EC2 N18

Methodological Guideline for AI-based System Design at Operational and System Level: System Approach

L2.3.2.3

L2.4.3.3

L2.4.4.3

L2.4.8.3

contact@confiance.ai | www.confiance.ai

Document reference: 218B

Contributors

	Name	Organisation	Role
Responsible for the deliverable	Kevin Mantissa	IRT SystemX	Research Engineer
Scientific responsible	Christophe Bohn	IRT SystemX	Technical Coordinator
Co-authors	Christophe Bohn	IRT SystemX	Technical Coordinator

Document control

Revision	Date	Commentary	Author
0.1	22.09.2023	Creation	Kevin Mantissa
0.2	29.11.2023	Update to new template	Kevin Mantissa
0.3	21.12.2023	Update for review	Kevin Mantissa
1.0	24.01.2024	Update following comments for final delivery	Kevin Mantissa, Christophe Bohn

Table of contents

A. Introduction and abstract

A.1 General introduction to trustworthy AI challenges

Trustworthiness in AI within critical systems (systems that can directly or indirectly affect human life and moral entities) is essential for its widespread adoption (by the industry, the decision makers, the general public, etc.) and poses the following significant challenges.

- First, how to design AI models, so that, by construction, they satisfy trustworthy properties (accuracy, robustness…).
- Secondly, how to characterize these AI models, for example to understand and explain their behavior and their adequacy to the operational domain.
- Then, how to implement and embed those AI models on hardware, by making them fit for the target without losing their trustworthy properties.
- Another question is, what methods of data engineering to apply in order to, among other topics, manage important volumes of data and adapt to the evolution of the operational domain.
- At system level, what verification and certification processes to consider specifically for AI-based systems.
- Finally, a federation of all these matters is necessary to build an end-to-end methodological approach, supported by a consistent engineering environment compatible with industrial practices.

These are the challenges, among others, that the Confiance.ai program addresses.

A.2 Context of the methodology

The quick progress of Artificial Intelligence systems has highlighted several challenges that the community is currently facing. Among those challenges, some of them illustrate the limits of current System Engineering methods and best practices, that are fit for Conventional Systems, but are not able to keep up with the rapid evolutions of AI-based systems and the peculiarities of such systems.

Working Groups for Standardization and Engineering Systems try to keep up to date with these evolutions by updating old System Engineering standards or releasing new ones with AI in mind, e.g. ISO 15288:2023 (*ISO/IEC/IEEE 15288*, 2023) or ISO 5338 (*ISO/IEC DIS 5338*, 2023).

However, what is needed to better apprehend AI from a System Engineering viewpoint is to revisit the operational and system activities formerly applied for Conventional Systems, and evaluates how Artificial Intelligence impacts those activities.

While major actors of the AI field, in the tech industry, try to incorporate some System Engineering framework into their AI Engineering, the approach described in this deliverable is at the other end. We aim to incorporate the new methods associated with AI development into the conservative System Engineering methods. In this manner, we can assess the impacts of AI on conventional methods, as well as precise the boundaries of AI engineering that are not as well-defined as other fields, due to its novelty.

In Confiance.AI context, this work on System Engineering of AI-based systems is the way to increase trustworthiness by applying on AI mainstream industrial processes. This allows to make AI more compliant with Quality Assurance (including validation) and Safety Assurance.

A.3 Purpose of the methodology

A.3.1 A two-step approach

The work presented in this document is built upon the notions of Intended Purpose and Design Intent. According to the Artificial Intelligence Act, first drafted in 2021 (AI Act Draft 2021, 2021) and updated pending validation in 2023 (AI Act Draft 2023, 2023), Intended Purpose means *"the use for which an AI system is intended by the provider, including the specific context and conditions of use, as specified in the information supplied by the provider in the instructions for use, promotional or sales materials and statements, as well as in the technical documentation"*.

Intended Purpose is a part of the Operational specification of an AI-based system. It could be considered as a global introduction before the analysis of stakeholders needs and lifecycle phases.

Design Intent describes design activities to address the needs synthesis at system level and translate it in a technical way in order to make it usable without ambiguity for ML and data engineering.

Both notions of Intended Purpose and Design Intent are part of architecting activities.

The aim of the following work is to guarantee the traceability of design and architectural choices at operational and system levels for AI-based systems. We focus on AI-based systems that rely on machine learning.

To address system engineering of AI-based systems, we split the methodology into two approaches:

- An Operational Approach, that aims to identify and characterize operational needs which can only be managed by an AI-based system (could not be reached by a conventional software)
- A System Approach, that aims to gather system-level artifacts that are required for the AI-component implementation and the coverage of operational needs

To simplify the following work, assumptions are made and explained in section A.10.

This document will focus on and detail the System Approach only. For the Operational Approach, please refer to the corresponding deliverable in 218A (Kevin Mantissa & Christophe Bohn, 2024). We strongly advise the readers to finish the Operational Approach document before reading the System Approach deliverable. Indeed, both deliverables have to be considered as a whole.

