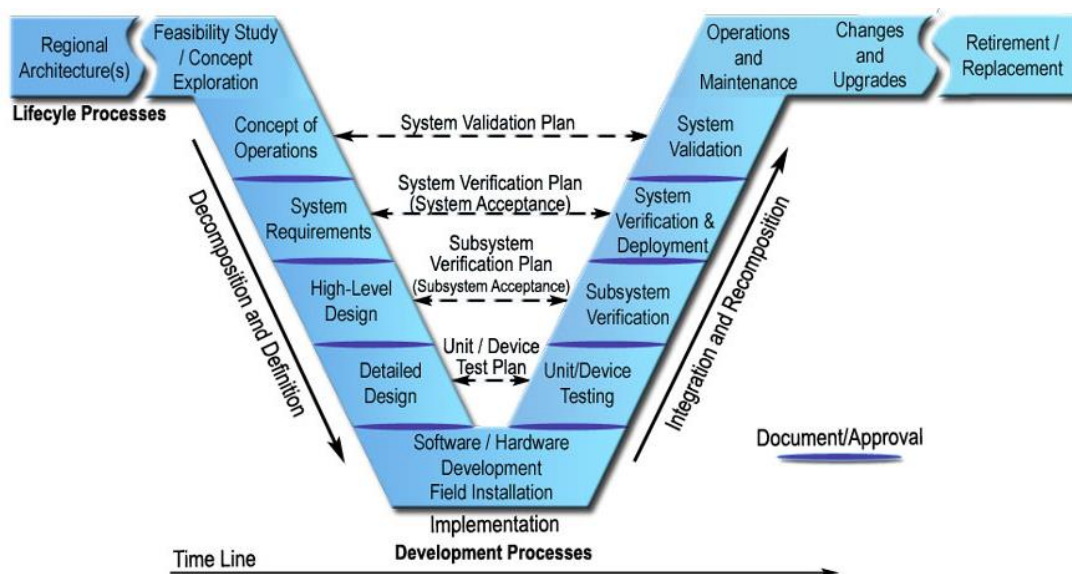


Figura 8 - Cicle de Validació-Verificació del model V

[39]

La forma característica en V, que pren com a nom el model, prové de fet per la forma en que es desenvolupen els processos, validant cada un el procés anterior, però sobretot per l'idea de verificació. És a dir, cada procés en la pota de la dreta verifica el procés del davant.

Procés de **validació**: En primer lloc, el procés de validació fereix una aproximació rigorosa per assegurar que tots els requisits tècnics del producte han sigut coberts, i un primer conceptes del producte. A mesura que anem descendint per la V i, es va profunditzant el nivell de detall, passant d'uns conceptes generals a nivell de sistema a la descripció refinada de component a component de tots els elements que conformen el sistema. A grans trets, la corrent d'especificació consta de: Conceptes d'operacions (què ha de fer el sistema), requisits del sistema i arquitectura del mateix, i disseny detallat.

Procés de **verificació**: Al igual que el procés de verificació, aquest és un procés iteratiu que es replica al llarg de tots els nivells. Mostrarà per un altre part, que el disseny compleix amb tots els requeriments i contribueix a reduir el risc de que el disseny no funcioni de manera apropiada quan el producte físic es verifiqui. Té un paper molt important a l'hora de demostrar la maduresa del producte final. En altres paraules, es realitza una avaluació de la implementació per comprovar que s'han satisfet els requisits. A grans trets consta de: Integració de les diferents parts, prova i verificació, verificació i validació del conjunt del sistema, i manteniment del sistema.

Tal com hem dit, el model V va molt alineat amb els objectius del MBSE, es a dir:

- **Minimitzar els riscos del projecte:** a través de la millora en la transparència i control, permet una detecció prematura de les desviacions en el projecte.
- **Millora i garantia de qualitat:** com un model de procés estàndard, assegura que els resultats que es proporcionen son complets. La uniformitat millora la llegibilitat i comprensió.
- **Reducció dels costs durant tot el projecte i sistema de cicle de vida:** el desenvolupament, producció, operació i manteniment del sistema pot ser calculat i estimat de manera transparent mitjançant la aplicació de models estandarditzats.

Millora de la comunicació: la descripció estandarditzada i uniforme de tots els elements es la base per la comprensió entre tots els *stakeholders*.

3.2.3.2 Mètodes de modelatge

Ara que ja s'ha realitzat una revisió de la literatura i es coneix el context el que es desenvolupa aquest projecte, es dedica un nou capítol al desenvolupament dels pilars de l'enginyeria de sistemes basada en models, que no són més que la metodologia, el llenguatge i la seva eina associada (Figura 9). D'aquesta manera, una vegada conegudes les característiques pròpies de cada mètode, juntament amb els avantatges i els inconvenients de les eines disponibles, es podrà fer una justificació de l'entorn elegit per al desenvolupament d'aquest projecte. A més, s'ha dedicat gran part del capítol a la definició de *SysML*, ja que suposa un dels grans pilars des de l'inici de la definició del MBSE sobre el qual posteriorment s'han desenvolupat diverses metodologies i eines.

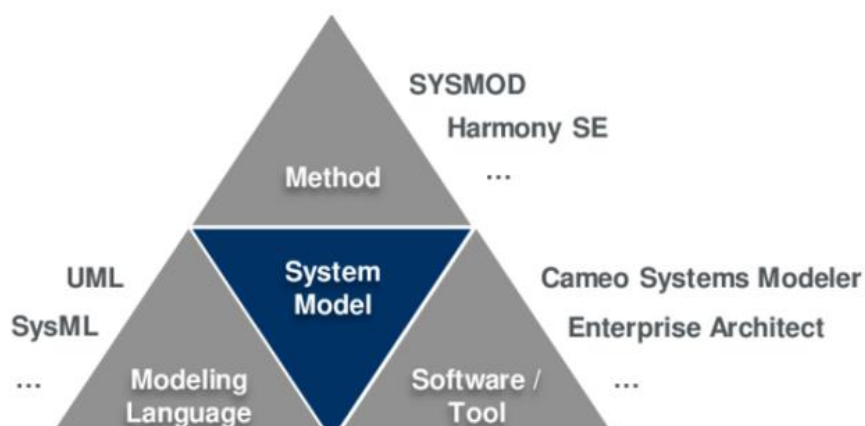


Figura 9 - Pilars del modelatge de l'Enginyeria de Sistemes

Tot i que totes les metodologies comparteixin trets similars, depenent de quin sigui el nostre propòsit, prendrem un camí o altre. Així podríem definir les metodologies del MBSE, com a mapes de ruta. Són una sèrie de tasques de disseny documentades que assegurin que tot l'equip de modelatge treballa cap a una fita comú. Sense aquestes guies hi hauria grans diferències d'amplitud, profunditat i fidelitat, de cada un dels equips davant el model de sistema.

Com tots els projectes, un projecte MBSE requereix un pla. I tots els plans comencen amb un propòsit. L'equip començarà responnent a les preguntes següents: Per què estem fent el model? Més precisament, quins són els resultats esperats? Estem creant un model només per a servir com a registre central d'autoritat per a totes les decisions de disseny? Utilitzarem el model per gestionar la traçabilitat dels requisits i realitzar anàlisis posteriors? Fareu servir el model per realitzar estudis comercials? El model del sistema s'integrarà amb eines dedicades per a la resolució d'equacions i eines de simulació per executar el model?

Les respostes a aquestes preguntes determinen el propòsit de l'esforç de modelatge del vostre equip. Un cop el vostre equip hagi definit aquest propòsit, podeu respondre a nou conjunt de preguntes. Quina part de l'entorn extern del nostre model cal que contingui el sistema proposat? Quins comportaments cal modelar? Com necessitem descompondre les estructures internes i els comportaments? Quins detalls han d'estar al model i quins detalls poden ser omesos?

Les respostes a aquestes preguntes determinen l'abast del model de sistema que el vostre equip ha de construir.

La definició de l'abast estableix l'objectiu cap al que el vostre equip està treballant i permet al vostre equip determinar quan s'ha completat el model. Per ser clar, amb el temps el vostre equip evolucionarà el model segons els canvis en els requisits i es prendran noves decisions de disseny. Si en aquest context es considera complet vol dir que el model compleix el propòsit que heu descrit al pla del projecte. L'abast del model també determina el mètode de modelització que seguirà el nostre equip. Diversos mètodes de modelització estan documentats a la literatura. El nostre equip pot adoptar un d'aquests mètodes existents i adaptar-lo a les nostres necessitats i objectius. O es pot crear un mètode de modelatge personalitzat si cap dels existents s'adapta bé. El focus aquí és ajudar-nos a ser competents en SysML, un llenguatge de modelatge. SysML és independent del mètode; podem utilitzar SysML per crear un sistema model sense importar quin mètode de modelatge decidim que és el més adequat a les nostres necessitats. [36]

Aquí es presenta a continuació una visió general de les principals metodologies actuals relacionades amb el MBSE:

Metodologia MBSE de IBM: Cal destacar l'ús de *Harmony* pel desenvolupament de sistemes integrats i de software. El procés Harmony, desenvolupat per l'empresa IBM, està basat en l'arquitectura dirigida per models. Aquest procés evoluciona per mitjà d'un guió a través d'una sèrie de passos, emprant el llenguatge SysML de manera que permet una transició sense cap problema al desenvolupament posterior del sistema.

Els requisits són analitzats i classificats en funcionals i no funcionals mitjançant models de requisits (no executables), en que es pot observar la seva estructura, i casos d'ús per poder agrupar-los. Aquests casos d'ús generats ofereixen la possibilitat de realitzar un anàlisi funcional del que derivaran diferents escenaris i activitats, i que posteriorment conformaran models executables anomenats models de comportament.

Un cop s'hagin conformat aquests subsistemes, es dissenya una síntesi per trobar el model de l'arquitectura conceptual. Aquest model es verifica davant de la versió del sistema combinat mitjançant l'execució d'ambdós models. Després d'aquest pas el treball de l'enginyer de sistemes haurà acabat per donar pas al desenvolupament de software i hardware, que no forma part del procés *Harmony*.

L'últim pas és verificar tots els components generats durant la fase de desenvolupament de software i hardware davant els requisits. Quan això es realitzi amb èxit, el sistema serà totalment acceptat.

