DASSAULT AVIATION

Vérification, validation et certification des applications embarquées : quid de la robotique ?

Paris, le 22 novembre 2018

Agenda

- Vérification, Validation et Certification
- Quelques Chiffres
- Le Monde Aéronautique
- Le Monde Automobile
- Et le Monde Robotique ?
- *Une conclusion ?*

Remerciements

Saadia Dhouib – CEA-LIST

Félix Ingrand – LAAS

Nicolas Dulac – INTEMPORA

Thierry Meslet – BA Systems

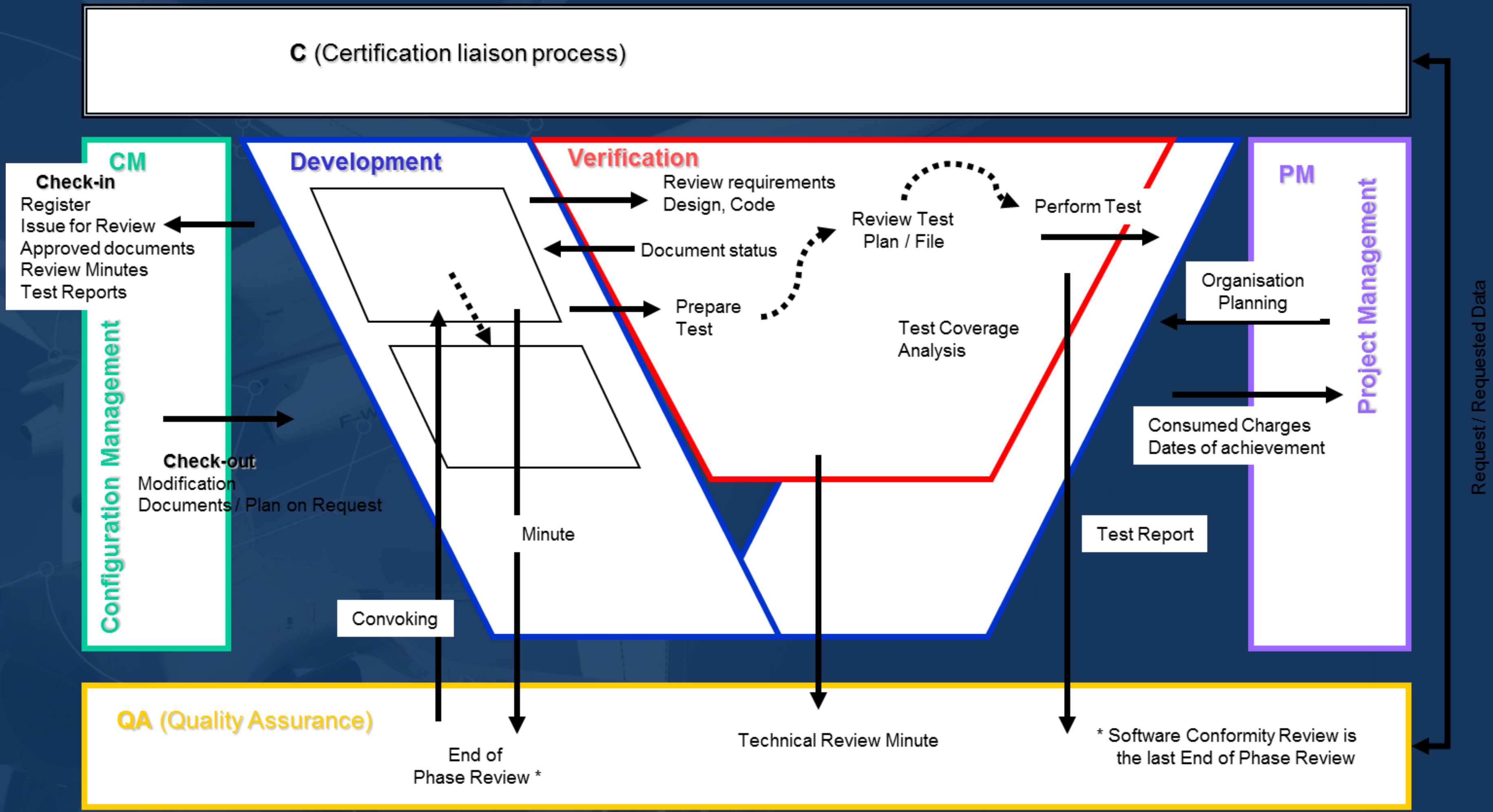
Herman Bruyninckx – UCL

DASSAULT AVIATION

Agenda

- Vérification, Validation et Certification
- Quelques Chiffres
- Le Monde Aéronautique
- Le Monde Automobile
- Et le Monde Robotique ?
- *Une conclusion ?*

Vérification, Validation et Certification



- - - ▶ Transition Inside a process: from one activity to another activity

⟶ Transition from 1 process to another process

DASSAULT AVIATION

Quelques Informations

	Aéronautique (Airliners, bizjets, GA et RA)	Automobile (dont la voiture « autonome »)	Drones	Robotique Logistique (hors cobot et extérieur*)	Robotique personnelle (hors aspirateurs)
Nombre de machines	~ 40000 airliners	>> 1Million	Qques milliers ↗	Quelques dizaine milliers ↗	Quelques centaines (↗↗?)
Personnes en connexion	Plusieurs centaines	1 dizaine max (hors autocar)	0 Personnes et biens survolés	Principe de ségrégation (cobotique en route)	Quelques personnes
Risque assumé	Airliner , bizjet : 10^{-6} / heure de vol GA : 10^{-4} / heure de vol	Pas de chiffre global mais constaté > 10^{-4}	Pas de chiffres mais volonté 10^{-4} pour les plus légers	10^{-6} / heure de fonctionnement (objectif – pas de recueil des incidents)	Pas de chiffres
Autorité de certification / Suivi indépendante	OUI / OUI	NON / EN COURS**	EN COURS / EN COURS	NON / NON	NON / NON
Middleware et OS	RT ARINC ECOA	Ad hoc RT (QNX-like) RTMAPS ROS	Ad hoc Paparazzi OROCOS ROS Xenomai	Ad hoc RT OROCOS (médical) ROS (XP)	Ad hoc (?) OpenRobots OROCOS ROS
Approche « Model Driven »	OUI	EN COURS	NON	NON	INITIATIVES

** Travail en cours pour la logistique en extérieur mais problème réglementaire délicat*

*** voir Etat de Californie, et ce travail universitaire.*

Agenda

- Vérification, Validation et Certification
- Quelques Chiffres
- Le Monde Aéronautique
- Le Monde Automobile
- Et le Monde Robotique ?
- *Une conclusion ?*

DASSAULT AVIATION

La vérification et validation du logiciel
Aéronautique – la DO178C

- Allocation des items software à des niveaux de qualité
- Les niveaux de qualité
- Quelques éléments demandés par le processus de développement

The software IDAL allocation process

- At system level, there are the system development process and the associated safety analysis process that identifies the failure conditions and their effects. The system is decomposed in items that are allocated to an IDAL*.
- The software items reliability is process based because it is considered too difficult to assess the software reliability *(Extract from DO178C : Many methods for predicting software reliability based on developmental metrics have been published, for example, software structure, defect detection rate, etc. This document does not provide guidance for those types of methods, because at the time of writing, currently available methods did not provide results in which confidence can be placed.)*.
- For this reason, the software reliability is demonstrated through a process application. The higher the software DAL** (IDAL), the higher the effort required by the process application : *(extract from ARP4754A : Errors are mitigated by implementation of a Development Assurance. Development Assurance Process establishes confidence that system has been accomplished in a sufficiently disciplined manner to limit the development errors that could impact aircraft safety.)*

** Item Development Assurance Level*
*** Development Assurance Level*

Some properties of the processus

- It is a very formal processus (see Annex A1)
  - The **software development** processes (**requirements, desing, coding, integration**) that are defined in the Software Development Plan (**SDP**).
  - The integral processes that ensure the correctness and control of, and confidence in the software life cycle processes and their outputs
- More than 10 documents per item are needed to follow it (see Annex A2)
  - Software plans
  - Software standards
  - Software specification
  - …
- 10 tables are managed to qualify each item (See Annex A3)
- Configuration Management issue (See Annex A5)
- Quality Assurance issue : <u>An independant quality Assurance shall be performed in order to assess that the plans/standards/procedures are correctly applied.</u>

The software DAL

There are 4 DAL levels (IDAL in ARP4754A). The higher is the DAL, the higher is the verification effort. In a very simple way, the IDAL definition is :