A.3.2 Rationale for the System Approach

Similarly, to the Operational Approach evoked in deliverable 218A (Kevin Mantissa & Christophe Bohn, 2024), the System Approach is also impacted by AI-based systems. It inherits the novelties from the Operational Approach and capitalizes on it. In this approach, we aim to identify technical elements to translate the need from the Operational Specification for AI-based systems into technical requirements in the System Specification. The technical elements are used to describe conditions and behaviors for the

AI-based systems. For instance, in vision-based AI agents, it is essential to distinguish what humans see from what a system/sensor perceives of the world.

The learning of AI-based systems aims to achieve automatically a correlation between the inputs and the outputs of the transfer function. This helps to solve questions that would be left unanswered otherwise with the conventional method: How inform an AI component that the image is an input and the annotation is an output? How do we explain that we want to correlate with the pixels for image-based systems?

The System Approach aims to be performed before the ML and Data Specification activities, as it will provide necessary technical elements to help perform these activities.

A.4 Confiance.AI scientific challenges addressed by the document

The deliverable addresses the following scientific challenges of Confiance.AI:

- Establish a methodology for defining the desired behavior of the trusted AI system

Besides, this deliverable also addresses questions inspired from deadlocks identified in EC2 project:

- **VR05** – How to design AI-based systems?
- **VR04** – How to properly describe operational specification regarding AI components issues?
- **VR11** – How to design AI-based systems that are User-Experienced oriented (human factor)?
- **VR19** – How to ensure the link with Functional Safety? (Quality Assurance part: we indicate the activities to perform, then we verify that those expected activities have been indeed performed)

A.5 Target audience

This deliverable (218B) and the deliverable dedicated to the Operational Approach (218A (Kevin Mantissa & Christophe Bohn, 2024)) target all types of profiles of the Confiance.AI program. It includes profiles with a System Engineering background. Indeed, with these approaches, the aim is to study how to appropriately incorporate AI in Conventional System Engineering methods and industrial context. Besides, it includes data and AI engineers as well, in order to give them a better understanding of what Operational and System Approaches could bring to AI implementation. This will empower them to absorb a global system view which encompasses the data viewpoint and the learning viewpoint. This approach supports the ability of the results to match with the needs, which is a condition for trustworthiness.

A.6 Glossary

Term	Signification
AEB	Advanced Emergency Braking
AI	Artificial Intelligence
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers

ISO	International Organization for Standardization
KPI	Key Performance Indicator
ML	Machine Learning
ODD	Operational Design Domain
QCDP	Quality Cost Delivery People
RGB	Red Green Blue
SEBoK	System Engineering Book of Knowledge
SOTIF	Safety of the Intended Functionality
TJC	Traffic Jam Chauffeur

A.7 Summary of limitations and perspectives

The efforts put in this document are focused in the formalization of encountered issues regarding AI-based System Approach with leads to solve them as a methodological guideline, instead of providing applicable solutions. In consequence, this work could serve as a basis for future works, such as:

- Multidisciplinary collaborative works on each part of this document with various engineer profiles, like system engineering, data engineering, machine learning engineering, quality assurance, etc.
- Proposals of an engineering process for System Approach detailing activities and tools
- Application of the detailed approach on an industrial use case

A.8 Document organization

In part 1, we explain what roles the inputs, outputs and Transfer Function play in the System Specification for AI-based systems.

In part 2, we indicate the correlation between the inputs and the outputs shall be monitored as it reveals particular behaviors to keep under control. To perform a correlation thanks to the transfer function, it is mandatory to bring the knowledge through the inputs and the outputs, that reveal four high-level classes of data.

In part 3, we talk about the role of sensors to capture a description of the real world for AI-based systems.

The other topics (4 to 7) are not developed, although we have identified them as key elements of this System Approach. However, pending future works that can touch upon this topic, for each of them we refer to existing work in the Confiance.AI program that provided some insights on these topics.

A.9 How to use the document

This document is a guideline rather than a proper process (that could result from future works).

In the Operational Approach, we give key elements that should be considered to realize an operational specification in order to identify the synthesis of the needs in preparation for its implementation in a given AI-based system.

Between the synthesis of needs and the implementation, the System Approach detailed in 218B will achieve the translation of operational needs into technical elements in the System Specification.

That is why the whole scope of system activities for AI-based systems engineering is covered by both deliverables.

A.10 Assumptions regarding this deliverable

In order to clarify the methodology explanation, we make the following assumptions but the resulting method can be generalized outside of these hypotheses:

- We consider an industrial project with clear milestones and clear goals defined by the product team:
  - The detailed activities are related to industrial project objectives with QCDP (Quality, Cost, Delivery, People) engagement,
  - The considered process assumes that the operational objectives can be achieved thanks to preliminary studies results,
  - Resulting industrial process aims to guarantee the achievement of the project, in accordance with its objectives
- In consequence, no preliminary work and no specific iteration is considered in our approach.
- The studied activities focus on design and architectural aspects: architectural choices and renunciation/opportunities (trade-offs).
  Detailed specifications, data specifications, datasets and testing activities should be considered out of scope of this activity.
- In accordance with Confiance.ai scope, the studied system is a critical industrial system. This kind of system requires a clear development process to grant quality assurance and safety.
- We focus on subsystems which can't be implemented by a conventional software development, and require AI-based implementation, as described in section A.3.2.