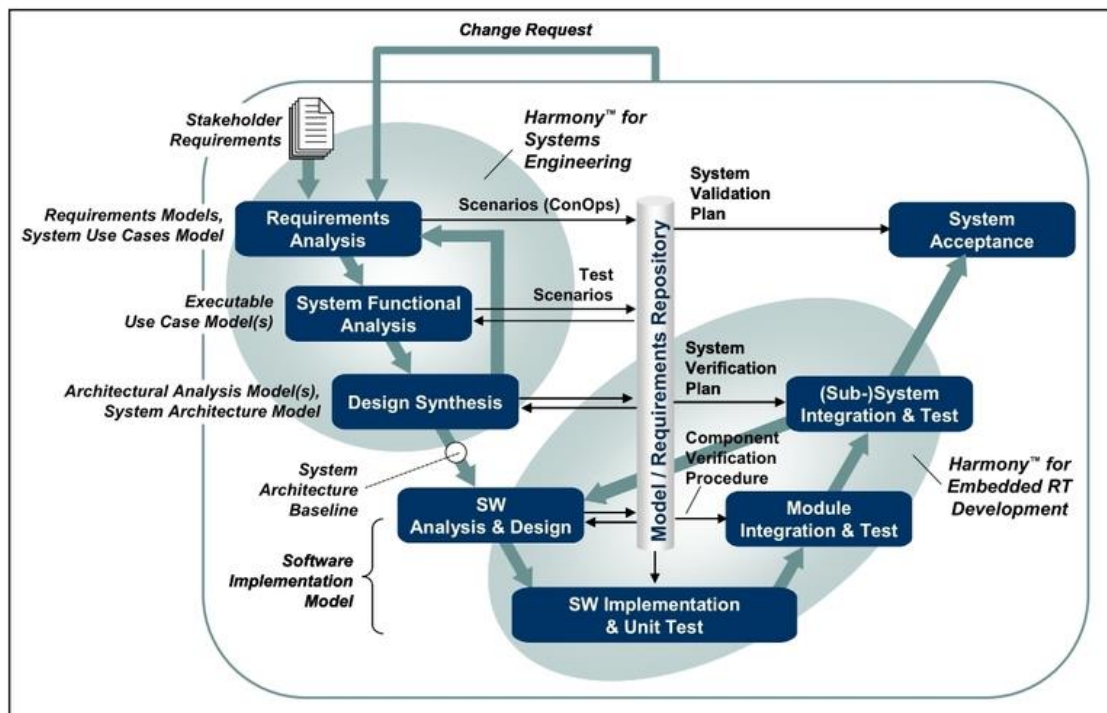


Figura 10 - Procés Harmony

[42]

Object-Oriented Systems Engineering Method (OOSEM): OOSEM és un procés de dalt nivell a baix nivell (*top-down*) basat en escenaris que utilitza SysML per donar suport a l'anàlisi, especificació, disseny i verificació de sistemes. El procés aprofita conceptes orientats a objectes i altres tècniques de modelatge per ajudar a dissenyar sistemes més flexibles i extensibles que puguin adaptar-se a la tecnologia en evolució i requeriments canviants. OOSEM també està pensat per facilitar la integració amb desenvolupament de software orientat a objectes, desenvolupament hardware i processos de prova.

En OOSEM i altres enfocaments d'enginyeria de sistemes basats en models, el model del sistema és el principal fruit del procés d'especificació i disseny del sistema. El mètode inclou una o més iteracions de les següents activitats per especificar i dissenyar el sistema:

- **Organitzar el model**
 - Definir el diagrama de paquets per al model del sistema.
- **Analitzar les necessitats dels grups d'interès** per entendre el problema a resoldre, els objectius als que el sistema pretén donar suport, i les mesures d'eficàcia necessàries per avaluar com de bé el sistema dona suport al objectiu.
 - Identificar les parts interessades i els problemes a abordar

- Definir el model de domini per identificar el sistema i els sistemes i usuaris externs
- Definir els casos d'ús de nivell superior per representar els objectius que el sistema pretén donar suport
- Definir les mesures d'eficàcia que es poden utilitzar per quantificar el valor d'una solució proposada
- **Especificar els requisits del sistema**, incloent-hi la funcionalitat del sistema, les interfícies, les interfícies físiques i característiques de rendiment i altres característiques de qualitat per donar suport als objectius i mesures d'eficàcia
 - Capturar els requisits basats en text en un diagrama de requisits que donen suport als objectius del sistema i mesures d'eficàcia
 - Modelar cada escenari d'ús (per exemple, diagrama d'activitats) per especificar els requisits de comportament del sistema
 - Crear el diagrama de context del sistema (diagrama de blocs intern) per especificar les interfícies externes del sistema
- **Sintetitzar solucions alternatives del sistema** subdividint el disseny del sistema en components que puguin satisfer els requisits del sistema
 - Descompondre el sistema mitjançant el diagrama de definició de blocs
 - Definir la interacció entre les parts mitjançant diagrames d'activitats
 - Definir la interconnexió entre les parts mitjançant el diagrama de blocs interns
- **Realitzar una anàlisi de terme mig** per avaluar i seleccionar una solució preferida que satisfaci els requisits de sistema i maximitzi el valor en funció de les mesures d'eficàcia
 - Capturar el context de l'anàlisi per identificar l'anàlisi a realitzar com ara el rendiment, fiabilitat, cost i altres propietats crítiques
 - Capturar cada anàlisi com un diagrama paramètric
 - Realitzar l'anàlisi d'enginyeria per determinar els valors de les propietats del sistema (Nota: l'anàlisi es realitza en eines d'anàlisi d'enginyeria)
- **Mantenir la traçabilitat dels requisits** per garantir que la solució proposada satisfà els requisits de sistema i necessitats de les parts interessades associades
 - Capturar la traçabilitat entre els requisits del sistema i les necessitats de les parts interessades
 - Mostrar com el disseny del sistema compleix els requisits del sistema

- Identificar casos de prova necessaris per verificar els requisits del sistema i capturar els resultats de la verificació

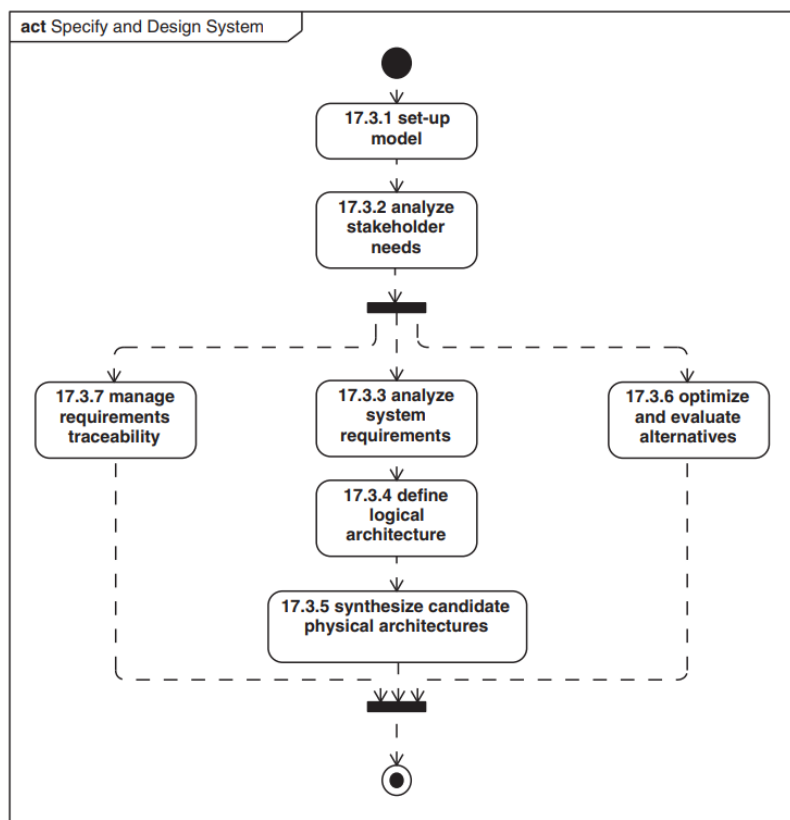


Figura 11 - Procés de disseny i especificació del sistema

(El números d'acció es refereixen a sub-seccions on estan descrits, provinents de la font)

[40]

Altres activitats de gestió d'enginyeria de sistemes, com ara la planificació, l'avaluació, la gestió de riscos i la gestió de la configuració, es realitzen conjuntament amb les activitats de modelització descrites anteriorment.

Arcadia: La majoria de processos de desenvolupament del cicle de vida d'un sistema estaven molt més centrats en la definició de requisits, com a la seva traçabilitat amb els diferents components. Tot i així, actualment se li ha donat més valor a l'anàlisi funcional per aconseguir un disseny de l'arquitectura del sistema més determinat des del principi del projecte. S'adopten un nombre molt major de punts de vista del sistema, afegint una percepció més íntegra de l'estructura i descomposició del sistema d'interès. Partint d'aquesta idea principal, l'empresa Thales va dissenyar una metodologia per desenvolupar activitats

d'enginyeria centrades en models que representaran l'arquitectura del sistema i el seu funcionament, denominada ARCADIA (*ARChitecture Analysis and Design Integrated Approach*).

Arcadia [43] és una metodologia tan sols enfocada a sistemes, i a l'arquitectura tant de el seu software com el seu hardware. La seva utilitat engloba des de comprendre les necessitats del client i definir l'arquitectura del sistema entre els *stakeholders*, fins a validar el disseny del sistema i justificar i facilitar les tasques d'integració, verificació i qualificació. D'aquesta forma, millora l'eficiència i la qualitat de l'enginyeria de sistemes, la millora de la complexitat del producte, redueix els costos de desenvolupament i el temps de disseny i promou el treball col·laborador entre els enginyers que tenen influència sobre el sistema.

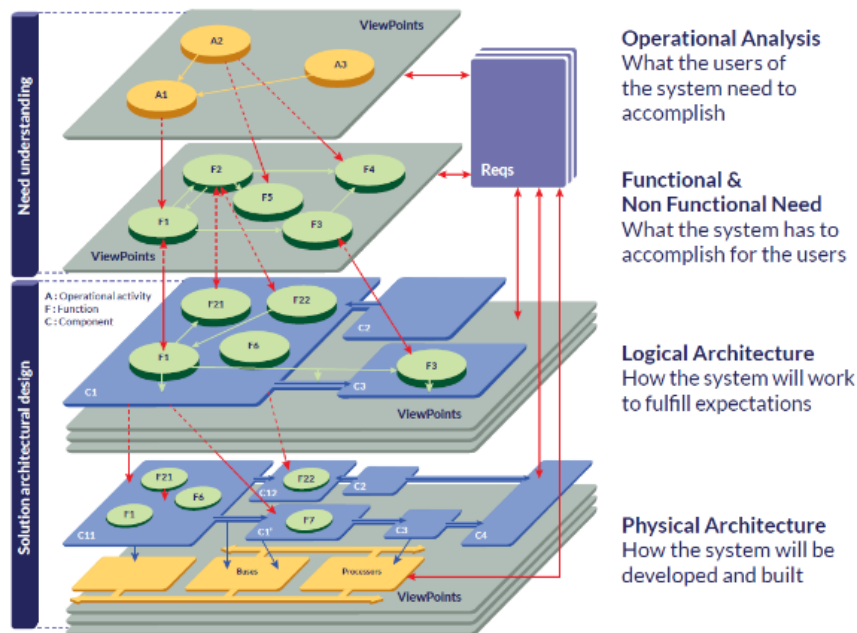


Figura 12 - Metodologia ARCADIA

[38]

3.2.4 System Modeling Language

El *System Modeling Language* o *SysML* és el llenguatge de modelat utilitzat en aquest projecte per descriure les diferents vistes del sistema.