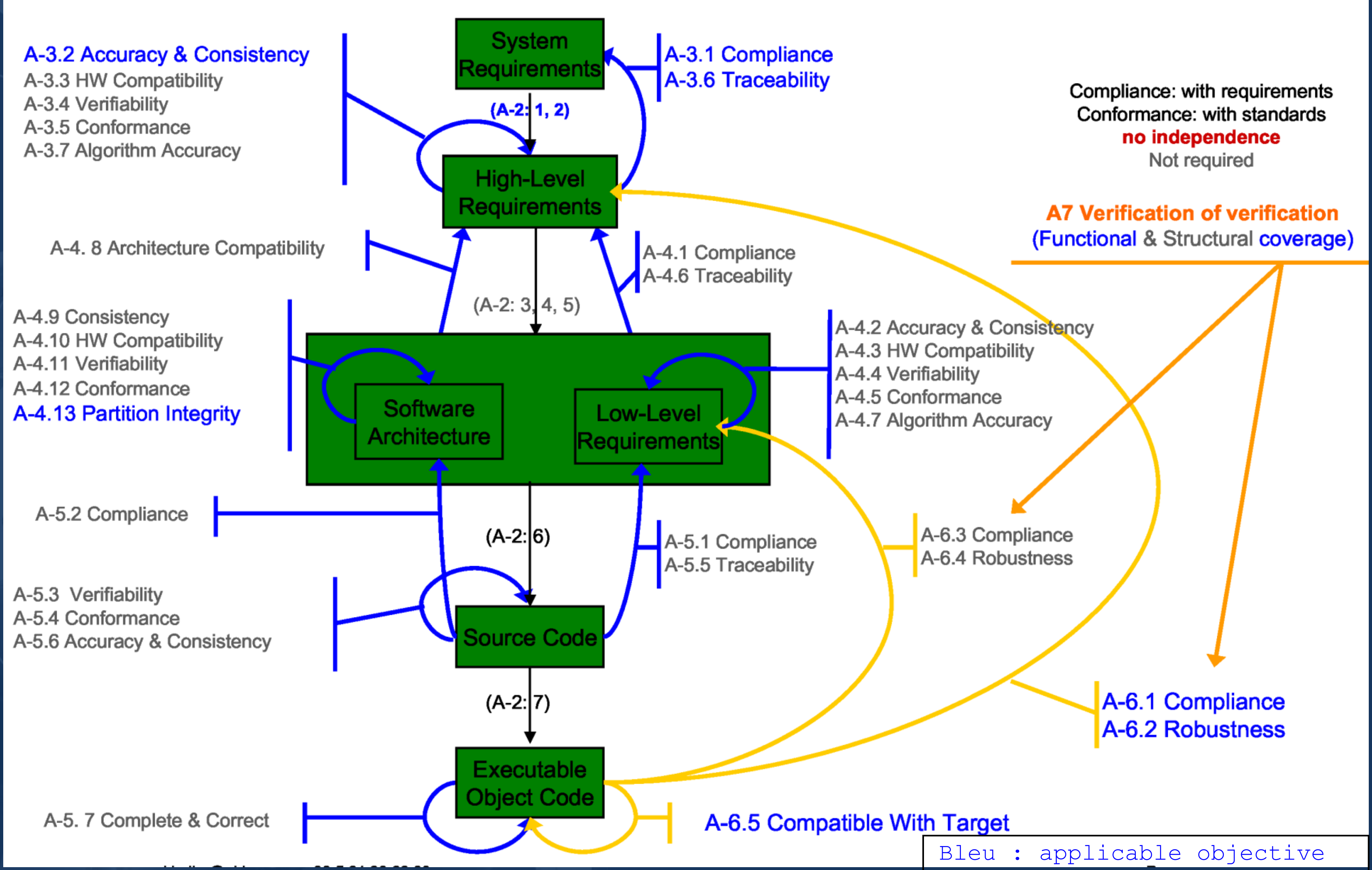
- DAL D : The software item is considered as a black box. There is only one level of specification (High level Requirements (HLR)). The verification of the compliance/traceability of the HLR against the System Requiements allocated to the software. The test cases are written against the HLR. The binary is executed on the target for verifying its compliance to the HLR and its robustness.

- DAL C : DAL D + The software item is considered as a white box. The detailed definition (Low Level Requirements (LLR)) is required with the verification of the compliance/traceability of the LLR against the HLR and source code against the LLR. The structural coverage of the source code is at statement level. The binary is executed on the target for verifying its compliance to the HLR and LLR and its robustness.

- DAL B : DAL C + compatibility target + independence for some verification activities + structural coverage at decision level

- DAL A : DAL B + higher level of independence for verification activities + structural coverage at MC/DC + traceability demonstration between object code and source code.

Software verification objectives
DAL D software



System Requirements

A-3.2 Accuracy & Consistency
A-3.3 HW Compatibility
A-3.4 Verifiability
A-3.5 Conformance
A-3.7 Algorithm Accuracy

A-3.1 Compliance
A-3.6 Traceability

(A-2: 1, 2)

High-Level Requirements

Compliance: with requirements
Conformance: with standards
no independence
Not required

A7 Verification of verification
(Functional & Structural coverage)

A-4. 8 Architecture Compatibility

A-4.1 Compliance
A-4.6 Traceability

(A-2: 3, 4, 5)

A-4.9 Consistency
A-4.10 HW Compatibility
A-4.11 Verifiability
A-4.12 Conformance
A-4.13 Partition Integrity

Software Architecture

Low-Level Requirements

A-4.2 Accuracy & Consistency
A-4.3 HW Compatibility
A-4.4 Verifiability
A-4.5 Conformance
A-4.7 Algorithm Accuracy

A-5.2 Compliance

(A-2: 6)

A-5.1 Compliance
A-5.5 Traceability

A-6.3 Compliance
A-6.4 Robustness

A-5.3 Verifiability
A-5.4 Conformance
A-5.6 Accuracy & Consistency

Source Code

(A-2: 7)

A-6.1 Compliance
A-6.2 Robustness

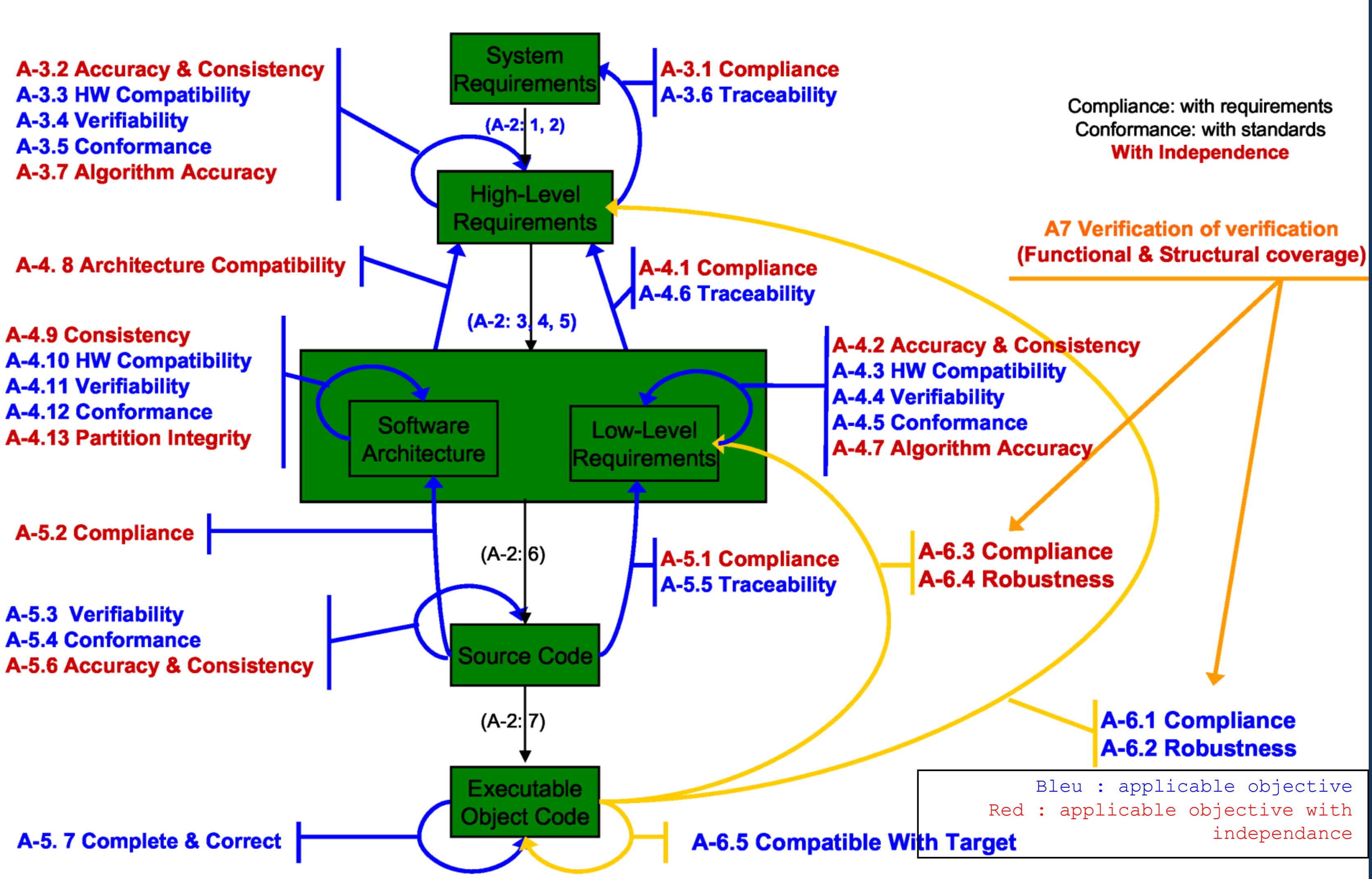
Executable Object Code

A-5. 7 Complete & Correct

A-6.5 Compatible With Target

Bleu : applicable objective

Software verification objectives
DAL D software



System Requirements

(A-2: 1, 2)

High-Level Requirements

(A-2: 3, 4, 5)

Software Architecture

Low-Level Requirements

(A-2: 6)

Source Code

(A-2: 7)

Executable Object Code

A-3.2 Accuracy & Consistency
A-3.3 HW Compatibility
A-3.4 Verifiability
A-3.5 Conformance
A-3.7 Algorithm Accuracy

A-3.1 Compliance
A-3.6 Traceability

Compliance: with requirements
Conformance: with standards
With Independence

A7 Verification of verification
(Functional & Structural coverage)

A-4. 8 Architecture Compatibility

A-4.1 Compliance
A-4.6 Traceability

A-4.9 Consistency
A-4.10 HW Compatibility
A-4.11 Verifiability
A-4.12 Conformance
A-4.13 Partition Integrity

A-4.2 Accuracy & Consistency
A-4.3 HW Compatibility
A-4.4 Verifiability
A-4.5 Conformance
A-4.7 Algorithm Accuracy

A-5.2 Compliance

A-5.1 Compliance
A-5.5 Traceability

A-6.3 Compliance
A-6.4 Robustness

A-5.3 Verifiability
A-5.4 Conformance
A-5.6 Accuracy & Consistency

A-6.1 Compliance
A-6.2 Robustness

A-5. 7 Complete & Correct

A-6.5 Compatible With Target

Bleu : applicable objective
Red : applicable objective with independance

Tests based*

DO178C verification means are reviews, analysis, tests and simulations (that have been added by the model based supplement DO331).