B. Part 1 - Engineering process comparison between conventional systems and AI-based systems

Introduction to part 1

The whole philosophy of this deliverable, along with the deliverable 218A on Operational Approach, is to integrate AI into a System Engineering culture. Indeed, we benefit from a large set of academic works and best practices that have been developed over the decades. This is particularly useful as we can check how such practices can provide new insights on the development of AI-based systems, and how AI can impact those practices.

In this context, we take inspiration from the vision of System Engineering on the decomposition of system-under-interest into functions, with their associated inputs and outputs, to suggest an adaptation that is consistent with the new stakes brought by AI-based systems.

Overview of part 1

In this section:

- We first introduce some definition related to the System Approach, and we describe the position of this approach within the System Engineering Process
- Then, we give an overview of the role played by the notion of transfer function in conventional System Engineering
- Afterwards, we give an overview of the changes that appear in AI-based System Engineering
- Finally, we go over the specific engineering item that are needed for each approach

B.1 Definitions & Position of System Approach in System Engineering process

B.1.1 Definitions

For the proper understanding of the following activities in this deliverable, we share the following definitions which are aligned with the philosophy of our System Approach. Those definitions rely on existing standards and academic references, such as the ISO 15288 standard or the System Engineering Book of Knowledge (*SEBoK*, 2023).

Design:

- *[process] to define the architecture, system elements, interfaces, and other characteristics of a system or system element (First definition)* (*ISO/IEC/IEEE 24765*, 2017)

- *"Specification of system elements and their relationships, that are sufficiently complete to support a compliant implementation of the architecture"* (*ISO/IEC/IEEE 15288*, 2023)

System Design:

- *"process of defining the hardware and Software architecture, components, modules, interfaces and data for a system to satisfy specified requirements"* (*ISO/IEC/IEEE 24765*, 2017)
- NB: System Design aims to express in technical terms the expected behavior to be considered for implementation. This means decomposing the system issue into smaller and more manageable issues

System Specification:

- *"documented set of mandatory requirements for a system"* (*ISO/IEC/IEEE 24765*, 2017)
- That we can consider as a document that integrates requirements related to the studied system

System requirements definition process:

- A system requirement definition process is a statement that *"transforms the stakeholder, user-oriented view of desired capabilities into a technical view, of a solution that meets the operational needs of the user. It creates a set of measurable system requirements that specify, from the supplier's perspective, what characteristics, attributes, functional and performance requirements the system is to possess, in order to satisfy stakeholder requirements"* ((*ISO/IEC/IEEE 15288*, 2023) section 6.4.3.1)*;*

The activities developed thanks to the System Approach are expected to be compiled into a deliverable that will serve the role of System Specification.

B.1.2 Position of System Approach in the System Engineering process

The System Approach goes hand in hand with the Operational Approach in deliverable 218A.

The reference System Engineering approach we will refer to in this deliverable is from the System Engineering Book of Knowledge (*SEBoK*, 2023). Using Figure 1 from SEBoK (Rick Adcock, 2023), we illustrate how our activities could combine Operational and System approaches similarly to the SEBoK.

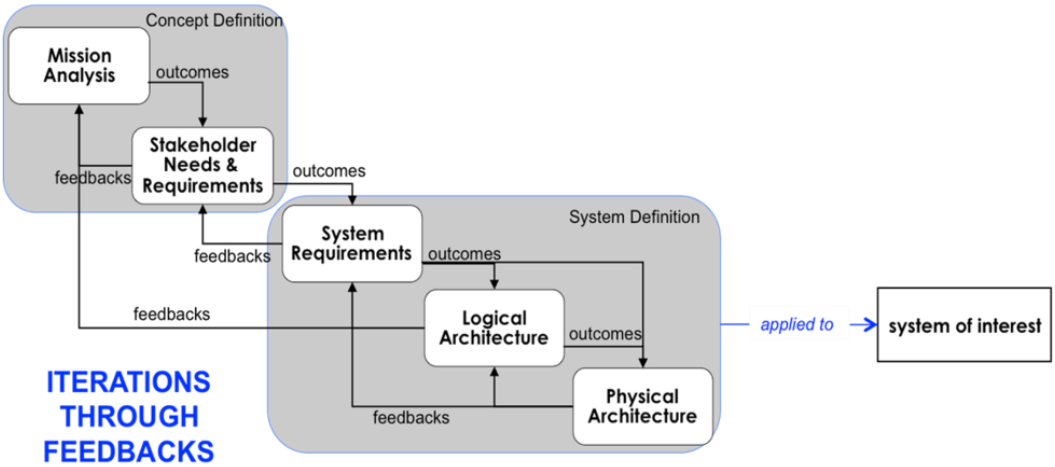


Figure 1 Example of iterations of Processed related to System Definition, from Faisandier 2012

Compared to this figure, we want to highlight several topics:

- While it is expected for AI-based systems to be subjects to iterations during the phases of their life cycle, we should aim to perform for a specification the most satisfactory possible from the get-go and not be over-reliant on iterations. It is expected that AI-based system design does not consider the use of iterations as a nominal pattern of development, but rather as a way to amend conflicts, oversights or emerging defects;
- More than just defining the System, we expect the System Approach to assist in designing AI-based systems.