Quan creem un model, estem parlant un llenguatge. No és un idioma comú sinó que és un llenguatge de modelat: un llenguatge semi informal que defineix els tipus d'elements que està permès usar en el teu model, les relacions permeses entre ells, i el tipus de notació per representar els dos anteriors. [36]

Els usuaris de MBSE comunament usen el SysML per construir models per l'estructura del sistema, el seu comportament, els requeriments, i les limitacions.

La gramàtica i notació del SysML estan definits per unes especificacions d'estàndards propietat del *Object Management Group, Inc. (OMG)*, que és un consorci de centenars de d'empreses del sector tecnològic. Tot i així, no és un llenguatge format per si mateix, podríem definir-lo millor com un perfil-una extensió-d'un subconjunt del *Unified Modeling Language (UML)*. Per tant, una gran part dels seus diagrames, per que sí, la base del llenguatge son els diagrames, la comparteix amb el UML.

El SysML es basa en un conjunt de diagrames diferents que utilitzarem per representar les diferents vistes d'un sistema.

La següent figura mostra tot els diagrames que comprèn el SysML.

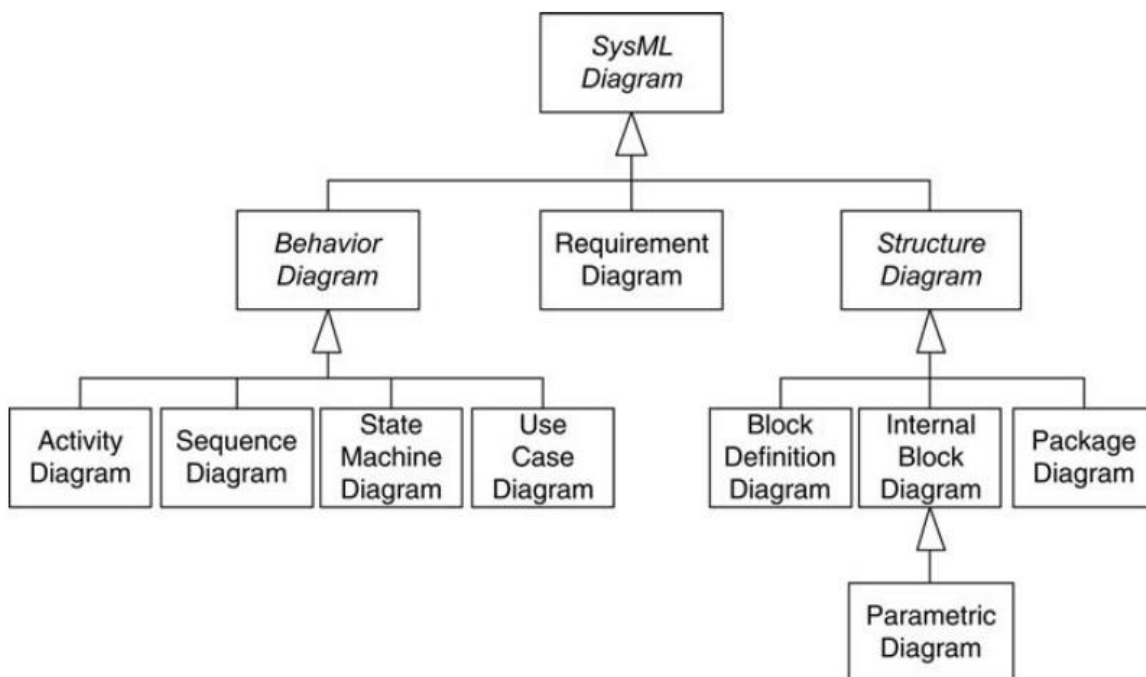


Figura 13 - Diagrames SysML

[36]

En aquest projecte hi ha alguns diagrames que no farem servir. La utilització dels diferents diagrames està lligat a les necessitats del usuari. En funció del objectiu, ens convindrà més un o altre diagrama. En el nostre cas podem descartar la utilització dels diagrames "Parametric Diagram", "Package Diagram", "Internal Block Diagram" i el "Sequence Diagram". Explicarem ara la resta de diagrames que realitzarem, entre de comportament i estructura.

3.2.4.1 Behaviour Diagrams

3.2.4.1.1 Use case diagram

El *Use Case Diagram* (UC) descriuen les accions necessàries que s'han de dur a compte pel sistema per aconseguir un cert objectiu. També descriu les interaccions entre el sistema i els seus actors (qui o què interactua en el sistema). Aquí s'exposa un exemple:

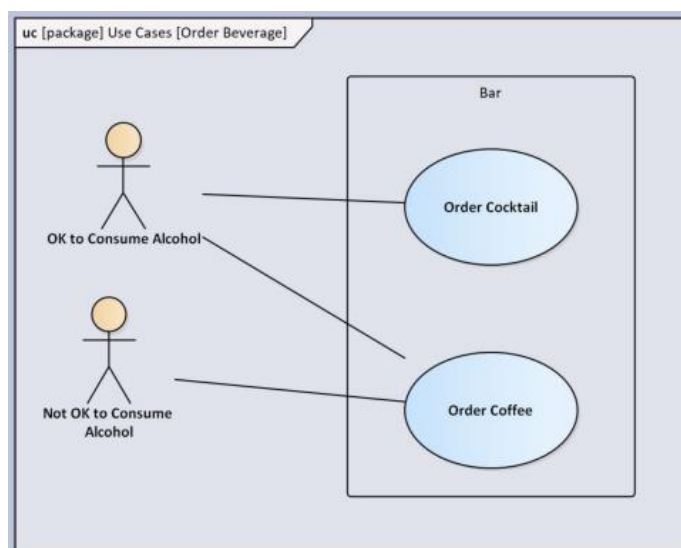


Figura 14 - Use Case Diagram

[37]

Aquest és el primer diagrama fet en el procés i, amb aquest comença l'anàlisi funcional. És important no oblidar-nos de cap cas d'ús per modelar correctament el sistema.

3.2.4.1.2 Activity Diagram

El *Activity Diagram* (ACT) és una representació gràfica del flux de treball fet per simples accions necessàries per aconseguir cada cas d'ús.

Aquest diagrama dona la representació de comportament del sistema i divideix les diferents activitats entre tots els elements de sistema, per això és possible si totes les activitats estan cobertes o no.

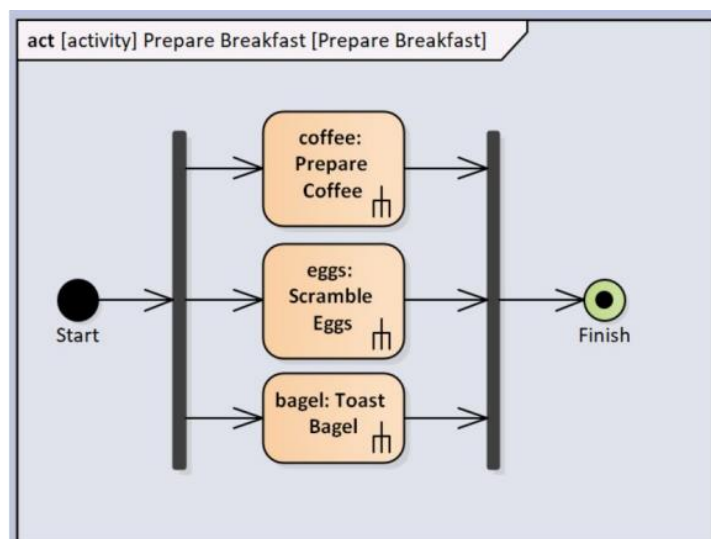


Figura 15 - Activity Diagram

[37]

En el costat esquerra es possible trobar el punt de partida amb la senyal d'entrada pel diagrama. Al mig hi ha les activitats requerides per la funcionalitat de cada cas d'ús que descriu, i a la part alta hi ha diferents elements de sistema. Amb aquest diagrama és fàcil veure, amb les diferents entrades (*input signals*), què s'ha de fer (*activities*) i qui ho ha de fer (*system elements*).

3.2.4.1.3 Sequence Diagram

Com el diagrama d'activitat, el *sequence diagram* presenta una vista dinàmica del sistema, una vista que expressa seqüències de comportaments i esdeveniments al llarg del temps. És una bona idea confeccionar un diagrama de seqüència quan volem ficar el focus en com interactuen les diferents parts de cada bloc a través de trucades i senyals asíncrones que produeixen comportaments emergents. Aquest comportament és conegut com interacció.

La llegibilitat del diagrama, tot així, esdevé més i més costosa com més complex sigui el control lògic. Per aquesta raó els modeladors solen utilitzar el diagrama de seqüència com un cas de test (*test case*) que mostra només un únic escenari d'un comportament d'ús més extens o complex.