There are a lot of reviews (plans, standards, procedures, HLR, architecture, LLR, code, integration, verification results, …) that means a lot of minutes and checklists that have to be managed in configuration.

Most of DO178C verification activities still rely on tests even if a lot of progress have been performed by enabling the usage of the formal methods and simulations for some dedicated verifications.

- Tests: several objectives of the table A6 required to perform tests on the target at least for the HLR test cases.
- Structural coverage is computed after executing requirements based tests (It is a way for identifying unintended function).

** Some possible activities described in Annex A4*

Tools qualification issue

- If the tool eliminates, reduces, or automates processes of DO178C and if its outputs are not verified as specified in section 6 of DO178C, **the tool shall be qualified** at the Tool Qualification Level (TQL) defined by table in Annex A7. Five levels of tool qualification (TQL-1 to TQL-5) are identified based on the tool use and its potential impact in the software life cycle processes. TQL-1 is the most rigorous level and TQL-5 is the least rigorous level. When assessing the impact of a given tool, the criteria should be considered sequentially from criteria 1 to criteria 3 (see Annex A6).

- For example, for a TQL-1 tool (e.g. a code generator used at DAL A), the tool shall be developed with a level of rigor very close to the DAL A of DO178C. DO330 defines the tool qualification objectives.

Certification liaison issue

- The certification process at software level is driven by DO178C and possibly several dedicated CRIs (Certification Review Items) / IPs (Issue Papers).

- ***It is submitted to the acceptance of a governmental Agency (EASA for Europe/FAA for US) thanks to specific audits (see FAA order 8110.49):***
  - Approval of the “Applicant" software plans
    - Audit SOI1 "planning review"
  - Approval of the compliance with applicable requirements of the “Applicant" software development data
    - Audit SOI2 "development review"
    - Audit SOI3 "verification review"
  - Approval of the “Applicant" Software Accomplishment Summary describing any deviations from the software plans and open problem reports
    - Audit SOI4 "final review"

SOI : Stage of Involvement

En guise de conclusion – Domaine Aéronautique

- La contrainte de sécurité entraine des exigences sur :
  - L'organisation mise en place
    - Développeurs
    - manageurs
  - Les méthodes de développement
    - Du matériel
    - Et du logiciel
    - y compris les outils utiles au développement
  - La différenciation des rôles à tous les niveaux
    - L'élaboration des règlements
    - L'élaboration des standards industriel
    - Le développement
    - La responsabilité de l'acceptation
- Le processus est très long
- Le processus est très couteux
- Pas de prise en compte des approches Machine Learning

- CE PROCESSUS EST INDISPENSABLE A GARANTIR LES NIVEAUX DE SÉCURITÉ CIBLÉS

DASSAULT AVIATION

Agenda

- Vérification, Validation et Certification
- Quelques Chiffres
- Le Monde Aéronautique
- Le Monde Automobile
- Et le Monde Robotique ?
- *Une Conclusion ?*

SIA SOCIÉTÉ DES INGÉNIEURS DE L'AUTOMOBILE

Bienvenue à la Journée d'études
SIA validation du véhicule autonome

Journée d'Etude et Exposition

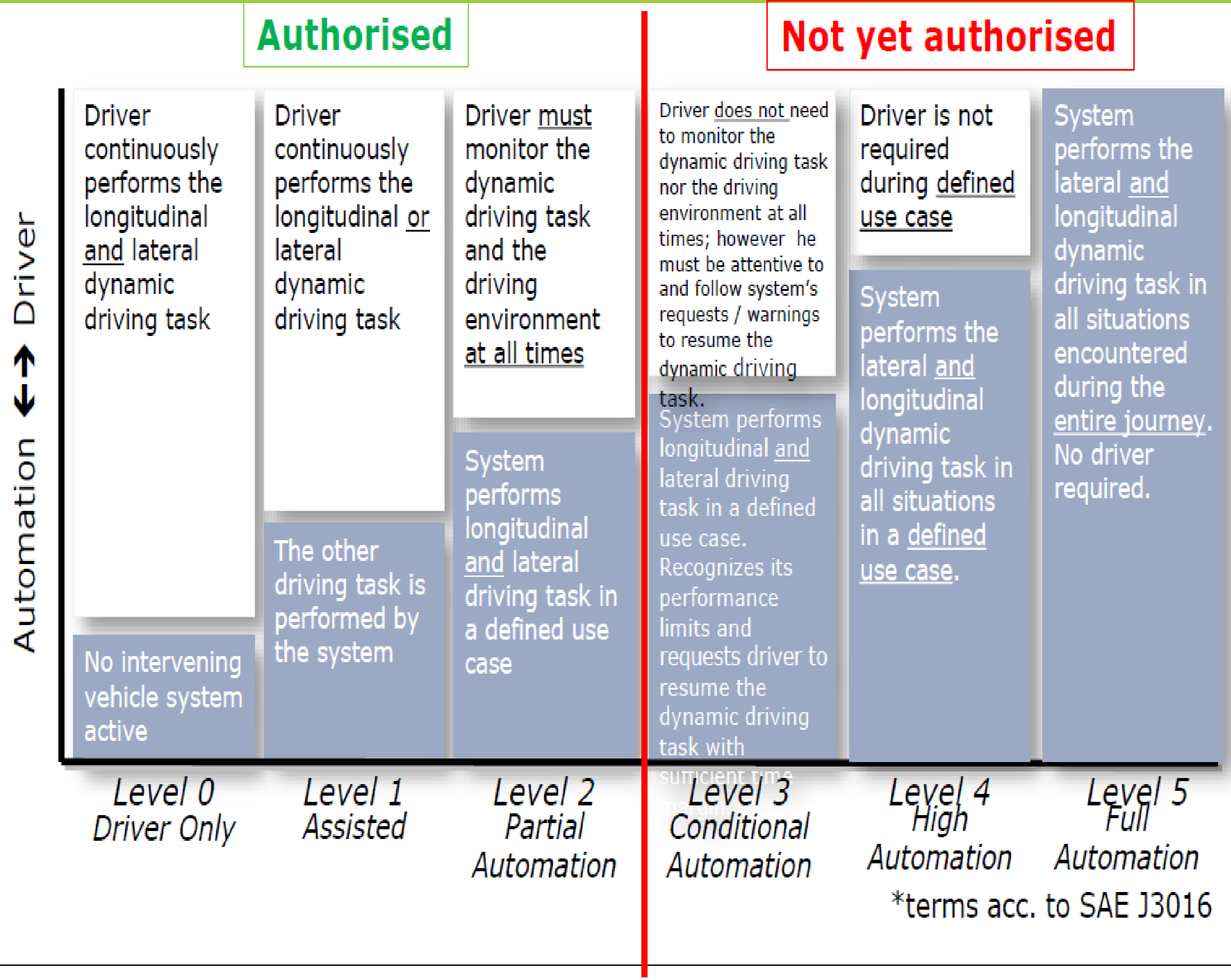
SIA VALIDATION DU VÉHICULE AUTONOME

Combinaison de simulations numériques, d'essais et de tests en simulateurs

MAISON DES TRAVAUX PUBLICS - FNTP
PARIS, FRANCE
27 JUIN 2018

SIA SOCIÉTÉ DES INGÉNIEURS DE L'AUTOMOBILE

TRUE AUTOMATION STARTS FROM LEVEL 3 (SAE)