These two points illustrate how the fully-matured System Approach may diverge from the vision above from the SEBoK. However, they still share interesting similarities.

B.2 Analysis of conventional system engineering

B.2.1 Definition of a transfer function

Every engineering process can be assimilated to a transformation, with expected inputs and outputs. From a mathematical viewpoint, a transformation F applied on an input “I” leads to a given output “O”. Depending on the nature of the transformation (or process), an output O could have more than one input and a single input.

A transformation is performed by a function, that results from the decomposition of a given system into subfunctions. It enables to describe what the system can do. We generally present a function in the format illustrated by Figure 2:

Figure 2 Illustration of a Transfer Function with inputs and outputs

In Conventional System Engineering, we already possess the methods and tools enabling the description of such functions. These methods rely on the fact that a system function can be decomposed into subfunctions iteratively until the level of software functions that could be implemented.

However, additional challenges appear for AI-based systems, as we may know the nature of the function but not the detail of its implementation. Thus, with this activity, we aim to explore the main differences between AI engineering and Conventional System Engineering on this particular topic, in order to propose a way to specify system design of AI-based systems.

B.2.2 Decomposition into functions for conventional System Engineering

The first step to deploy the System Engineering process is to define the transformation that the System Under Interest will perform. This transformation, called ∑, results in the following expression:

$$O = \sum(I)$$

Where I is the input and O is the output that addresses one or several requirements.

This first decomposition can be further refined to design all operations required to perform the transformation from “I” to “O”.

To design the overall transformation, we:

- refine “∑” into sub-functions: f_1, f_2, f_3, … f_n;
- describe f_i to f_j interactions;
- refine until we reach the lower level of refinement possible.

Figure 3 synthetizes the expected decomposition from the initial Transfer function of our System Under Interest to its decomposition into sub-functions with associated inputs and outputs.

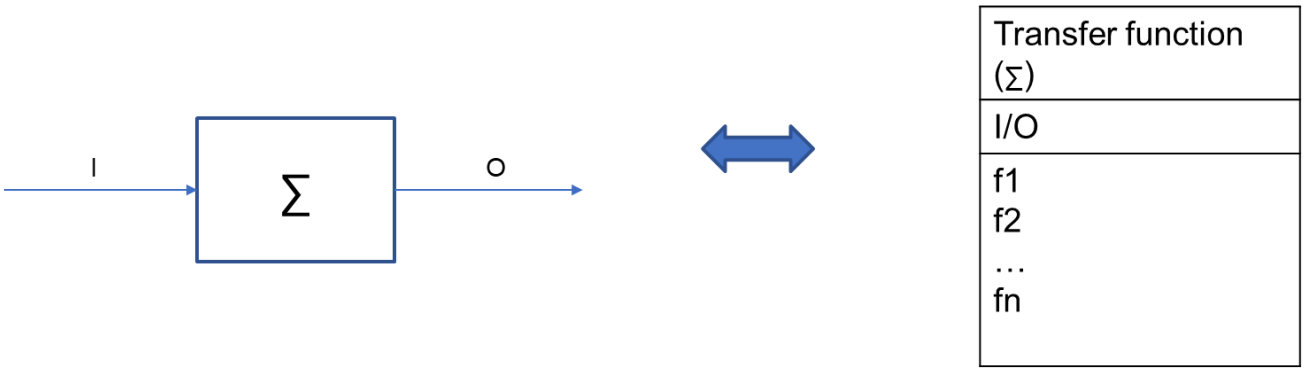


Figure 3 Decomposition of a Transfer function with their inputs, outputs and related sub-functions

The conventional approach to system engineering relies on the segmentations of the task, completed by several functions. The system addresses several operational objectives and therefore, the functions enables the system to fulfill these objectives.

From an external viewpoint, the function takes inputs and produce outputs. If the system has been specified and verified, the relationship between inputs and outputs are well defined and well known.

Thus, with this vision, a System is considered as an organization of functions and sub-functions that apply transformations on inputs to deliver outputs. Those functions are represented through the use of logical architectures and allocated to physical elements thanks to physical architectures. They are captured with the formalization of Functional requirements, which identify at a given level (system, function, sub-functions, etc.) the expectations to satisfy in order to realize the functions.

Functions are refined into Sub-functions, each level having their own set of inputs, outputs and requirements. When they cannot be refined anymore, sub-functions are allocated to a given element (also called “System Element” using SEBoK vocabulary).

For each function, we can describe a set of inputs, outputs and requirements (relationships) between inputs and outputs. We can also description of the interfaces, describing the format of exchange for the I/O.

If we divide the function into sub-functions, each sub-function will have its own set of inputs, outputs and requirements. The outputs of sub-function may be the inputs to other sub-functions. If the decomposition

is adequate, the completion of each sub-function requirement ensures the completion of the function requirement.

In Conventional Software engineering, a function can be decomposed in a top-down fashion until we can produce the source code that answers to the defined expectations. And a bottom-up approach can enable infer the whole implementation logic between the main function of the system and the implemented code that answers to software functions.