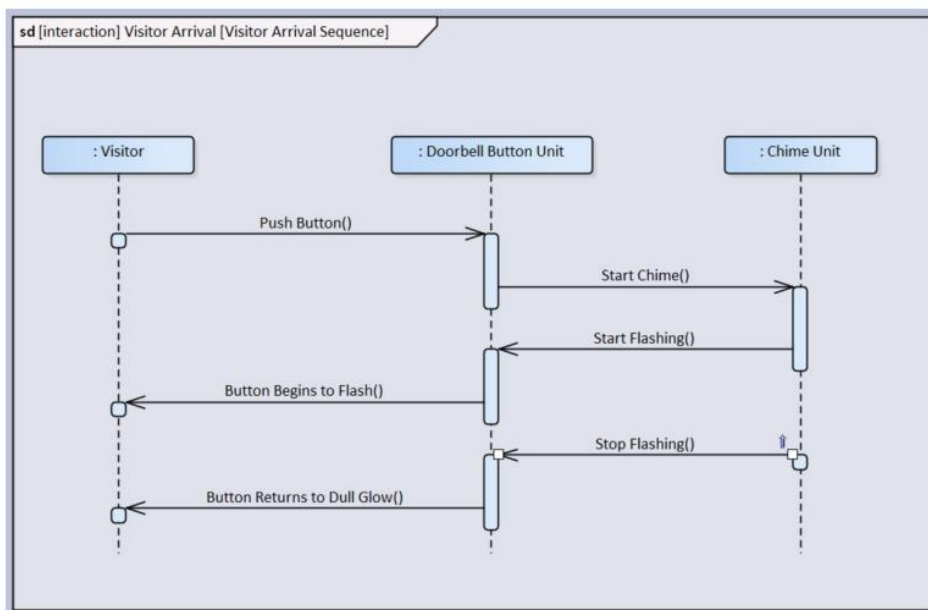


Figura 16 - Sequence Diagram

[37]

3.2.4.1.4 State Machine Diagram

El *State Machine Diagram* mostra el comportament d'una entitat mentre va canviant d'estat. En el cas del SysML, aquesta entitat es tracta normalment d'un bloc. Amb diferència al diagrama d'activitat, la màquina d'estats fica el focus en com evoluciona una estructura dintre el sistema al llarg del temps. També es pot fer servir com un classificadors dels comportaments dels diferents blocs.

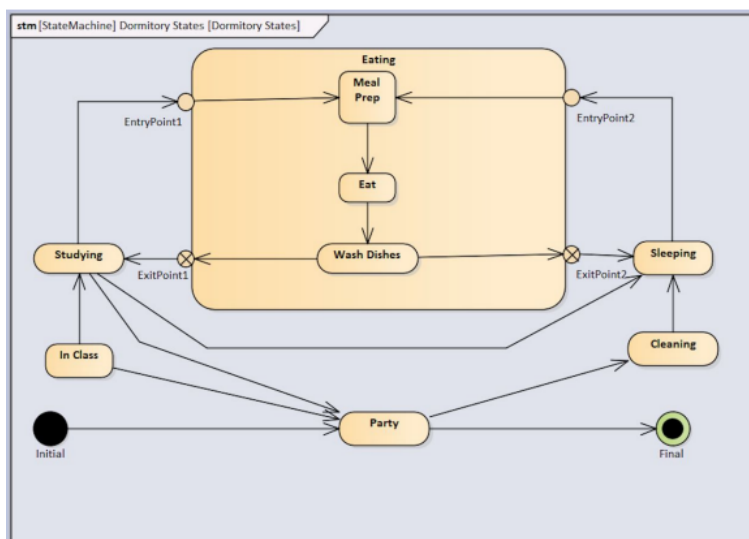


Figura 17 - State Machine Diagram

[37]

3.2.4.2 Structure Diagrams

3.2.4.2.1 Block Definition Diagram

És el tipus de diagrama més comú. Es poden representar gran varietat de models i relacions amb un BDD per expressar informació de l'estructura del sistema. Els tipus d'elements que es mostren en els *BDDs*, com ara blocs, actors, blocs de limitacions, interfícies, etc. Serveixen després com a tipus pels altres elements que apareixen en els altres diagrames. Ens referim als elements que surten en els *BDDs* com a **elements de definició**.

La idea d'incorporar un diagrama BDD és sovint bona idea ja que no esta lligat a cap estat del cicle de vida o nivell de disseny. Es pot influir en l'anàlisi dels requeriments de *stakeholders*, en la definició dels requeriments, en el disseny de l'arquitectura (el nostre cas), en l'anàlisi del rendiment, desenvolupament de *test cases* (casos de test), i integració. També es poden crear BDD amb combinació amb altres diagrames SysML per donar una visió complementària d'un aspecte del sistema d'interès.

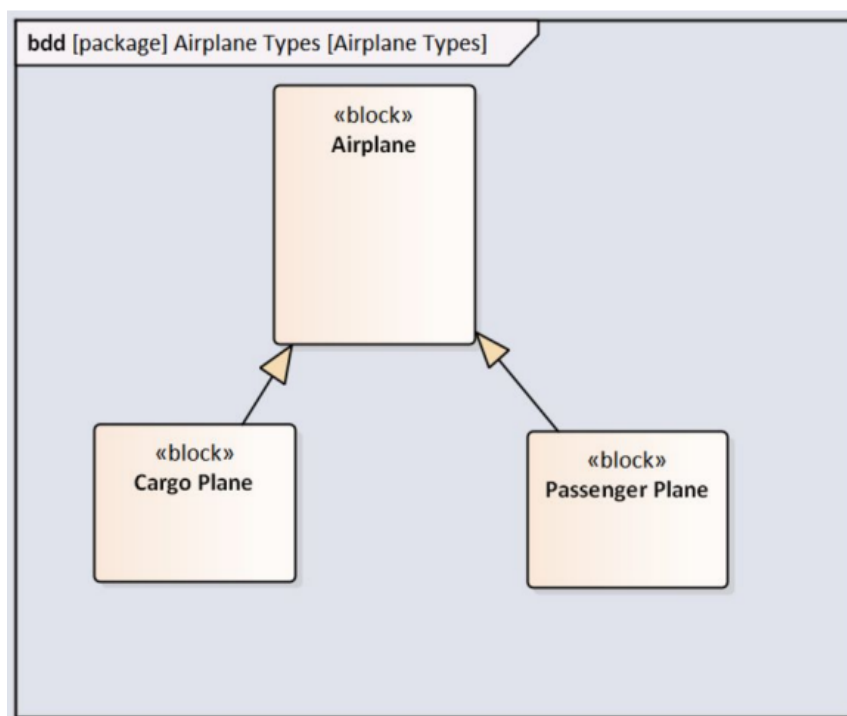


Figura 18 - *Block Definition Diagram*

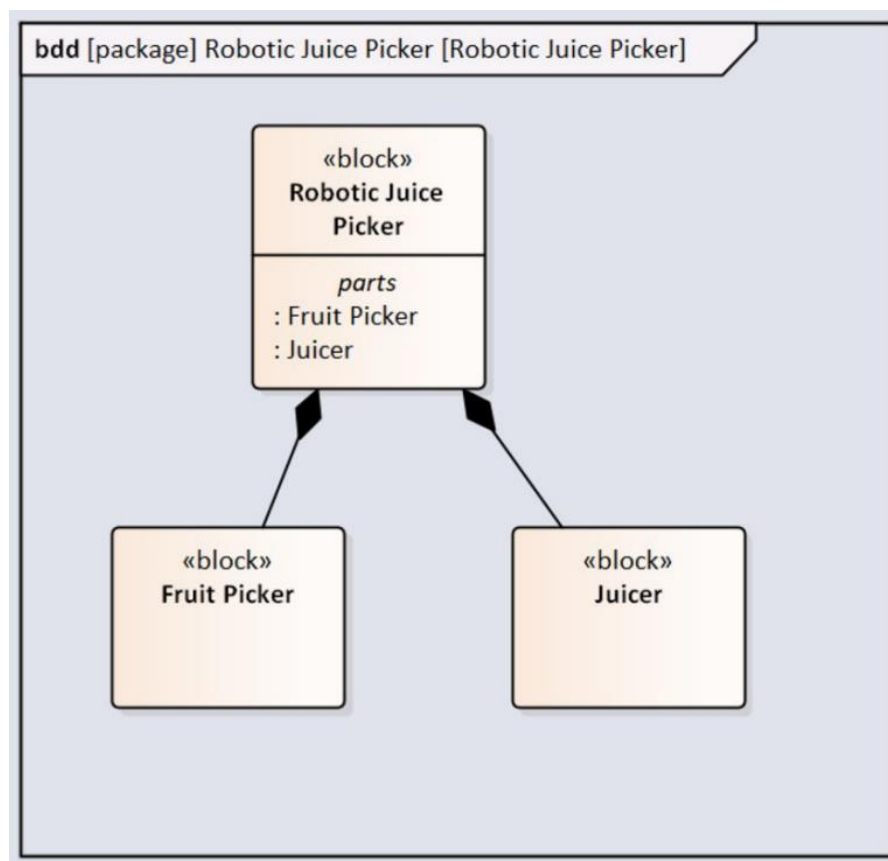
[37]

3.2.4.2.2 Internal Block Diagram

El *Internal Block Diagram* (IBD) té una relació propera amb el BDD. Té sentit mostrar alguns elements en un IBD per expressar aspectes de l'estructura de sistema que complementa els aspectes transmesos en els BDDs. Òbviament els IBDs també tenen capacitats úniques que el fa essencial afegir-lo al modelat del sistema.

Es sol crear un IBD per especificar l'estructura interna d'un sol bloc. Com el BDD, es una vista estàtica (estructural) del sistema o una de les seves parts, però a diferència, no mostra simples blocs, sinó els seus usos. Diguéssim que fem el focus en les parts que el componen i les seves referències. Eren característiques que el BDD ens permetia però d'una manera que aquestes parts i referències es situaven a mode de petits compartiments en cada bloc, però un IBD et permet expressar informació addicional que no podem transmetre solament amb el BDD: les connexions entre les diferents parts i referències; el tipus de matèria, energia, o dades que circulen per les connexions; i els serveis que s'estipulen i es requereixen.

Això es una gran capacitat. És important, tot i així, ser conscient de les limitacions del SysML. El SysML no ofereix medis per modelar la geometria del sistema. Un IBD permet modelar quines parts han de ser connectades entre elles, però no modela la forma del espai que conformen. Per fer-ho ens hauríem de recolzar en eines de disseny (CAD).

Figura 19 - *Internal Block Diagram*

[37]

3.2.4.2.3 Package Diagram

El *Package Diagram* es un tipus de diagrama que creem quan necessitem mostrar com està organitzat el nostre model. Aquesta organització s'extrau de la jerarquia de "packs" que creem per partir els elements del model en grups cohesionats de forma lògica.

Per això, un cop tenim clara una idea efectiva del model d'estructura que volem seguir, formular un *Package Diagram* és útil per proveir els *stakeholders* una visió senzilla del sistema.