Authorised

Not yet authorised

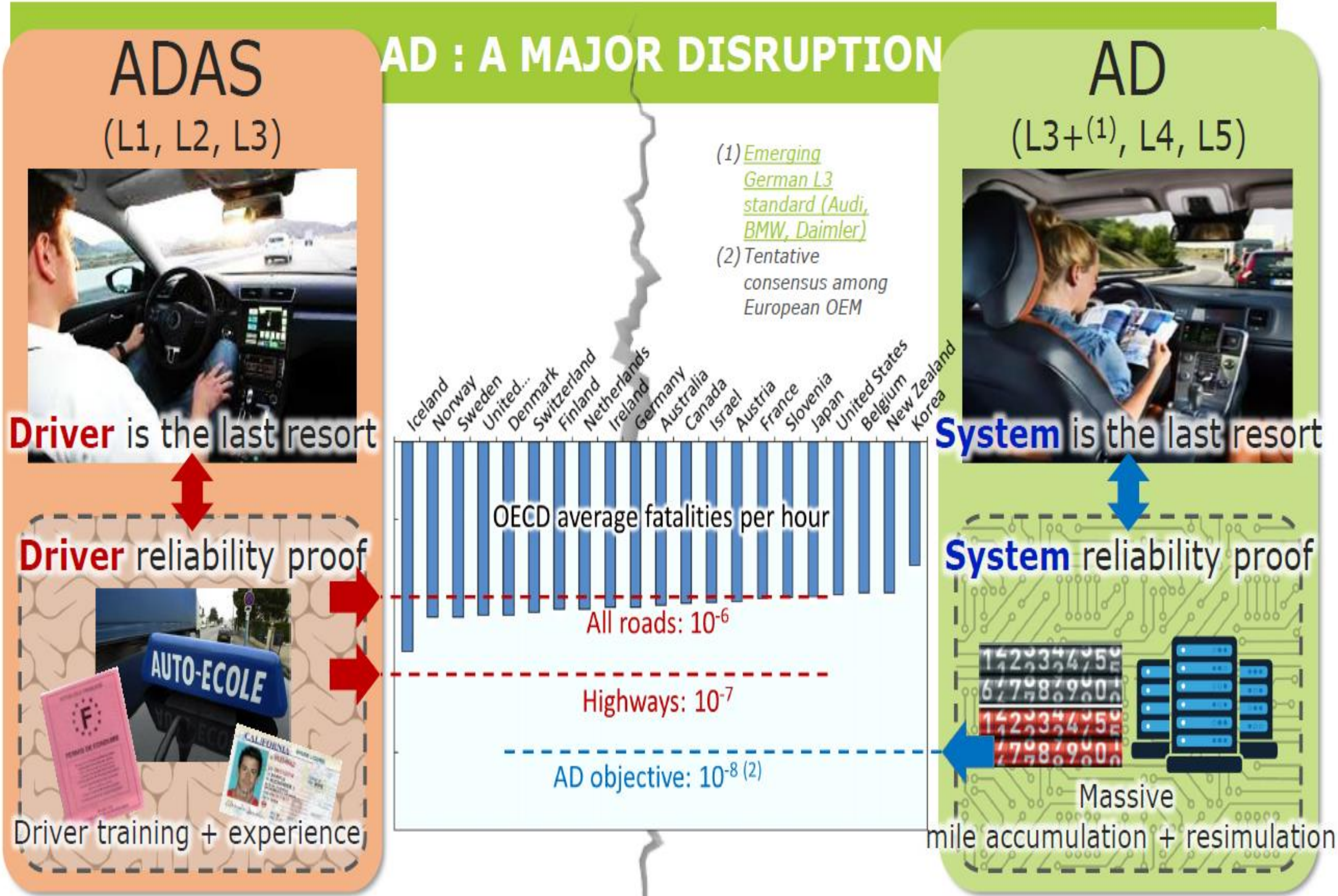
Automation ↔ Driver

Level 0 Driver Only	Level 1 Assisted	Level 2 Partial Automation	Level 3 Conditional Automation	Level 4 High Automation	Level 5 Full Automation
Driver continuously performs the longitudinal and lateral dynamic driving task	Driver continuously performs the longitudinal or lateral dynamic driving task	Driver must monitor the dynamic driving task and the driving environment at all times	Driver does not need to monitor the dynamic driving task nor the driving environment at all times; however he must be attentive to and follow system's requests / warnings to resume the dynamic driving task.	Driver is not required during defined use case	System performs the lateral and longitudinal dynamic driving task in all situations encountered during the entire journey. No driver required.
No intervening vehicle system active	The other driving task is performed by the system	System performs longitudinal and lateral driving task in a defined use case	System performs longitudinal and lateral driving task in a defined use case. Recognizes its performance limits and requests driver to resume the dynamic driving task with sufficient time margin.	System performs the lateral and longitudinal dynamic driving task in all situations in a defined use case.	

*terms acc. to SAE J3016

THE MAJOR STAKE IS SAFETY

AD : A MAJOR DISRUPTION



ADAS

(L1, L2, L3)

Driver is the last resort

Driver reliability proof

Driver training + experience

AD

(L3+(1), L4, L5)

System is the last resort

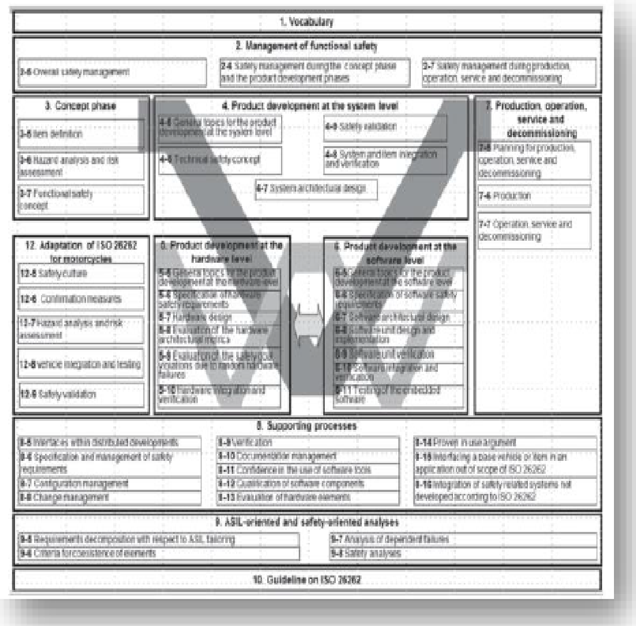
System reliability proof

Massive mile accumulation + resimulation

(1) Emerging German L3 standard (Audi, BMW, Daimler)

(2) Tentative consensus among European OEM

SAFETY DEVELOPMENT AND VALIDATION



1. Vocabulary

2. Management of functional safety
- 2-5 Overall safety management
- 2-6 Safety management during the concept phase and the product development phases
- 2-7 Safety management during production, operation, service and decommissioning

3. Concept phase
- 3-5 Item definition
- 3-6 Hazard analysis and risk assessment
- 3-7 Functional safety concept

4. Product development at the system level
- 4-5 General topics for the product development at the system level
- 4-6 Technical safety concept
- 4-7 System architectural design
- 4-8 System and item integration and verification
- 4-9 Safety validation

7. Production, operation, service and decommissioning
- 7-5 Planning for production, operation, service and decommissioning
- 7-6 Production
- 7-7 Operation, service and decommissioning

12. Adaptation of ISO 26262 for motorcycles
- 12-5 Safety culture
- 12-6 Confirmation measures
- 12-7 Hazard analysis and risk assessment
- 12-8 Vehicle integration and testing
- 12-9 Safety validation

5. Product development at the hardware level
- 5-5 General topics for the product development at the hardware level
- 5-6 Specification of hardware safety requirements
- 5-7 Hardware design
- 5-8 Evaluation of the hardware architectural metrics
- 5-9 Evaluation of the safety goal violations due to random hardware failures
- 5-10 Hardware integration and verification

6. Product development at the software level
- 6-5 General topics for the product development at the software level
- 6-6 Specification of software safety requirements
- 6-7 Software architectural design
- 6-8 Software unit design and implementation
- 6-9 Software unit verification
- 6-10 Software integration and verification
- 6-11 Testing of the embedded software

8. Supporting processes
- 8-5 Interfaces within distributed developments
- 8-6 Specification and management of safety requirements
- 8-7 Configuration management
- 8-8 Change management
- 8-9 Verification
- 8-10 Documentation management
- 8-11 Confidence in the use of software tools
- 8-12 Qualification of software components
- 8-13 Evaluation of hardware elements
- 8-14 Proven in use argument
- 8-15 Interfacing a base vehicle or item in an application out of scope of ISO 26262
- 8-16 Integration of safety related systems not developed according to ISO 26262

9. ASIL-oriented and safety-oriented analyses
- 9-5 Requirements decomposition with respect to ASIL tailoring
- 9-6 Criteria for coexistence of elements
- 9-7 Analysis of dependent failures
- 9-8 Safety analyses

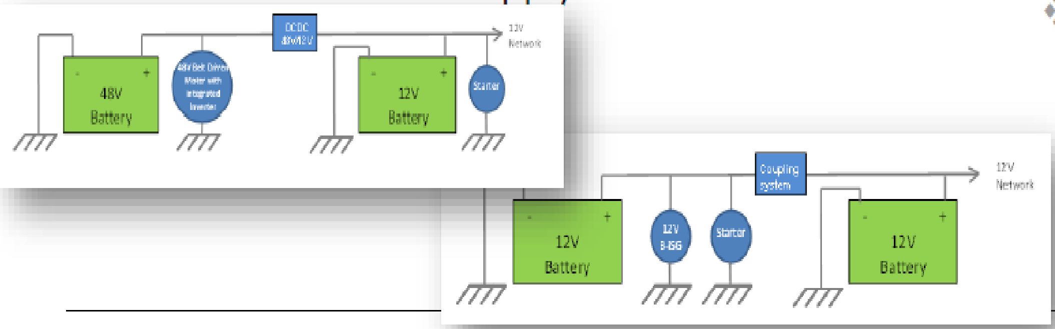
10. Guideline on ISO 26262

ISO 26262 defines how to assess a risk and the nec
for each step:

- System
- Software
- Hardware
- Production…

Redundancy for Autonomous Driving:

- Redundant Sensors & Actuators
- Redundant Communication Networks
- Redundant Power supply Networks



Additional Safety Stakes:

- For Autonomous Driving, Automotive EE Architecture has to switch from Fail Safe design to Fail Operational.
- Safety has also to consider SOTIF (Safety of the Intended Functionality)

CONCLUSION

- AD success depends on societal acceptance
- Safety is the biggest stake
- E/E architecture is the first lever
- Validation by combination of data collection and simulation are the second lever.