B.3 Analysis of AI-based system engineering

At the boundaries of the AI component, it is possible to define a function. However, the main difference between Conventional System Engineering and AI-based System engineering is that it is not possible to go into the detail of the implementation, i.e. decompose into sub-functions inside the AI component. It is no longer possible to refine an AI-based element, due to the following reasons:

- The behavior of the function is “learnt” while running inferences with the inputs and outputs: It is not implemented as a code that we could trace back to a particular function
- The elements responsible for the transformation of the function cannot be internally described, e.g. in deep learning, the process leading from inputs to outputs is known, however, what happens exactly in each neuron and/or layer of the algorithm remains unknown

These reasons directly impact the System Engineering process applied to AI-based systems. When developing a System element with AI features, it is no longer possible:

- to refine and reduce the requirements;
- to split the task into sub-tasks (sub-functions): The AI-based system element is considered as a single black box;
- to explicitly describe the transfer function.

This is an incentive to opt for other methods. For instance, it should be noted that the behavior of the Transfer Function does not need to be specified for AI-based system development: the behavior is “learnt” on the basis of the inputs/outputs and can be traced back to the operational objectives / Intended Purpose of the system.

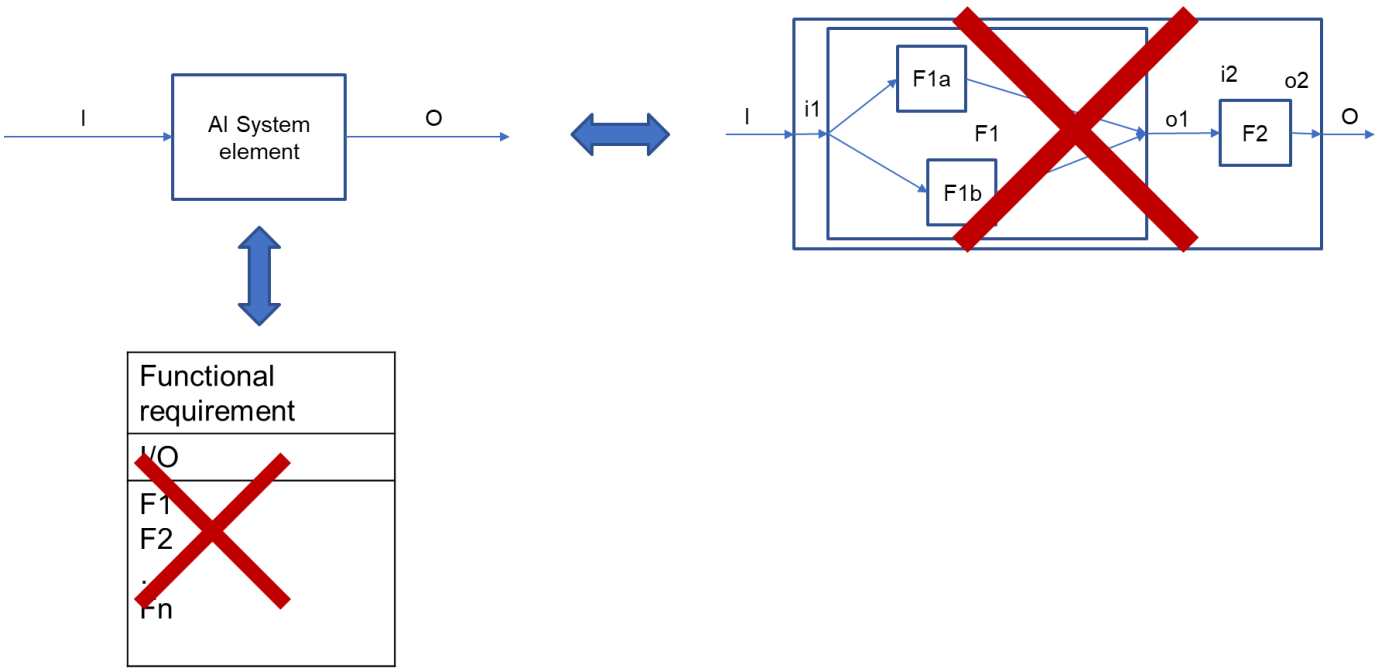


Figure 4 Incompatibility of AI with conventional functional decomposition

For an AI-based system, the Transfer function that drives the expected behavior of the system becomes a mean to achieve the design rather than the goal resulting from specification activities.

B.4 Engineering items for implementation of systems

System design is required both for conventional and AI-based systems. From conventional system methods and tools, we will establish parallels with the existing AI learning habits.

The AI based systems design will probably require new specific methods and tools. These new artefacts, exclusive to AI, shall be included in the future methodology for the design and development process.

At component level, we have the highest level of refinement. We have “small” input and output and a well-defined transfer function between them. To create the component in conventional Software Engineering, we usually need:

- a source code, produced by a human developer using a defined programming language
- When relevant, a compiler to create an executable binary (interpreted languages such as Python do not need compiler)

This binary may need a hardware component on which the binary code will run, or the code could directly be run using cloud-hosting services (although it will still need to be stored on physical servers, meaning that hardware components remain necessary). The hardware is a mandatory element and common ground with the conventional approach.