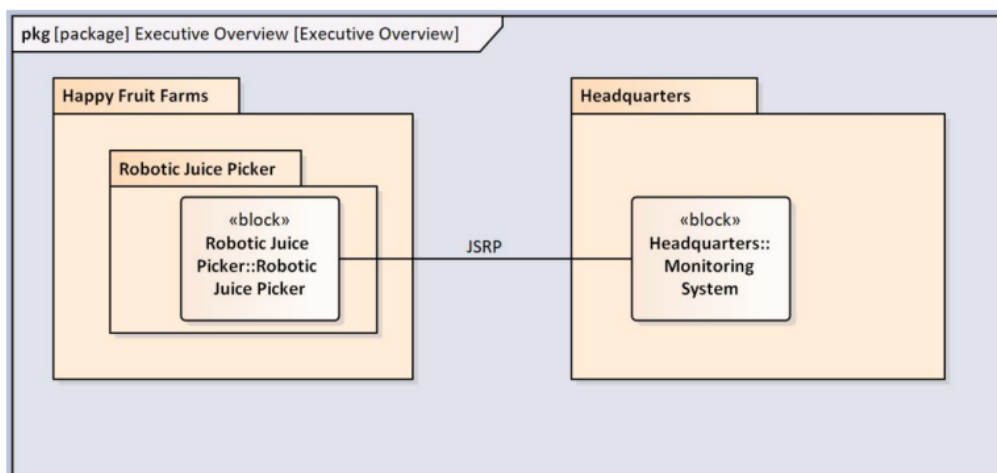


Figura 20 - Package Diagram

[37]

3.2.5 Polarion

Polarion és un solució unificada d'*Application Lifecycle Management (ALM)*, en la que es pot definir, crear, provar i gestionar sistemes de software complexes completament basada en Web que serveix tant per equips petits com per mils d'usuaris.

Serà el software amb el que gestionarem els requisits i els casos d'us del nostre projecte, aprofitant la feina que han realitzat anteriorment els nostres companys.



Figura 21 - Polarion

3.2.6 Enterprise Architect

Sparx Enterprise Architect (Sparx EA) representa una de les eines MBSE que compleix l'estàndard OMG SysML i és relativament fàcil d'usar [29]. Sparx EA permet activitats MBSE essencials, com la traçabilitat de requisits, la simulació de diagrames de comportament (activitat, màquina d'estat, seqüència) per a l'anàlisi i el disseny, la simulació de diagrames paramètrics per a estudis comercials i la generació de documentació automatitzada. A més, Sparx EA ofereix *plugins* propis i comercials per eines de gestió de requisits (com és el nostre cas amb Polarion) i eines de simulació (MATLAB/Simulink). Sparx EA també s'integra bé amb

els estàndards de codi obert per al modelatge en equip i les simulacions de diagrames paramètrics (Open Modelica).

Els punts forts a destacar de Sparx EA per a MBSE inclouen: diagrames SysML relativament fàcils de dibuixar, suport d'estàndards oberts per al modelatge en equip i simulacions de diagrames paramètrics, a un preu relativament baix.

Els defectes de Sparx EA per a MBSE inclouen, entre d'altres, una interfície d'usuari força complexa, compliment irregular de la sintaxi i semàntica d'OMG SysML, i mecanismes d'extensió encara immadurs (perfils, scripts).

Així, Sparx EA per a MBSE es podria dir que destaca des del punt de vista de la seva compatibilitat amb SysML, el seu conjunt de característiques MBSE i el seu preu relativament baix, havent estat designada com una "Best Value" entre les eines de modelatge MBSE de la competència.

Serà la eina amb la que modelem tots els nostres sistemes, mitjançant el ús de diagrames en SysML. Ens ajudarem deL seu complement *Pro Cloud Server* que ens servirà per integrar-ho amb el Polarion.



Figura 22 - Sparx EA i PCS

3.2.7 Meta-modelat

A part de contar amb tres pilars fonamentals com metodologia, llenguatge i eina, el nivell més ampli del model s'aconsegueix quan es defineixen els conceptes i les relacions entre conceptes propis del model en qüestió. Aquest conceptes es denomina meta-modelat [45]. Ja que un dels punts forts del modelat de sistemes es la capacitat de compartir informació entre els diferents *stakeholders* que intervenen d'alguna forma amb el sistema d'interès, sorgeix la necessitat que tots ells tinguin la mateixa noció del que es pretén transmetre a través de qualsevol element o les connexions establertes entre aquests. Per tant, un meta-model es la especificació explícita de l'abstracció que suposa un model, es a dir, la representació d'un model.

Un dels objectius del MBSE és fer servir models per guiar el desenvolupament d'un sistema. Anàlogament, per tenir un marc de referència amb el que crear aquests models necessitem un model superior que descriu la resta de models elaborats. Es tracta per tant d'un nivell més alt d'abstracció, però sense informació ni contingut tangible.

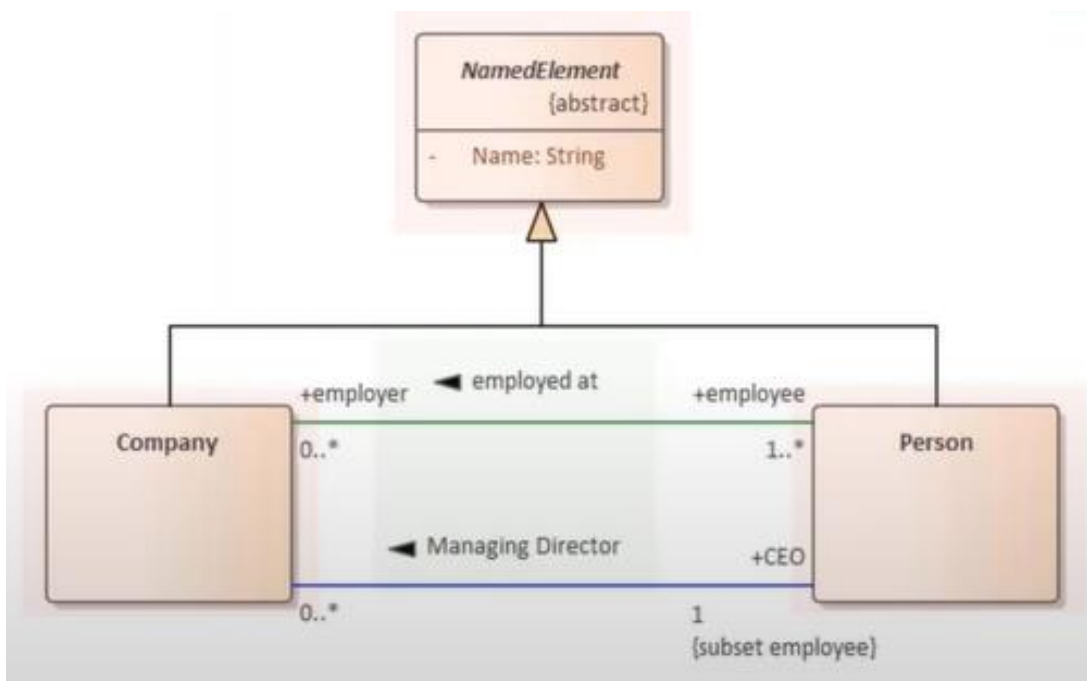


Figura 23 - Exemple Meta-model

Sense un meta-modelat, tots els models del sistema son simplement vistes desconegudes del sistema, un conjunt de diagrames que no aporten gaire valor afegit pel disseny. Amb aquest model de models, és possible completar la informació, per evitar múltiples interpretacions i enriquir el sentit de tots els esquemes. En la imatge anterior () es pot observar que el model del sistema seria un conjunt d'entitats amb atributs i relacions entre sí, però sense l'estructura apropiada per poder descriure'ls apropiadament.

En l'àmbit de l'automoció, és una eina molt efectiva per a que, en la fase pre-contractual, totes les necessitats del client traduïdes en requisits quedin recollides, de manera que totes les parts entenguin el mateix per cada element. El meta-model pretén ser un instrument de comunicació amb la intenció de substituir els documents en els que es concreten els aspectes relatius al sistema.

Per visualitzar aquest concepte des de una altra perspectiva, es possible comparar-ho amb les regles sintàctiques d'un llenguatge. Amb lletres es poden construir paraules, però

sense unes normes pre-determinades les paraules manquen de sentit. Es precisa d'alguns tipus de meta-llenguatge que expliqui l'estructura de les oracions, i que el seu sentit es completi quan tinguin subjecte, verb i predicat.

La plenitud del model d'un sistema, en definitiva, requereix de l'especificació de les idees que el componen i de la relació que hi ha entre elles. D'aquesta manera es torna consistent i per això l'enginyer de sistemes adquireix el poder de construir una forma de modelar apropiada per un model en concret.

3.2.8 Automotive SPICE

Cal, en aquest moment parlar del life cycle, ja que és el concepte sobre el que gira tot el ASPICE. Podem definir el *Life Cycle*, o com cicle vida, a totes les fases per les que arriba a passar un producte, donant-li la dotació, en el seu propi nom, de cicle. Perquè un producte no acaba quan s'ha definit, sinó que es reutilitza o revisa en moltes de les seves etapes de procés o posterior, com integració, validació etc. Aquest concepte està totalment relacionat amb el model V.

La indústria de l'automòbil ha crescut molt i ha notat un gran canvi, tal que en les últimes dècades, sistemes que fa anys eres només un simple somni, avui en dia estan implementats com un estàndard en els automòbils. Sistemes com el *Antilock Brake Systems (ABS)*, la injecció electrònica de combustible, suspensió automàtica, fars intel·ligents, etc. A més, el món de l'automòbil és un món competitiu, un món on un petit problema pot costar milions d'euros, o inclús pitjor, la reputació d'una empresa. Per aquest motiu, els fabricants de cotxes tenien un problema de confiança amb els seus proveïdors perquè no hi havia una manera de "quantificar" la qualitat dels productes que estaven comprant. Per això al 1993 la *Automotive Special Interest Group (SIG)*, format per un grup de companyies d'automoció alemanyes, va fundar el que coneixem actualment com *Automotive SPICE* [39].

ASPICE (Automotive Software Process Improvement Capability dEtermination) es una directriu estàndard [35] del sector per avaluar els processos del desenvolupament de software. Introduït al 2005, ASPICE ajuda als proveïdors automotrius a incorporar les millors pràctiques per identificar els defectes en una fase més prematura del desenvolupament i garantir el compliment dels requisits dels *Original Equipment Manufacturer (OEM)s*.

Es tracta d'un estàndard essencial per companyies que busquin desenvolupar software d'altíssima qualitat, ja que ajuda a identificar i eliminar ineficiències, millorar la comunicació, i reduir els costos. El marc d'actuació d'ASPICE cobreix el desenvolupament complet del cicle de vida, des de la gestió dels requeriments fins la fase de testeig i manteniment del software, i ajuda a les institucions a assegurar-se que el seu desenvolupament és efectiu i eficient.

ASPICE és una adaptació específica de la norma ISO 33061 [38], que s'ha utilitzat per una varietat d'indústries durant dècades per perfeccionar el desenvolupament software. ASPICE aborda necessitats específiques de la indústria de l'automòbil pel que la seva predecessora no va ser dissenyada, influint un major èmfasi en la ciberseguretat.