En guise de conclusions – Domaine Automobile

- L’avènement du véhicule autonome amène à renforcer les exigences
- Le monde Automobile se rapproche du monde Aéronautique
- Le processus Aéronautique actuel est trop long et trop couteux
- Application au domaine Automobile – véhicule autonome
  - La nature du marché est très différente (~40000 / >> Millions)
  - Nécessité de diminuer le temps et les coûts
  - Possibilité par l’amortissement de l’investissement ?
  - Mise en place en cours d’autorités indépendantes (Certification et Opération)
  - Certification hors auto-certification probablement à venir

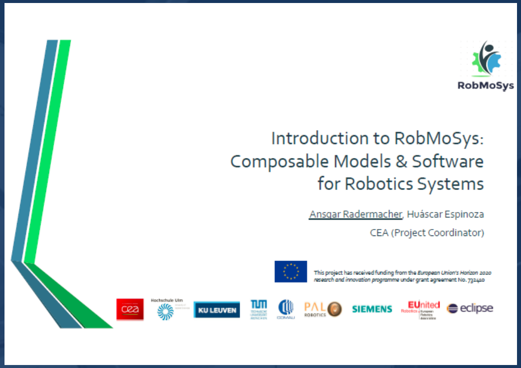
- Opportunités pour les autres domaines ?

Agenda

- Vérification, Validation et Certification
- Quelques Chiffres
- Le Monde Aéronautique
- Le Monde Automobile
- Et le Monde Robotique ?
- *Une Conclusion ?*

Les Initiatives en cours – Deux exemples

- Les approches « Model-Driven » : le projet Européen RobMoSys



RobMoSys

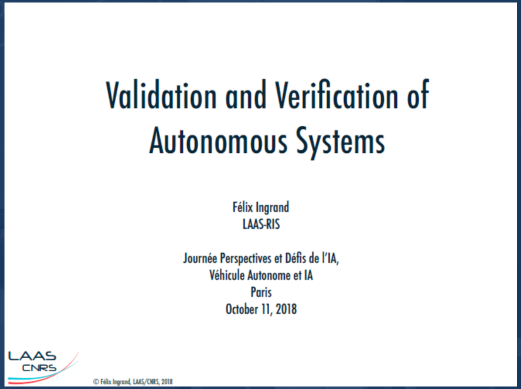
Introduction to RobMoSys:
Composable Models & Software
for Robotics Systems

Ansgar Radermacher, Huáscar Espinoza

CEA (Project Coordinator)

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No. 732410

- Les travaux de Vérification / Validation : Propositions LAAS



Validation and Verification of
Autonomous Systems

Félix Ingrand
LAAS-RIS

Journée Perspectives et Défis de l'IA,
Véhicule Autonome et IA
Paris
October 11, 2018

© Félix Ingrand, LAAS/CNRS, 2018

- Axes de recherche - Propositions : Point de vue

RoMoSys in a Nutshell

RobMoSys

- **RobMoSys**: Composable Models and Software for Robotic Systems
- In response to **H2020** Project – ICT-26- TOPIC : System abilities, development and pilot installations
- SubTopic c: **Innovation Action** on systems development technology.

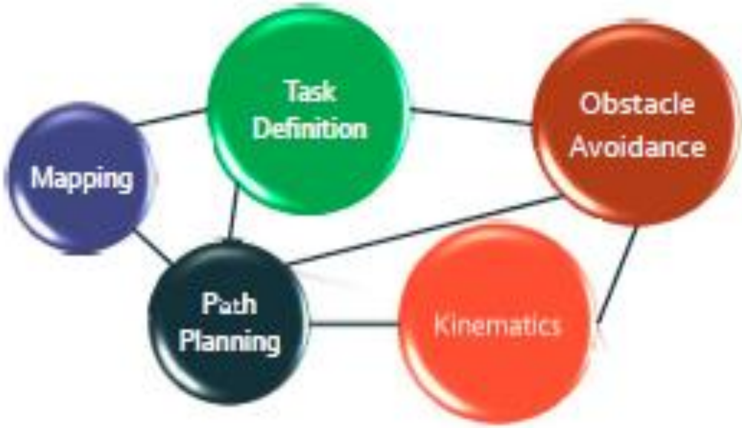
The "**System development tools**" sub-call

- **Start Date 01/01/2017**
- **End Date 31/12/2020**
- **Duration 4 Years**
- **Budget 8M,** where **4 M for Open-Calls**
- **Web Site** http://robmosys.eu/

Complexity in SW Robotics

RobMoSys

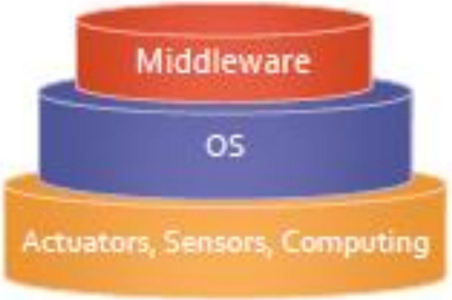
Application



Algorithms

Component Release
Quality Management
Design
Implementation
Safety Assessment
Validation & Verification