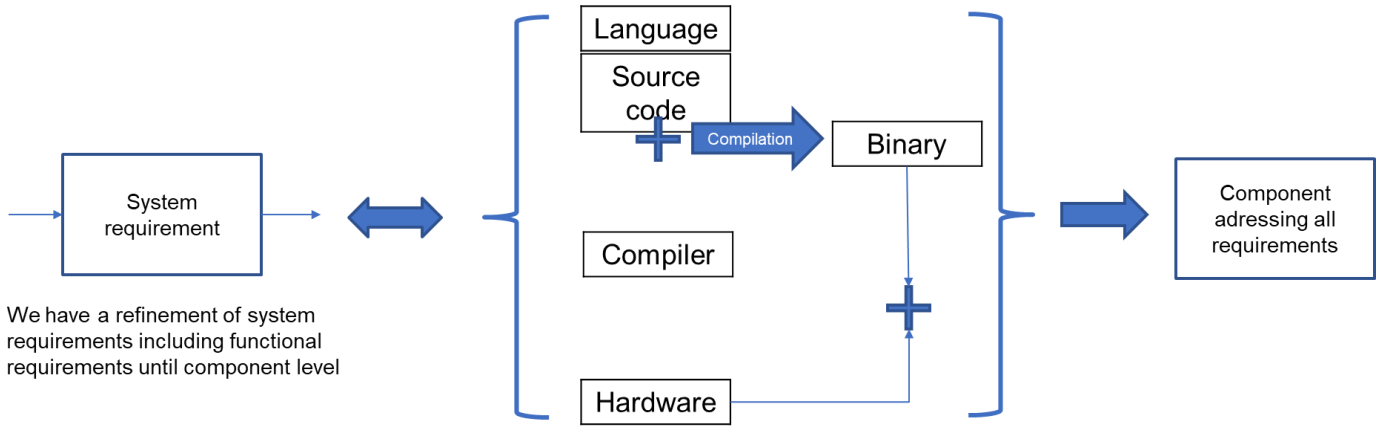


Figure 5 Refinement of System requirements in Conventional Software

However, in AI-based development, code and compilation are not the main expected items. While we still need a human developer to produce a source code using a given programming language, this code is used to specify the parameters and expected inputs and outputs of the learning of the AI-based system, i.e. the source code is used to achieve a given behavior. This behavior is learnt from a set of the following elements:

- Inputs: the elements the system is going to learn with
- Outputs: the expected behavior of the system (wanted and forbidden)
- The model (Machine Learning, Neural Network, etc.)
- The Hyperparameters to perform the learning
- The KPI and success criteria that will drive the learning and give objectives.

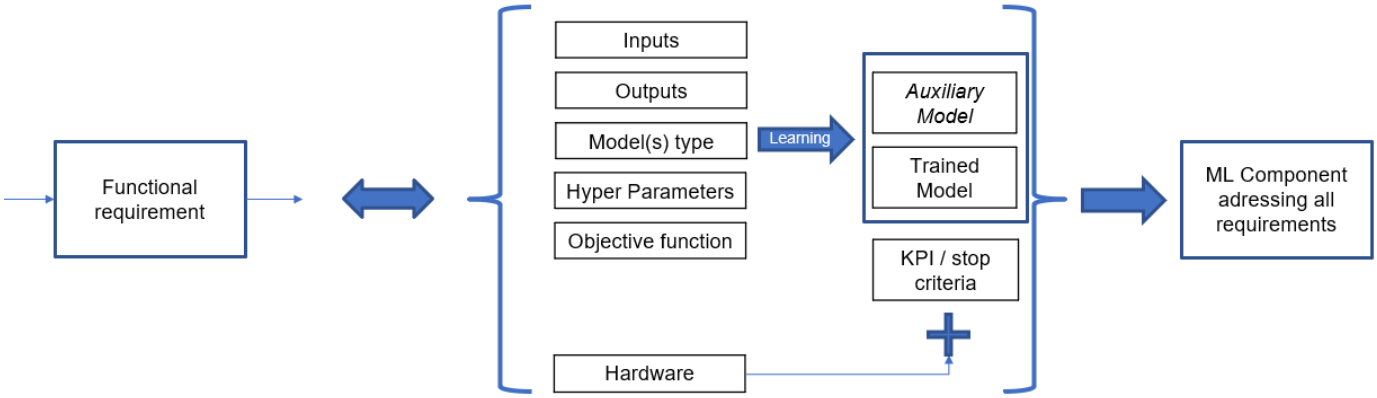


Figure 6 Refinement of Functional requirements in AI-based systems

The hardware remains a common artefact between the approaches, although we can expect to have more powerful components as training and operating AI-based systems is particularly demanding in resources.

This shows that for AI-based systems, a new set of tools to address from system design is expected in order to cover the system requirements.

B.5 Conclusion on Part 1

The development of AI-based systems highlights a change of paradigm regarding Conventional System Engineering: the former describes the way to learn the transfer function while the latter describes and refines the transfer function.

In Conventional Software, when we describe the transformation:

- System Engineers describe the transformation inside the function in order to perform the implementation;
- Inputs and outputs are considered as external interfaces of the system;
- System requirements are refined until they can be implemented.