3.2.8.1 Focus

Process assessment (procés d'avaluació) és una avaluació disciplinada de dels processos al voltant d'una unitat organitzativa comparada amb un *process assessment model* (model d'avaluació del procés). La intenció del PAM és dur a terme una avaluació en conformitat amb un model, de les capacitats d'un procés. Aquest model de referència serà el *process reference model* (PRM).

Aquest model d'avaluació de procés incorpora una sèrie d'indicadors que es tenen compte a l'hora d'interpretar el propòsit de seguir el PRM.

3.2.8.2 Determinació de la capacitat del procés

El concepte de determinació de la capacitat del procés es basa en l'ús d'un model d'avaluació del procés en un marc bidimensional. La primera dimensió la proporcionen els processos definits en un procés model de referència (dimensió del procés). La segona dimensió consisteix en els nivells de capacitat que són subdividit, a més, en atributs de procés (dimensió de capacitat). Els atributs del procés proporcionen les característiques mesurables de la capacitat del procés. El model d'avaluació de processos selecciona processos d'un model de referència de processos i el complementa amb indicadors. Aquests indicadors donen suport a la recollida d'evidències objectives que permeten una avaluador per assignar qualificacions pels processos segons la seva dimensió de capacitat.

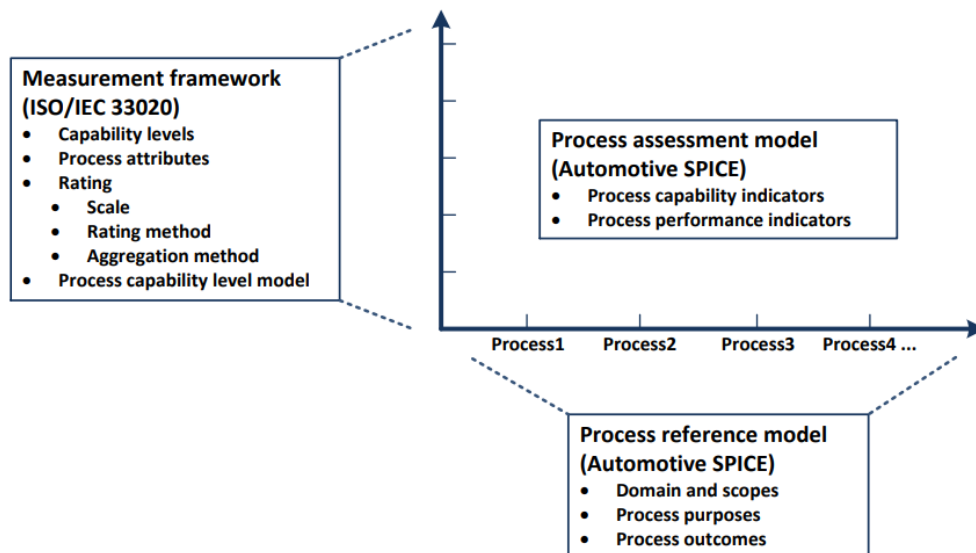


Figura 24 - Relació del PAM

[30]

3.2.8.3 Process reference model – PAM

El processos s'agrupen en categories de processos i en un segon nivell dependent de a quin tipus d'activitat s'adrecen.

Hi ha 3 categories: processos en fase prematura del cicle de vida, processos organitzatius del cicle de vida i processos de suport del cicle de vida. A continuació veurem com es desglossen les categories:

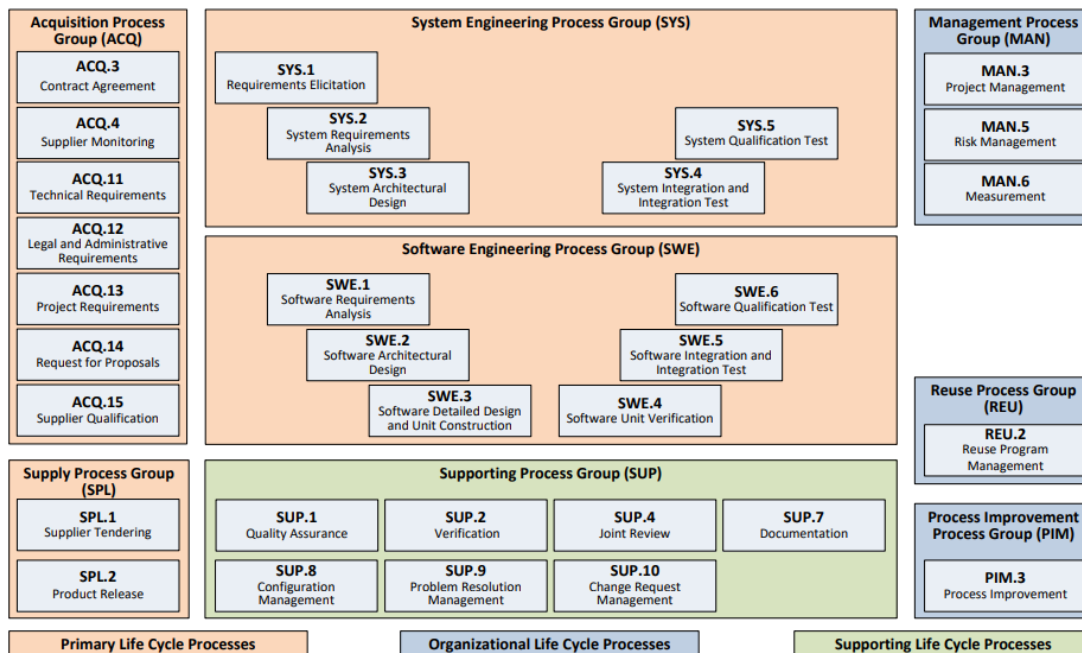


Figura 25 - Resum del PRM d'Automotive SPICE

[30]

En el nostre cas, ens trobem en la *Primary Life Cycle Processes*. Dintre de la família de *System Engineering Process Group (SYS)*.

El grup SYS consisteix en processos adreçats a l'obtenció i la gestió dels requeriments de interns i del client, la definició de l'arquitectura de sistema i la integració i testeig a nivell de sistema.

SYS.1	Requeriments Elicitation
SYS.2	System Requeriments Analysis
SYS.3	System Architectural Design
SYS.4	System Integration and Integration Test
SYS.5	System Qualification Test

Taula 1— Primary life cycle processes – SYS process group

[30]

Aquest projecte ficarà el focus en crear la part de *System Architectural Design*. Per tal de fer-ho correctament seguirem la taula específica pel procés, que comprèn els "productes" que s'esperen de cada *base practices* (pràctiques establertes).

Process ID	SYS.3
Process name	System Architectural Design
Process purpose	The purpose of the System Architectural Design Process is to establish a system architectural design and identify which system requirements are to be allocated to which elements of the system, and to evaluate the system architectural design against defined criteria.
Process outcomes	As a result of successful implementation of this process: <ol style="list-style-type: none">1) a system architectural design is defined that identifies the elements of the system;2) the system requirements are allocated to the elements of the system;3) the interfaces of each system element are defined;4) the dynamic behavior of the system elements is defined;5) consistency and bidirectional traceability are established between system requirements and system architectural design; and6) the system architectural design is agreed and communicated to all affected parties.
Base practices	SYS.3.BP1: Develop system architectural design. Develop and document the system architectural design that specifies the elements of the

	<p>system with respect to functional and non-functional system requirements. [OUTCOME 1] <i>NOTE 1: The development of system architectural design typically includes the decomposition into elements across appropriate hierarchical levels.</i></p> <p>SYS.3.BP2: Allocate system requirements. Allocate the system requirements to the elements of the system architectural design. [OUTCOME 2]</p> <p>SYS.3.BP3: Define interfaces of system elements. Identify, develop and document the interfaces of each system element. [OUTCOME 3]</p> <p>SYS.3.BP4: Describe dynamic behavior. Evaluate and document the dynamic behavior of the interaction between system elements. [OUTCOME 4] <i>NOTE 2: Dynamic behavior is determined by operating modes (e.g. start-up, shutdown, normal mode, calibration, diagnosis, etc.).</i></p> <p>SYS.3.BP5: Evaluate alternative system architectures. Define evaluation criteria for the architecture. Evaluate alternative system architectures according to the defined criteria. Record the rationale for the chosen system architecture. [OUTCOME 1] <i>NOTE 3: Evaluation criteria may include quality characteristics (modularity, maintainability, expandability, scalability, reliability, security realization and usability) and results of make-buy-reuse analysis.</i></p> <p>SYS.3.BP6: Establish bidirectional traceability. Establish bidirectional traceability between system requirements and elements of the system architectural design. [OUTCOME 5] <i>NOTE 4: Bidirectional traceability covers allocation of system requirements to the elements of the system architectural design.</i> <i>NOTE 5: Bidirectional traceability supports coverage, consistency and impact analysis.</i></p> <p>SYS.3.BP7: Ensure consistency. Ensure consistency between system requirements and the system architectural design. [OUTCOME 1, 2, 5, 6] <i>NOTE 6: Consistency is supported by bidirectional traceability and can be demonstrated by review records.</i> <i>NOTE 7: System requirements typically include system architectural requirements. Refer to BP5.</i></p> <p>SYS.3.BP8: Communicate agreed system architectural design. Communicate the agreed system architectural design and updates to system architectural design to all relevant parties. [OUTCOME 6]</p>
<p>Output work products</p>	<p>04-06 System architectural design → [OUTCOME 1, 2, 3, 4, 5] 13-04 Communication record → [OUTCOME 6] 13-19 Review record → [OUTCOME 5] 13-22 Traceability record → [OUTCOME 5] 17-08 Interface requirements specification → [OUTCOME 3]</p>

Taula 2 - Resum procés SYS.3

[30]

3.2.8.4 Producte final: *System architectural design*

En aquesta secció s’analitzarà els diferents punts del producte final (*work product*) “*System architectural design*” descrit en ASPICE per veure quints cobrirem en el projecte.

04-06	System architectural design	<ul style="list-style-type: none"> • Provides an overview of all system design • Describes the interrelationship between system elements • Describes the relationship between the system elements and the software • Specifies the design for each required system element, consideration is given to aspects such as: <ul style="list-style-type: none"> - memory/capacity requirements - hardware interface requirements - user interface requirements - external system interface requirements - performance requirements - command structures - security/data protection characteristics - settings for system parameters (such as application parameters or global variables) - manual operations - reusable components • Mapping of requirements to system elements • Description of the operation modes of the system components (startup, shutdown, sleep mode, diagnosis mode, etc.) • Description of the dependencies among the system components regarding the operation modes • Description of the dynamic behavior of the system and the system components
-------	-----------------------------	--

Taula 3 - Característiques del "System architectural design"

[30]

Si ens fixem en la taula proporcionada per la guia (Taula 3) podem veure com tots els productes són fruit o estan molt lligats a què hauríem d'extreure del procés SYS.3. D'aquí podem extreure tots els punts que realment creiem que siguin d'interès per la construcció posterior del sistema.