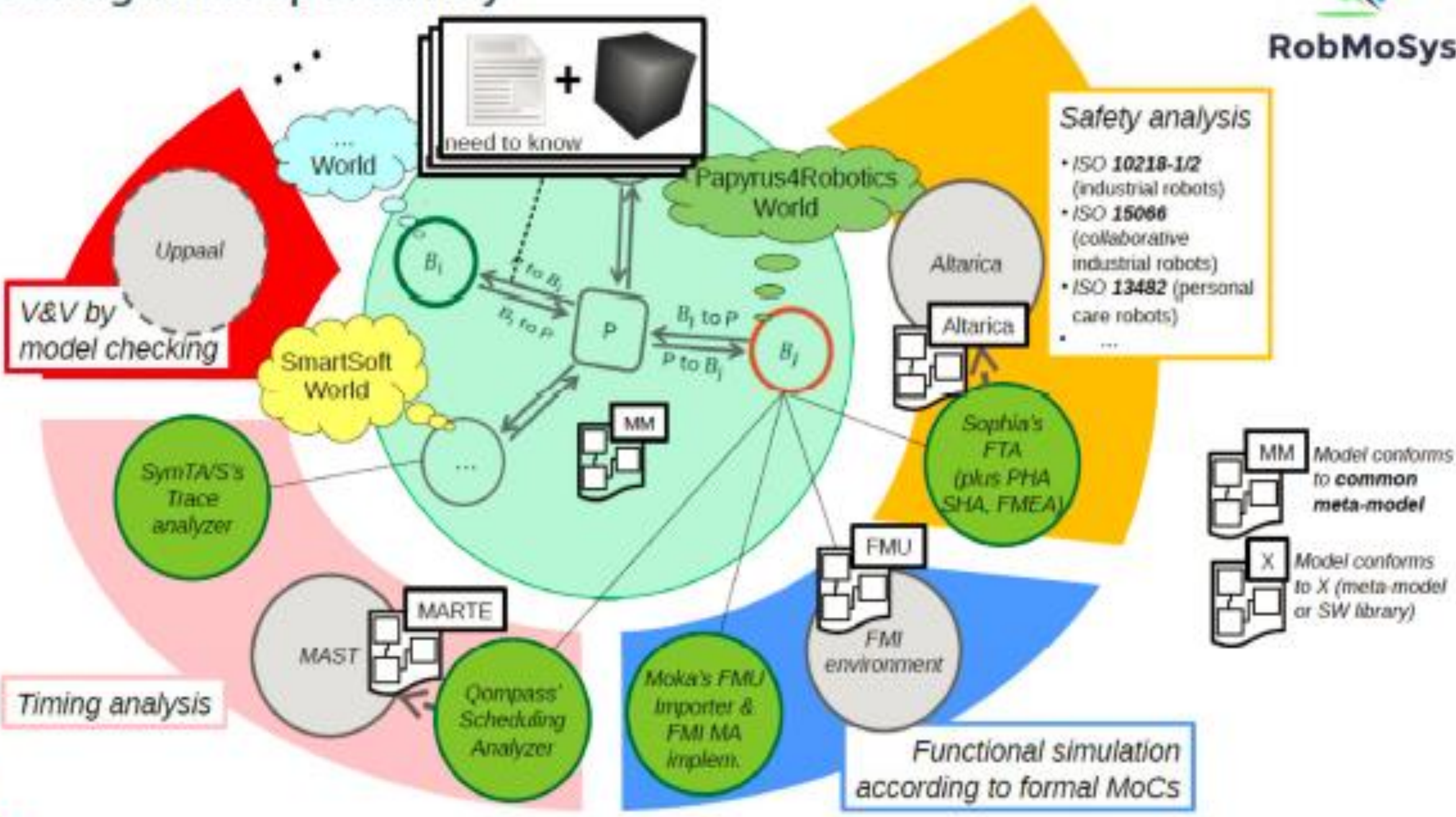
Stakeholders



Platforms

RobMoSys

Tooling Interoperability



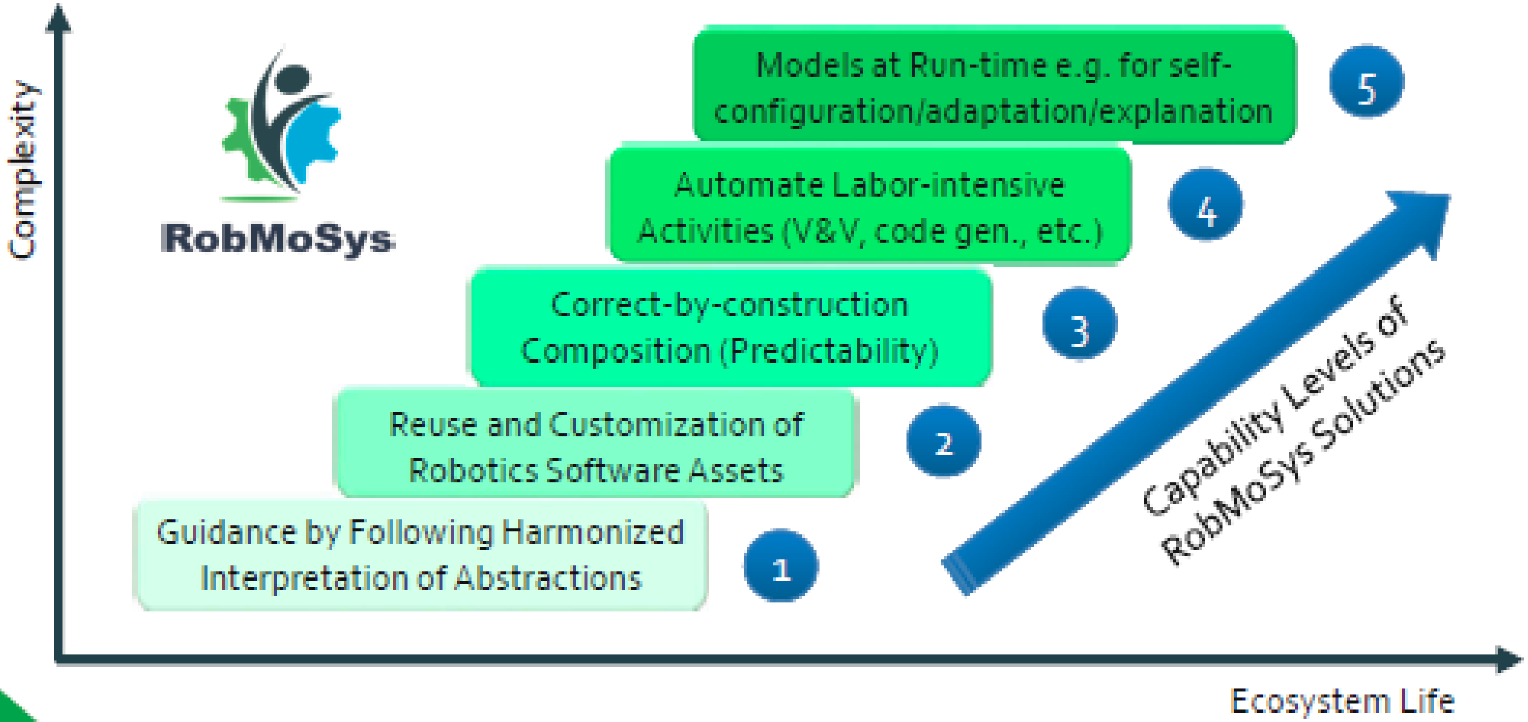
Safety analysis

- *ISO **10218-1/2*** (industrial robots)
- *ISO **15066*** (collaborative industrial robots)
- *ISO **13482*** (personal care robots)
- ...

MM *Model conforms to **common meta-model***

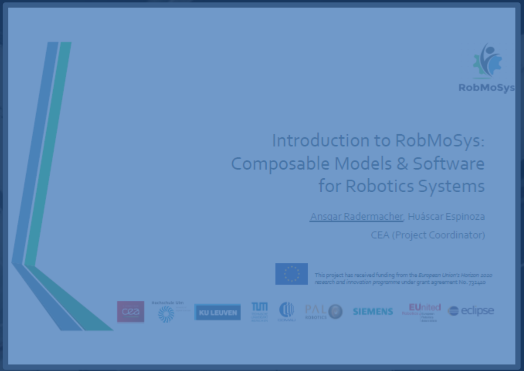
X *Model conforms to X (meta-model or SW library)*

Ambition of Creating Models


RobMoSys
Models at Run-time e.g. for self-configuration/adaptation/explanation
5
Automate Labor-intensive Activities (V&V, code gen., etc.)
4
Correct-by-construction Composition (Predictability)
3
Reuse and Customization of Robotics Software Assets
2
Guidance by Following Harmonized Interpretation of Abstractions
1
Capability Levels of RobMoSys Solutions
Complexity
Ecosystem Life

Les Initiatives en cours – Deux exemples

- Les approches « Model-Driven » : le projet Européen RobMoSys



RobMoSys

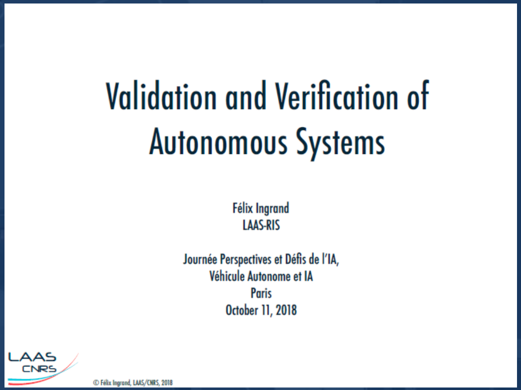
Introduction to RobMoSys:
Composable Models & Software
for Robotics Systems

Ansgar Radermacher, Huáscar Espinoza

CEA (Project Coordinator)

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No. 732410

- Les travaux de Vérification / Validation : Propositions LAAS



Validation and Verification of
Autonomous Systems

Félix Ingrand
LAAS-RIS

Journée Perspectives et Défis de l'IA,
Véhicule Autonome et IA
Paris
October 11, 2018

LAAS CNRS

© Félix Ingrand, LAAS/CNRS, 2018

- Orientations : Propositions

Software Validation and Verification

- Require formal models and mathematically/logically sound "checking" techniques
  - formal models (e.g., FSM, IO automata, Petri nets, timed automata, situation calculus, synchronous systems, etc)
  - checking by reachable state exploration (e.g. model checking), logical induction (e.g. theorem proving, sat solving, etc) or runtime verification
  - complete methods, over approximation, statistical methods, etc...

V&V models: Different situations over a complete autonomous system

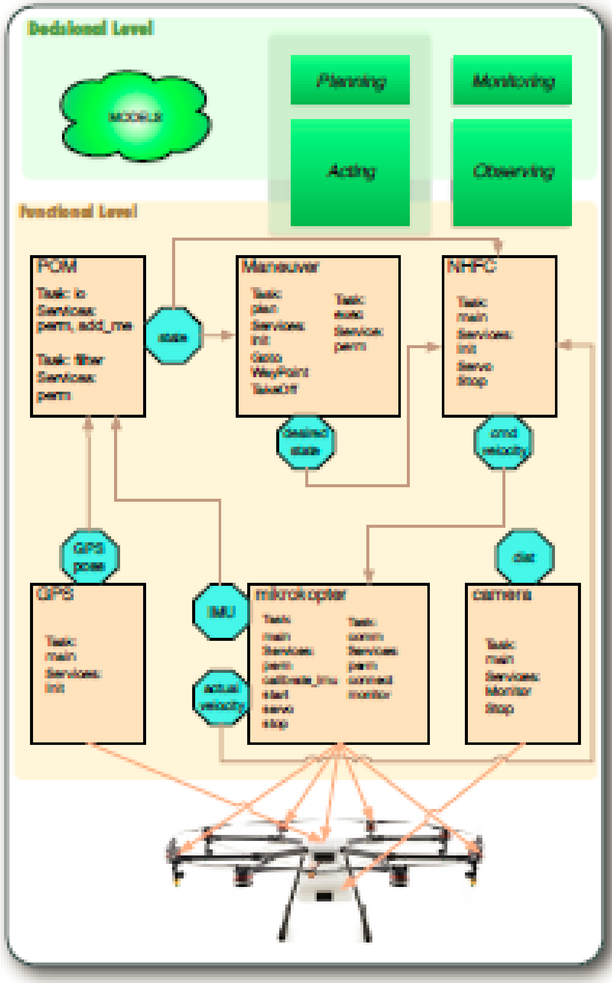
Formal models: decisional : planning (e.g. UPPAAL, model checking), monitoring, FDIR, observing

Learned models: Reinforcement learning models, perception models, etc.

Specification models: Software engineering models: e.g. GenoM3, Oroccos, MAUVE, RobotML, etc.

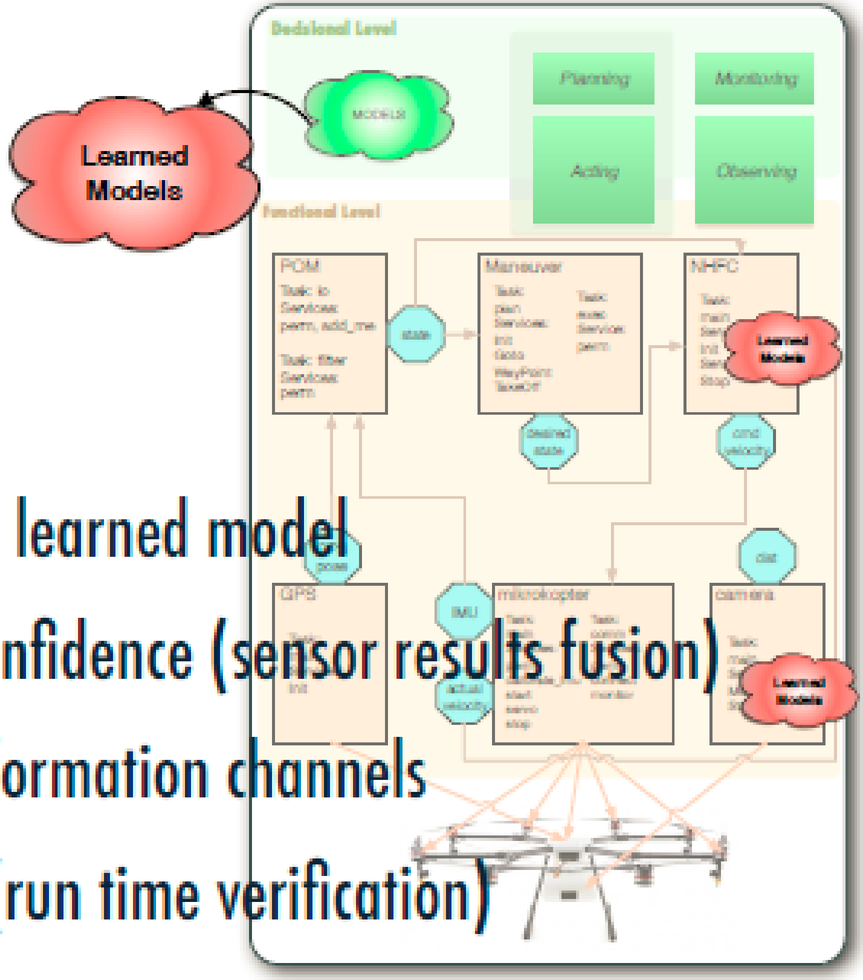
Programming directly the Model: Orccad, Scade, etc.