Meanwhile, in AI-based system design, describing the expected components to learn the transformation means that we must specify from a technical viewpoint the following artefacts:

- inputs and outputs;
- model type and associated hyperparameters;
- objective function, KPIs and success criteria

System Specification has to change its paradigm in order to adapt to this learning method. It is required to describe the inputs and outputs of the function in order to learn the transformation. The Design Intent drives the learning methods to converge toward the expected behavior.

It is important to notice that system design specification will be impacted by:

- the choice between AI and conventional implementation;
- for AI-based systems, the model family (sensors are evoked in more details in section D below).

The fact the design of AI is learning-based and data-based means that the identification of technical elements is essential in order for an AI-based system to achieve a behavior that is compliant with its Intended Purpose. We focus in the next section on the nature of technical elements that are to be expected.

C. Part 2 - Types of technical elements to address the subfunction at operational level

Introduction to part 2

In section B, we presented how the emergence of AI-based systems induced a change in paradigm compared to conventional Engineering, regarding the achievement of operational needs through a transfer function. We shift from a conventional system engineering vision, where we specify the transfer function, to an AI-based system engineering vision, where we specify the means for an AI-based system to learn the transfer function. This implies that we identify the technical elements to enable an AI-based system to learn its transfer function.

For reminder, the System Approach do not touch upon the activities at the level of the AI component. It aims to translate the needs from the Operational Specification into technical requirements for the system, and comes before the step of Dataset Specification and ML Engineering. In this context, the technical elements that we consider in this section should be considered as high-level categories of data used to guide the way datasets should be built in coherence with the Operational Specification.

The elements to consider in the System Specification to guide the learning and testing activities in order to cover the Operational Specification will be evoked in the section E below (Part 4).

For instance, for an Advanced Emergency Braking (AEB) use case; one expected behavior is to brake to avoid pedestrian collision. The following questions should be answered at system level to achieve the correct dataset to answer the need:

- What is a pedestrian and how to detect them?
- What is the relevant situation of a pedestrian for the system at time of operation?
- What are the elements likely to mislead the detection (it looks like a pedestrian, but is not one of them)?

The choice of braking upon seeing a pedestrian waiting at a crosswalk for an automated vehicle is decided at operational level. The goal here is to determine how to do so.

Overview of part 2

In this section:

- We first evoke how to specify elements for the learning process,
- Then, we focus on the following decomposition of technical elements regarding:
  - Expected Environmental Conditions;
  - Harmful Environmental Conditions;
  - Required Intended Behavior;
  - Unwanted or Disturbing Behavior.

C.1 Specifying system technical elements in preparation for dataset specification

C.1.1 Technical elements for traceability between Operational Specification and Dataset Specification

The System Specification aims to ensure the traceability with the Operational Specification, by translating the operational needs into technical requirements. It has to cover at system level all design decisions that could impact implementations activities.

This methodological guideline gives a direction on what should be described.

The System Specification shall describe the following types of technical elements:

- The elements related to the environment in which the system will evolve;
- The elements related to the expected behavior that the system should learn.

Elements related to this environment could be considered as input of the transfer function realized by AI components. Elements relative to expected behavior could be considered as output of the transfer function realized by AI component. The principle of AI is to learn the transfer function by identifying the correlations between outputs and inputs. That is why it is important to describe precisely both categories before Data Specification activities.

For these two categories, it is useful to describe technical elements in a positive way (what we want) and in a negative way (what we fear). Especially, describing the unwanted situations or behaviors seems to be elements useful to prevent irrelevant correlations.

C.1.2 Expected Environmental Conditions

The **expected environmental conditions (expected input of the transfer function)** is a set of environmental conditions under which the AI-based system is specifically designed to function. This is the nominal environment, the nominal operating conditions the system is supposed to evolve in.

The goal of describing these expected inputs is to define what the system will encounter during its nominal use:

- The first aspect is to describe the nominal environmental conditions of use of the system;
- The second is to describe the elements, actors, scenery that the system will encounter and likely to influence the perception of the environment;
- The description shall be made considering the sensors and their ability to perceive the world in nominal conditions at a given time. For instance, the description will change between a camera and a radar.

The description of these expected environmental conditions shall be consistent with the perimeter defined by the Operational Design Domain (ODD).

C.1.3 Harmful Environmental Conditions

The **harmful environmental conditions (harmful inputs of the transfer function)** are environmental conditions that could lead to the malfunction of the AI-based system. In this environment, the system is likely to encounter elements that may represent a challenge, even a threat (exploiting insufficiencies of specification or causing potential failures) to the sensors, or more globally to the system operation. They are part of the global inputs, but bring particular challenges. However, it should be noted that these technical elements are not unsafe per se, it is rather the resulting behavior of the AI-based system in response to these elements that is hazardous. The harmful environmental conditions are of the same type than the triggering conditions proposed by the ISO 21448 related to the Safety of the Intended Functionality (SOTIF) (*ISO 21448*, 2022).

- Example 1: It could be edge cases of the Operational Design Domain (ODD)
- Example 2: Drive Pilot TJC from Mercedes only works in daylight. However, what happens in case of an eclipse, AKA an event that happens very occasionally and that could be reasonably foreseeable?