A continuació afegirem la nostra taula amb l'equivalència que hem decidit que construirem per assolir la majoria de punts del SYS.3, tot i que potser necessitarem més d'un diagrama per cada un dels productes, i tots els productes que extraurem d'acord a cada punt.

WP ID: AEB		
WP Name: Disseny d'arquitectura de sistema		
1.	Dona una visió general del disseny de sistema	Diagrama de blocs que juntament amb el diagrama de paquets
2.	Describeix la relació entre els elements de sistema	El IBD mostra la relació entre els elements de sistema mitjançant ports
3.	Describeix la relació entre els elements de sistema i el software	Enllaçarem els elements de sistema i de software en un diagrama de blocs

4.	Especifica el disseny per a cada element del sistema requerit, es tenen en compte aspectes com ara:	
4.1	Requeriments de memòria/capacitat	No és la intenció del projecte mostrar aquesta informació
4.2	Requeriments d'interfície de hardware	Els requeriments d'interfície de hardware estaran representats en el diagrama IBD.
4.3	Requeriments d'interfície d'usuari	No és la intenció del projecte mostrar aquesta informació
4.4	Requisits d'interfície del sistema extern	No és la intenció del projecte mostrar aquesta informació
4.5	Requeriments de rendiment	No és la intenció del projecte mostrar aquesta informació
4.6	Estructures de comandament	No és la intenció del projecte mostrar aquesta informació
4.7	Característiques de seguretat/protecció de dades	No és la intenció del projecte mostrar aquesta informació
4.8	configuració dels paràmetres del sistema (com ara paràmetres d'aplicació o variables globals)	No és la intenció del projecte mostrar aquesta informació
4.9	Operacions manuals	No és la intenció del projecte mostrar aquesta informació
4.10	Components reutilitzables	No és la intenció del projecte mostrar aquesta informació
5.	Mapatge de requisits als elements del sistema	És possible assignar els requeriments en el EA mitjançant la integració amb Polarion

6.	Descripció dels modes de funcionament del components de sistema (inici, apagat, mode de repòs, mode de diagnòstic, etc.)	Modelarem el funcionament amb una màquina d'estats
7.	Descripció de les dependències entre els components de sistema relacionats amb els modes de funcionament	Relacionarem aquests estats amb els elements de sistema en un diagrama de blocs.
8.	Descripció del comportament dinàmic del sistema i els components del sistema	El comportament dinàmic es representarà en el <i>Activity Diagram</i> .

Taula 4 - Producte final d'arquitectura de sistema

[Algunes parts han sigut eliminades per motius de confidencialitat,
per més informació llegir
TFG_Acord_Confidencialitat_IglesiasLlaberia_Marc.pdf]

4. Conclusions i extensions

Finalment dedicarem aquest últim capítol a les conclusions i extensions que se'n poden extreure de la realització d'aquest treball. D'aquesta forma, es presentaran les limitacions amb que ens hem trobat durant el seu desenvolupament, tant des d'una perspectiva general del MBSE, com en casos particulars de la implementació a realitzar. També detallarem en una altre secció algunes de les possibles extensions i línies futures que podrien estudiar-se per obtenir un millor nivell de compliment amb la guia A-SPIICE que desemboca en un millor anàlisis i comprensió dels temes tractats en el projecte. Per últim, un cop descrites tals limitacions i futures exploracions, es presentaran les principals conclusions obtingudes que donaran clausura a aquest projecte de fi de grau.

4.1 Conclusions

El treball realitzat ha servit a l'empresa i a mi per portar a la pràctica els processos que ha de prendre un equip d'enginyeria de sistemes per poder recrear la representació més precisa i propera a la realitat del sistema d'interès.

El fet de centralitzar la informació del sistema en el model, en lloc de fer-ho en documentació massificada, ha proporcionat la capacitat d'agilitzar la recerca i la localització d'errors de disseny. Així, la complexitat del sistema s'ha pogut manejar de manera eficaç i eficient. Això suposa que s'agilitzi el procés de desenvolupament, i al estar centrats en aquesta metodologia obtindrem beneficis econòmics tant a curt com a llarg termini. Primer a curt, perquè es estalvia temps en la detecció de defectes en el disseny i no cal com tradicionalment cercar a la documentació pàgina per pàgina fins a trobar l'error en alguna característica del sistema o qualsevol altre error. I segon a llarg termini, perquè tenim al nostre poder més punts de vista a l'hora de comprendre el sistema en la seva totalitat, comptant amb un anàlisi de la seva descomposició funcional, podent modificar l'arquitectura realitzant estudis de *trade-off*, verificant aspectes específics dels seus components fent *gap-analysis*, etc. D'altra banda, l'avaluació de l'impacte en fer qualsevol modificació en un dels components del conjunt, és fàcilment observada atenent l'extensa traçabilitat generada.

Com a conclusions personals m'agradaria comentar que m'ha servit per dominar un llenguatge de modelatge com és el SysML juntament amb una eina tan potent com l'*Enterprise Architect* que indubtablement valoro molt en vistes del meu futur professional perquè dota al enginyer d'una capacitat d'anàlisis i comprensió de sistemes molt complexes.

4.2 Limitacions

Arribats a aquest punt, es pot afirmar que els objectius exposats al inici de la memòria no s'han realment assolit. El fet de dur a terme una "prova pilot" de l'empresa en un sector desconegut per mi ens ha dut a proposar objectius massa ambiciosos ja que partíem gairebé d'una base 0 sent una aposta pel futur el implantar aquestes pràctiques en l'equip d'enginyeria de sistemes. Principalment volíem dissenyar l'arquitectura de sistema d'una funció amb molta autonomia i que implícitament tenia implementades funcions molt diverses pel que vam decidir quedar-nos tan sols amb una d'elles.

També ens hem trobat amb la limitació de la traçabilitat, ja que ens la nostre base de requisits de *Polarion* no teníem cap mena de vincle entre requisits i tan sols els teníem agrupats en famílies. Aquesta feina ens hauria ajudat a poder traçar tots els requisits i poder dotar el sistema de una consistència completa.

En quant a l'eina que hem fet servir des d'un inici ens vam adonar que teníem una eina massa potent pel servei que en volíem fer, és a dir, és una eina molt útil però que en mans inexpertes perd d'utilitat envers altres opcions més limitades però d'una interfície més . Al no tractar-se d'una eina especialitzada ni a la automoció ni als sistemes ADAS prèviament a utilitzar-la l'hem de mollejar i pre-configurar de manera molt conscient ja que és una feina que condiona el resultat.

I com a limitació externa al projecte, cal dir que al tractar-se d'un projecte fora de l'àmbit educatiu i de caire tan empresarial, la informació és nul·la o molt poca. La competència en el sector fa que la permeabilitat de la informació no existeixi si no es de forma en que es pagui per aquesta.

4.3 Línies futures

La línia futura del projecte és molt clara perquè venim seguint la guia A-SPICE, per tant caldria començar a analitzar quins requisits tindrem a nivell de *software*, dissenyar-ne l'arquitectura, fer un disseny detallat de les unitats i verificar-ne la integració i per últim fer el test.

Però abans d'avançar en el següent procés cal revisar la feina feta de manera que es faci una verificació (l'eina té l'opció) si es creu que el nivell de detall és l'adequat pels fins que l'empresa en vulgui arribar a donar, sinó de igual manera es pot seguir precisant fins arribar

a un nivell de detall més baix, donant amb les lògiques de control dels circuits electrònics del controlador, o especificant les equacions de dinàmiques.

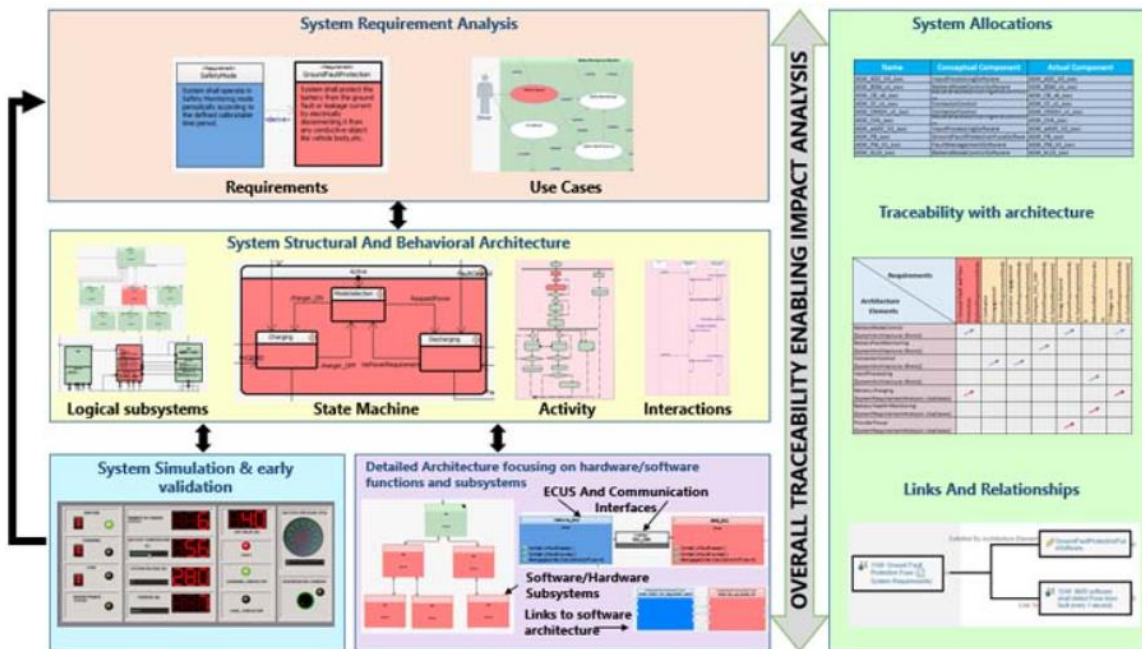


Figura 61 - Resum d'activitats disseny ADAS

[35]

Per donar una mica de context on ens trobem i fins on hem d'arribar per completar tot el *life-cycle* ens podem fixar en la Figura 61, que cobreix tot el sistema des de que obtenim els requisits fins a la seva simulació i validació o veiem a l'esquerra que en tot moment es porta un seguiment i una traçabilitat de tot.