No Model…



V&V of learned models…

- Machine learning is the new AI…
- Hard to extract a formal model… but we should try
- Proper environment modeling
- Properly characterize the bound of the learned model
- Use multiple sources to improve the confidence (sensor results fusion)
- Consistency checking over different information channels
- Safety bag around these components (run time verification)



[1] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete Problems in AI Safety," arXiv.org, 1606.06565v2, vol. cs.AI. 21-Jun-2016. http://arxiv.org/abs/1606.06565v2

[2] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Towards Verified Artificial Intelligence," arXiv.org, 1606.08514v3 vol. cs.AI. 28-Jun-2016. http://arxiv.org/abs/1606.08514v3

Conclusion

- When there are models... there is hope!
- Adapt the model and the V&V techniques
- Try to keep the overall consistency
- AI components are mostly OK (wrt formal V&V)
- V&V and certification of "learned model" based components remain a challenge

Des axes de recherche*
Point de vue : Vérification, Validation et Certification

Research question 1

How can **knowledge-driven** (“affordance-based”)**robot programming, perception and learning** be made more **realtime**, while still taking into account more **prior knowledge** about the tasks, the robots, the objects they interact with, and the environment they have to survive in?

Research question 3

Which new design paradigm can provide **cheap, light and safe** (hence, “lousy”) robot hardware? This is a necessary evolution before **robotics platforms** can become a**commodity**.

Research question 5

What is the **essential and minimal structure** to model the **functional aspects** of robotic systems

Research question 2

What are the formal Domain Specific Languages (DSLs) that can make the **knowledge representation** (and hence the **programming** of robots) a lot more easy? And, at the same time, a lot more *semantically consistent*, and (hence!) *deterministic*, and (hence?) *verifiable*, and (hence!) **certifiable**, and (hence?) societally **trustworthy**.

Research question 4

What is the **essential and minimal structure** to model the **software aspects** of robotic systems? How should **robot control software** be developed in the future? What **architectural patterns** can help us cope with the **exploding complexity** in knowledge, task variations, and distribution over several sub-systems?

Research driver 1

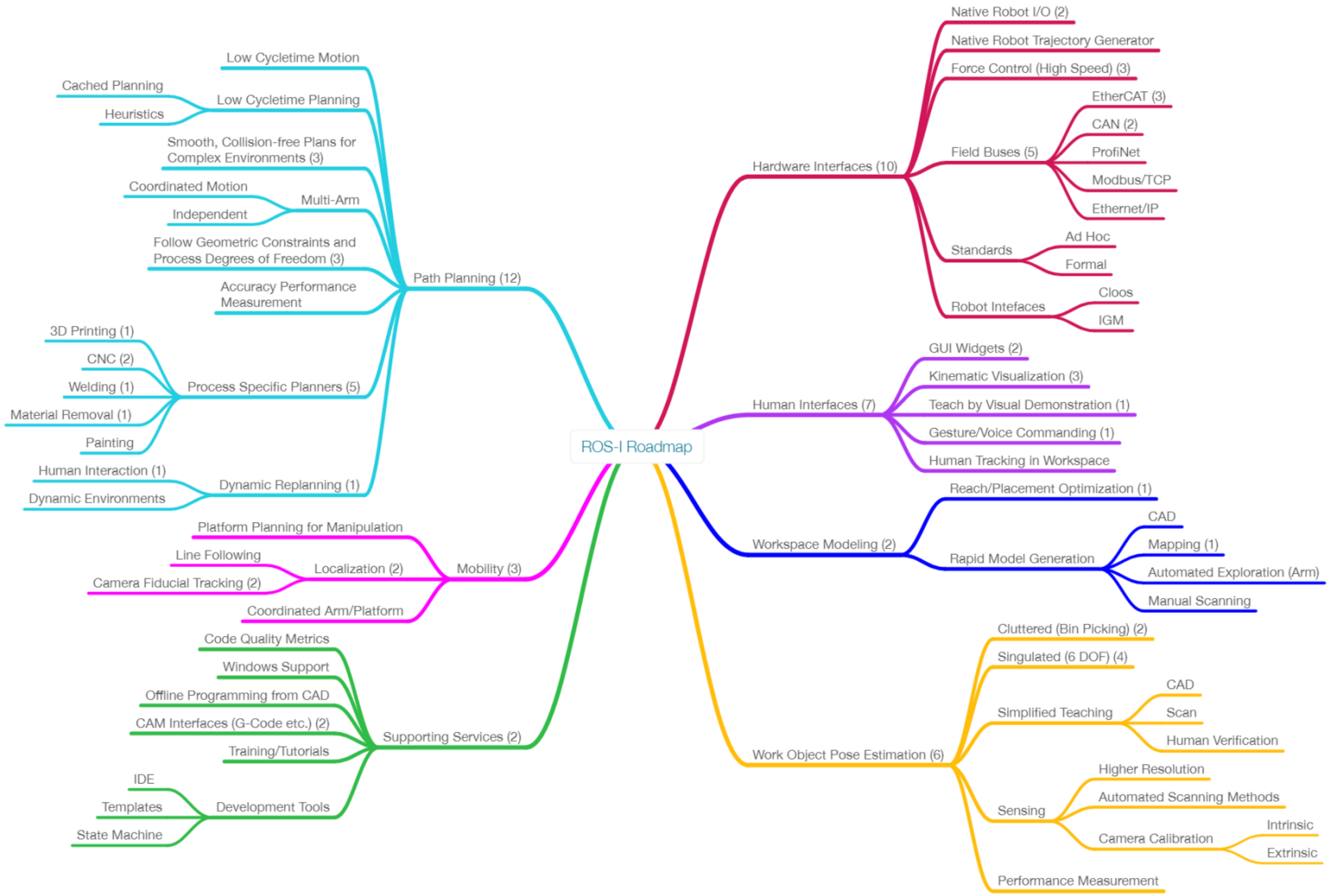
The societal expectation to have access to (only) trustworthy robots

Research driver 2

The societal expectation to have trustworthy Artificial Intelligence

* *Origine : Herman Bruyninckx*

Des axes de recherche
et ROS ?



- ROS-I Roadmap
  - Path Planning (12)
    - Low Cycletime Motion
    - Low Cycletime Planning
      - Cached Planning
      - Heuristics
    - Smooth, Collision-free Plans for Complex Environments (3)
    - Multi-Arm
      - Coordinated Motion
      - Independent
    - Follow Geometric Constraints and Process Degrees of Freedom (3)
    - Accuracy Performance Measurement
    - Process Specific Planners (5)
      - 3D Printing (1)
      - CNC (2)
      - Welding (1)
      - Material Removal (1)
      - Painting
    - Dynamic Replanning (1)
      - Human Interaction (1)
      - Dynamic Environments
  - Mobility (3)
    - Platform Planning for Manipulation
    - Localization (2)
      - Line Following
      - Camera Fiducial Tracking (2)
    - Coordinated Arm/Platform
  - Supporting Services (2)
    - Code Quality Metrics
    - Windows Support
    - Offline Programming from CAD
    - CAM Interfaces (G-Code etc.) (2)
    - Training/Tutorials
    - Development Tools
      - IDE
      - Templates
      - State Machine
  - Hardware Interfaces (10)
    - Native Robot I/O (2)
    - Native Robot Trajectory Generator
    - Force Control (High Speed) (3)
    - Field Buses (5)
      - EtherCAT (3)
      - CAN (2)
      - ProfiNet
      - Modbus/TCP
      - Ethernet/IP
    - Standards
      - Ad Hoc
      - Formal
    - Robot Intefaces
      - Cloos
      - IGM
  - Human Interfaces (7)
    - GUI Widgets (2)
    - Kinematic Visualization (3)
    - Teach by Visual Demonstration (1)
    - Gesture/Voice Commanding (1)
    - Human Tracking in Workspace
  - Workspace Modeling (2)
    - Reach/Placement Optimization (1)
    - Rapid Model Generation
      - CAD
      - Mapping (1)
      - Automated Exploration (Arm)
      - Manual Scanning
  - Work Object Pose Estimation (6)
    - Cluttered (Bin Picking) (2)
    - Singulated (6 DOF) (4)
    - Simplified Teaching
      - CAD
      - Scan
      - Human Verification
    - Sensing
      - Higher Resolution
      - Automated Scanning Methods
      - Camera Calibration
        - Intrinsic
        - Extrinsic
    - Performance Measurement

Version de 2014 la plus récente (?)