5. Ontologia

Activity

Un conjunt d'accions que consumeixen temps i recursos i de les quals el rendiment és necessari per aconseguir o contribuir a consecució d'un o més resultats.

Capability

Una expressió d'un sistema, producte, funció o capacitat de procés d'assolir un objectiu específic en les condicions establertes.

Derived Requirement

Característiques detallades del sistema d'interès que normalment s'identifiquen durant l'obtenció dels requisits de les parts interessades, anàlisi de requisits, estudis comercials o validació.

Enabling System

Un sistema que complementa un sistema d'interès durant les etapes del seu cicle de vida, però no necessàriament hi contribueix directament la seva funció durant el funcionament.

Feature

Conjunt de funcions que en conjunt reben el nom comercial que la caracteritza.

Feature

Blocs lògics que duen a terme operacions i que en conjunt conformen característiques pel sistema.

Requirement

Una declaració que identifica un sistema, producte, característica del procés o restricció, que és inequívoca, pot ser verificat, i es considera necessari per l'acceptabilitat de les parts interessades.

Stakeholder

Una part que té un dret, una participació o una reclamació en un sistema o en la seva possessió de les característiques que satisfan les necessitats d'aquesta part i les expectatives.

System

Una combinació d'elements interactius organitzats per aconseguir-ho una o més finalitats declarades.

System Element

Un membre d'un conjunt d'elements que constitueixen un sistema.

System-of-interest

El sistema del qual esta sota consideració el *life cycle*

System Life Cycle

L'evolució amb el temps d'un sistema d'interès des de la concepció fins a la retirada.

System of Systems

Sistema de sistemes s'aplica a un sistema d'interès del qual els elements del sistema són els mateixos sistemes; típicament aquests comporten problemes interdisciplinaris a gran escala amb múltiples sistemes heterogenis i distribuïts.

Trade-off

Accions de presa de decisions que seleccionen entre diverses requisits i solucions alternatives en base al benefici net per a les parts interessades.

[51]

6. Referències

1. Report. (2020). United States Department of Transportation, NHTSA and FARS "Fatal Accident Reporting System" (FARS) [En línia] 2023.
<https://www-fars.nhtsa.dot.gov/People/PeopleAllVictims.aspx>
2. Vargas, J.; Alswiss, S.; Toker, O.; Razdan R.; Santos, J. *An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions*. *Sensors* 2021, 21, 5397. <https://doi.org/10.3390/s21165397>
3. Belvoir Defence Acquisition. 2001. *System Engineering Fundamentals*. Virginia: defense acquisition univ ft belvoir va. <http://www.dtic.mil/get-tr-doc/pdf?AD=ADA606327>.
4. Cockburn, Alistair. 2000. *Writing effective use cases*. Addison-Wesley Professional Reading.
5. California Department of Motor Vehicles. "Key Autonomous Vehicle Definitions," 2020. <https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/definitions>.
6. Wikipedia. Vehicle autònom [En línia] 2023.
https://ca.wikipedia.org/wiki/Vehicle_aut%C3%B2nom
7. SAE. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2018
8. History of SAE. [En línia] 2023 <https://www.sae.org/about/history>
9. UN Regulation on uniform provisions concerning the approval of vehicles with regards to Automated Lane Keeping System.
10. E. Donges. A Conceptual Framework for Active Safety in Road Traffic. *Vehicle System Dynamics*, 32(2-3):113–128, 1999.
11. BSI Standards Limited 2020. PAS 1883:2020. *Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specification*.
12. A. Eichberger. *Traffic safety systems and their benefit potential*, capítol 3, pàgines 66–105. Contributions to Primary, Secondary and Integrated Traffic Safety. Verlag Holzhausen GmbH, Austria, 1 edició, 2011.
13. CARE (base de dades de la UE d'accidents en carretera). [En línia]
https://ec.europa.eu/commission/presscorner/detail/en/ip_23_953
14. Economic Commission for Europe. Convention on Road Traffic Done at Vienna on 8 November 1968, 1968.
15. Standard (2021) *UN Regulation on uniform provisions concerning the approval of vehicles with regards to Automated Lane Keeping System*

16. Economic Commission for Europe. ECE-R 13-H - Uniform Provisions Concerning the Approval of Passenger Cars with Regard to Braking, 2015.
17. National Highway and Traffic Safety Administration (NHTSA). Laboratory Test Procedure for the New Car Assessment Program Electronic Stability Control Testing and FMVSS No. 126, Electronic Stability Control Systems Indicative Test for Compliance. Testing protocol, September 9, 2011.
18. Technical Committee ISO/TC 204. Intelligent transport systems - Adaptive Cruise Control systems - Performance requirements and test procedures. Standard ISO 15622:2010(E), International Organisation for Standardization (ISO), April 4 2010.
19. SAE Safety and Human Factors Committee. SAE J2399 - Adaptive Cruise Control (ACC) Operating Characteristics and User Interface. Standard SAE J2399, SAE International, December 2003.
20. Euro-NCAP. *Assesment Protocol – Safety Assist Safe Driving*. Version 10.1.2. February 2023.
21. Euro-NCAP. *Vision 2030. A Safer Future for Mobility*. November 2022.
22. Wikipedia. *Software architecture description*. [En línia] 2023.
https://en.wikipedia.org/wiki/Software_architecture_description.
23. Serban, Alexandru Constantin, Erik Poll, and Joost Visser. "A Standard Driven Software Architecture for Fully Autonomous Vehicles." 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), [En línia] 2018.
<https://doi.org/10.1109/icsa-c.2018.00040>.
24. M Staron, *Automotive Software Architectures*, Springer International Publishing AG 2017.
25. Koopman, Philip, Uma Ferrell, Frank Fratrick, and Michael Wagner. "A Safety Standard Approach for Fully Autonomous Vehicles." Lecture Notes in Computer Science Computer Safety, Reliability, and Security, [En línia] 2019, 326–32.
https://doi.org/10.1007/978-3-030-26250-1_26
26. Ian Alexandre, *Stakeholders – Who is Your System For?* [En línia] 2023
<https://www.scenarioplus.org.uk/papers/stakeholders/stakeholders.htm>
27. Hermann Winner, Stephan Hakuli, Felix Lotz, Christina Singer. *Handbook of Driver Assistance Systems. Basic Information, Components and Systems for Active Safety and Comfort*. Springer International Publishing Switzerland, 2016.
28. Alexander, Kossiakoff, William N Sweet, Samuel J Seymour, and Steven M Biemer. 2011. *Systems engineering principles and practice*. John Wiley & Sons.

29. Brian Dunbar, NASA. *Systems Engineering Handbook, 2.0 Fundamentals of Systems Engineering*, 2019 [En línia]
<https://www.nasa.gov/seh/2-fundamentals>
30. VDA QMC Working Group 13 / Automotive SIG, *Automotive SPICE Process Assessment / Reference Model*, version 3.0, November 2017.
31. Dori D. *Object-Process Methodology – A Holistic Systems Paradigm*. Verlag, Berlin, Heidelberg, New York: Springer. 2002.
32. Miles, R.F. *System Concepts*. New York, NY, USA: Wiley and Sons, Inc. (ed) 1973.
33. Johnson, R.A., F.W. Kast, and J.E. Rosenzweig. 1963. *The Theory and Management Systems*. New York, NY, USA: McGraw-Hill Book Company. 1963.
34. Steiner, R., Naval, R., & Systems, M. (n.d.). *Systems Architectures and Evolvability: Definitions and Perspective*. [En línia] 2023
https://members.tripod.com/Rick_Steiner/Evolarch.pdf
35. Mobility insider. "¿Qué es ASPICE?". APTIVE.com. [En línia] 2023.
<https://www.kpit.com/insights/model-based-system-engineering/>
36. Lenny Delligatti, *SysML Distilled-A Brief Guide to the Systems Modeling Language*. Pearson Education, Inc. 2014.
37. David Hetherington, *SysML for Beginners-Using Sparx Enterprise Architect*. Austin, Texas, USA: Asatte Press, 2019.
38. Siemens, "Scaled-Agile with Integrated System Modeling", [blogs.sw.siemens.com](https://blogs.sw.siemens.com/teamcenter/scaled-agile-with-integrated-system-modeling/). [En línia] 2023
39. Federal Highway Administration. U.S. Department of Transportation, "Systems Engineering for Intelligent Transportation Systems", [ops.fhwa.dot.gov](https://ops.fhwa.dot.gov/publications/seitsguide/section3.htm). [En línia]
<https://ops.fhwa.dot.gov/publications/seitsguide/section3.htm>
40. Friedenthal, Sanford, et al., *A Practical Guide to SysML, Second Edition: The Systems Modeling Language*, Boston: MK/ OMG Press, 2011.
41. E. R. Carroll y R. J. Malins, «Systematic Literature Review: How is Model-Based Systems Engineering Justified?,» Albuquerque, 2016.
42. IBM, [En línia] 2023. <https://www.ibm.com/es-es/products/uml-tools>.
43. Arcadia, [En línia]. 2023. <https://www.eclipse.org/capella/arcadia.html>

44. J.P. Elm, D.R. Goldenson, "The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey." Special Report CMU/SEI-2012-SR-009 CERT Program. Carnegie Mellon University. November 2012.
45. Cihan H. Dagli, "Requirements for a Metamodel to Facilitate Knowledge Sharing between Project Stakeholders", Missouri University of Science and Technology, St. Louis, MO, USA, 2012
46. R. O. Lavandeira, "Análisis y Aplicación de la Ingeniería de Sistemas Basada en Modelos" de Treball de Fi de Grau, Madrid, 2020.
47. L. S. Alonso, "Modelado de un sistema de planificación de procesos de fabricación mediante técnicas basadas en Model Based System Engineering" de Treball de Fi de Grau, Sevilla, 2022.
48. Sparx Systems Prolaborate. Meta Modeling (Jul. 28, 2020) Accedit: Ago. 23, 2023. [En línia] <https://www.youtube.com/watch?v=i7hSF0nQiMI>
49. Sparx Systems. The Basics of MDG - How to Build a Custom MDG from Scratch. (Des. 13, 2021) Accedit: Ago. 23, 2023. [En línia] <https://www.youtube.com/watch?v=Rz9aCvh9Nq4>
50. Sparx Systems. MDGs the Next Steps. (Nov. 15, 2021) Accedit: Ago. 23, 2023. [En línia] <https://www.youtube.com/watch?v=UgKTAIymgeo&t=2742s>
51. Incose. *SYSTEMS ENGINEERING HANDBOOK A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES*, 2006