Agenda

- Vérification, Validation et Certification
- Quelques Chiffres
- Le Monde Aéronautique
- Le Monde Automobile
- Et le Monde Robotique ?
- *Une conclusion ?*

Tentative de conclusion générale

- La sécurité implique des développements couteux et long
- Des hauts niveaux de sécurité impliquent une séparation stricte des rôles
- Des hauts niveaux de sécurité impliquent la maîtrise de toute la pile logicielle (OS/Middleware/Composants)

- La montée en puissance des voitures autonomes permet d'envisager une réduction des couts des processus de développement
  - Montée en puissance des approches modèles
  - Montée en puissance des générateurs de code

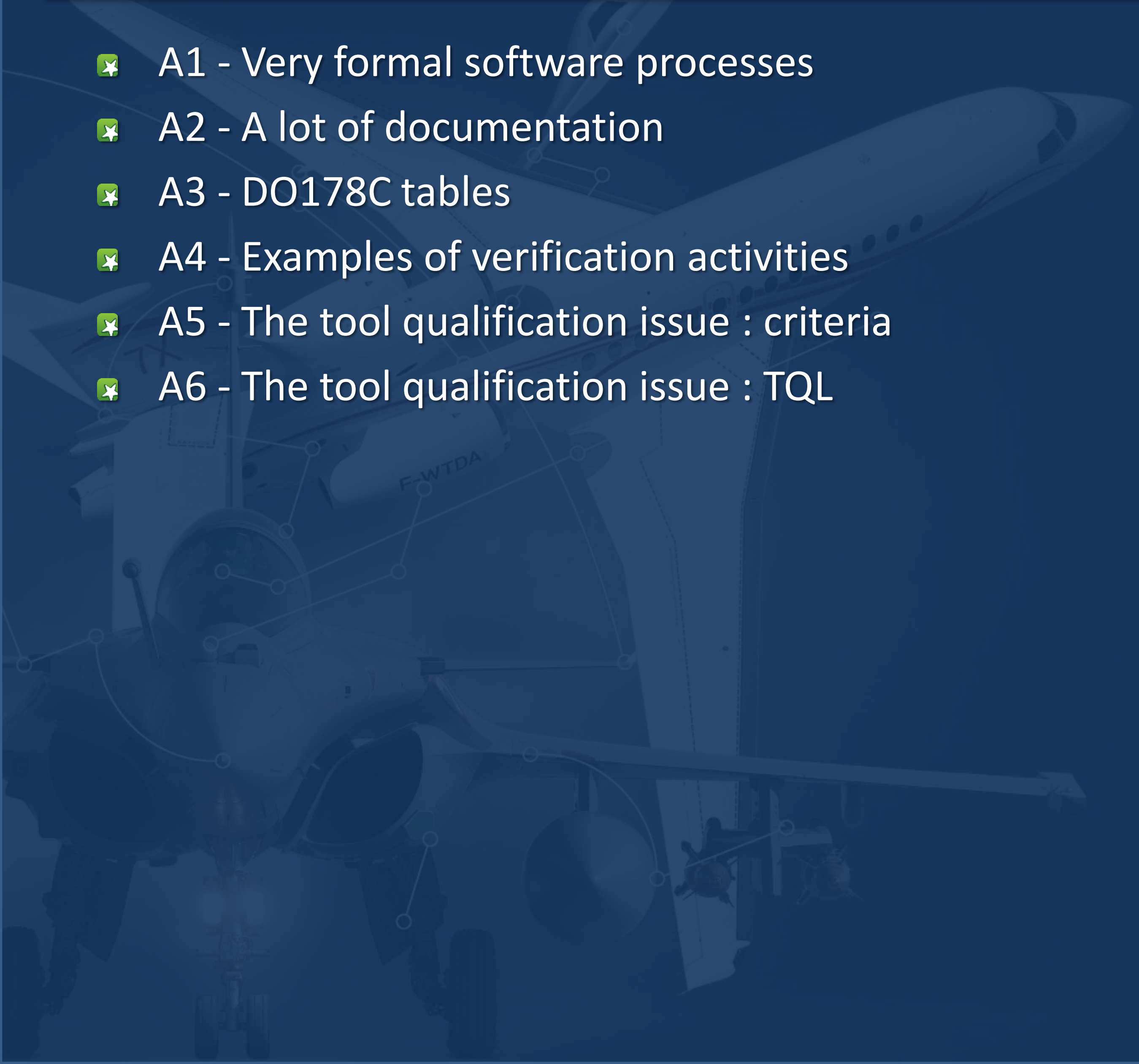
- L'approche modèle favorise la montée en puissance des middleware

- Le développement de ROS n'apporte pas à ce jour les garanties nécessaires
- Les aspects Vérification et Validation doivent être considérés par les travaux de recherche dès l'origine pour favoriser leur utilisation
- Lors de l'étude d'une nouvelle méthode de contrôle ou autre il est nécessaire de réfléchir dès l'origine aux moyens de sa vérification et de son contrôle

Quelques questions ?

Annexes pour le curieux

- A1 - Very formal software processes
- A2 - A lot of documentation
- A3 - DO178C tables
- A4 - Examples of verification activities
- A5 - The tool qualification issue : criteria
- A6 - The tool qualification issue : TQL



A1 - Very formal software processes

- DO178C considers different types of software processes :
  - The **software development** processes (**requirements, desing, coding, integration**) that are defined in the Software Development Plan (**SDP**).
  - The integral processes that ensure the correctness and control of, and confidence in the software life cycle processes and their outputs:
    - The **verification** process that is defined in the Software Verification Plan (**SVP**)
    - The **Configuration Management** that is defined in the Software Configuration Management Plan (**SCMP**)
    - The **Software Quality Assurance** that is defined in the Software Quality Assurance Plan (**SQAP**)
    - The **certification liaison** process that is defined in the Plan for Software Aspect of Certification (**PSAC**)

A2 - A lot of documentation :
main outputs of the software processes

- Software plans : PSAC, SDP, SVP, SCMP, SQAP
- Software standards : software requirements, design, code
- Software specification (HLR)
- Software design including architecture (LLR)
- Source code
- Executable Object Code (EOC)
- Software test cases and procedures
- Software verification results (minutes of review, tests report, …)
- Software Configuration Index (SCI)
- Software Configuration Management records
- Software Configuration Index (SCI)
- Software life cycle Environment Configuration Index (SECI)
- Problem Report (PR)
- Software Quality Assurance records
- Software Accomplishment Summary (SAS)

A3 - DO178C tables

- DO178C defines software objectives thanks to 10 tables (see appendix A) that depend on the software IDAL.
  - table A-1 : Software planning process Some of the expected products
  - table A-2 : Software development processes
  - table A-3 : Verification of outputs of Software requirements process
  - table A-4 : Verification of outputs of Software design process
  - table A-5 : Verification of outputs of Software coding process & integration process
  - table A-6 : Verification of outputs of Software integration process
  - table A-7 : Verification of outputs of verification results
  - table A-8 : Software configuration management process
  - table A-9 : Software quality Assurance process
  - table A-10 : Certification liaison process

A4 - Examples of verification activities

- The WCET (Worst Case Execution Time) shall be defined as the verification of the stack usage.
- MC/DC structural coverage shall achieved for the DAL A software items.
- The data flow coverage
- The control flow coverage
- The integration tests
- The robustness test cases :

    a) Real and integer variables should be exercised using equivalence class selection of invalid values.

    b) System initialization should be exercised during abnormal conditions.

    c) The possible failure modes of the incoming data should be determined, especially complex, digital data strings from an external system.

    d) For loops where the loop count is a computed value, test cases should be developed to attempt to compute out-of-range loop count values, and thus demonstrate the robustness of the loop-related code.

    e) A check should be made to ensure that protection mechanisms for exceeded frame times respond correctly.

    f) For time-related functions, such as filters, integrators, and delays, test cases should be developed for arithmetic overflow protection mechanisms.

    g) For state transitions, test cases should be developed to provoke transitions that are not allowed by the software requirements.

DASSAULT AVIATION

A5 - Configuration Management issue

- **All the outputs** of the DO178C activities **shall be managed in configuration** depending on the of the software item DAL. Most of the time, the configuration management is performed thanks a Configuration Management tool such as Synergy, …

- The configuration level of the software items depends on the DAL and the software item.

- The Baselines (The approved, recorded configuration of one or more configuration items) shall be defined.

- The modifications of some configurations items shall be defined thanks to formalized Problem Reports (PR). Most of the time, the PR are during a Change Control Board (CCB)

A6 - The tool qualification issue :
The tool criteria

- Software Tools Classification
  - **Criteria 1 (Development Tool) :** A tool whose output is part of the resulting software and thus could insert an error. Example : code generator
  - **Criteria 2** : A tool that automates verification process(es) and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of:
    - Verification process(es) other than that automated by the tool,(Example, an analysis tool that is able to detect overflow/divisionby0/stackoverflow by analysis and the results of which are used to eliminate overflow/divisionby0/stackoverflow tests) or
    - Development process(es) (which could have an impact on the resulting software).
  - **Criteria 3 (Verification Tool)** : A tool that, within the scope of its intended use, could fail to detect an error.

A7 - The tool qualification issue :
The Tool Qualification Level (TQL)

- The tool qualification objectives are defined thanks the tool criteria and the software Assurance Level.

DAL	criteria		
	1	2	3
A	TQL-1	TQL-4	TQL-5
B	TQL-2	TQL-4	TQL-5
C	TQL-3	TQL-5	TQL-5
D	TQL-4	TQL-5	TQL-5