The identification of harmful environmental conditions at system levels should help the work of the Data Scientist, either to over sample this data to enable the system to cover it properly, or to collect it in a specific dataset dedicated to robustness validation tests.

C.1.4 Required Intended Behavior

The **required intended behavior (required/expected output of the transfer function)** is an expression of the intended behavior to consider for the phases of learning, testing.

What are the expected behaviors the system should learn? How do we describe these behaviors in the operating environment?

The required intended behavior shall be described in the system requirements as an expected output of the transfer function of the system. It constitutes a specific part of the dataset specification and learning process. It could be expressed in the following manner:

- Requirements for annotation of the input data;
- Requirements for KPI and reward functions design;
- Requirements for Stop criteria / Success criteria;

These requirements could be expressed in different forms:

- Objects to consider;
- Set of rules;
- Parameters with limits and thresholds;
- Set of alternatives;
- Etc.

C.1.5 Unwanted or Disturbing Behavior

The **unwanted or disturbing behavior (harmful outputs of the transfer function)** is an expression of the behavior we want to avoid for the system under development. It is likely to render the system unable to reach his goals or can be considered as a hazardous behavior that is unacceptable in the operational

viewpoint of the system. The definition of this behavior will impact the dataset specification, the learning and testing phases, and the monitoring strategy.

During the implementation process, the dataset need examples of disturbing behaviors in order to avoid them:

- Example of unwanted behavior: An AEB system that brakes on road signs used to warn drivers about the presence of a pedestrian crossing (like figurines on the sidewalk or pedestrians painted on the road). These particular elements should be taken into consideration and the AI agent should be trained not to brake on these specific infrastructure elements without degrading real pedestrian detection.

During the V&V phase of the AI-based system:

- Adequate tests shall be performed to detect instances of these outputs during testing to improve the model/system

C.2 Further examples related to technical elements classification

Two examples of Pedestrian Detection:

A pedestrian on the sidewalk that disappears from the perception
  - From the ML Component viewpoint: it seems hazardous
  - From the Operational viewpoint: it is not necessarily an issue if the pedestrian trajectory is not convergent with the vehicle one;
- Driving with a foggy weather, where a pedestrian shadow could be erroneously perceived as a real pedestrian. The key is to ensure that it should not be a relevant class for the detection system.

Example of the Drive Pilot TJC from Mercedes:

- Use Case of the emergency vehicles with detection of their sirens (via microphones, etc.)
- How to define the sound environment? What are the elements to distinguish?

C.3 Conclusion of Part 2

The distinction between environmental conditions (inputs of the transfer function) and behaviors (outputs of the transfer function) compose the System Specification of AI that replaces the conventional Software Specification on a data-centric approach.

Through this activity, we want to highlight that there is no universal dataset. The need synthesis defined in the Operation Approach will drive the Data Specification, and require the intermediate step of System Specification to instantiate operational requirements into technical elements. Technical elements shall describe either expected elements or unwanted ones in order to prevent irrelevant correlations.

The result of the System Specification is an input for the Data Specification.

Reusing existing datasets impose that we can verify that their construction is consistent with the System Specification. Otherwise, a new dataset should be produced to guarantee the coverage of the expected behaviors.

D. Part 3 - Capabilities of data to cover system needs

Introduction to Part 3

The previous activities highlighted the change of paradigm that explain why AI-based systems need to learn their transfer function and what should be the environmental conditions and expected behaviors guiding their learning. To do so, AI-based systems rely on sensors to consider the environmental conditions in order to make a decision and apply it. The goal of this chapter is to highlight the possible discrepancies between System Specification written in engineer human language, and information collected by sensors and used in a digital way (data). We will study the relevance of data from sensors supposed to capture the “real world”. This entails the following questions:

- What is the ability of digital data (from sensors) to represent the “real world” described in operational and system requirements?
- What should be the strategy to implement at system level shall to deal with sensors limitations?
- What kind of architectural patterns shall be considered to ensure the best performances of sensing (redundancy, dual rating, etc.?),
- What types of failures mitigation strategies are expected? (transition to minimum risk conditions)
- What kinds of sampling and combinatorial aspects of expected input shall be considered?

Overview of the Part 3

In the following section, we will focus on:

- The role of sensors to capture data for AI-based development;
- how the choice of sensors and their combinations is important for AI-based system design.

D.1 Sensors in AI-based systems

As we demonstrated previously, to specify an AI-based system, we need to explicitly describe its inputs (operating environment of the AI-based system) and outputs (behavior of the AI-based system). At a lower level of description of the system, the AI model will be fed data coming from the environment.

To perceive this environment, the system will either be equipped with sensors or connected to another system with sensors. Sensors acquire a part of the reality, gathering information based on their environment, their capabilities and their state. The data provided by the sensor is a model of some parts of reality that are both spatially and temporally sampled. Therefore, the direct input of the AI model is a representation of the environment through the model.

To represent the part of acquisition in an AI-based system, two choices exist as illustrated by Figure 7:

- The sensor is a part of the system: we describe the inputs as the sensor will perceive it;
- The sensor is external to the system: we describe the inputs as agnostic to the sensing capacity ahead

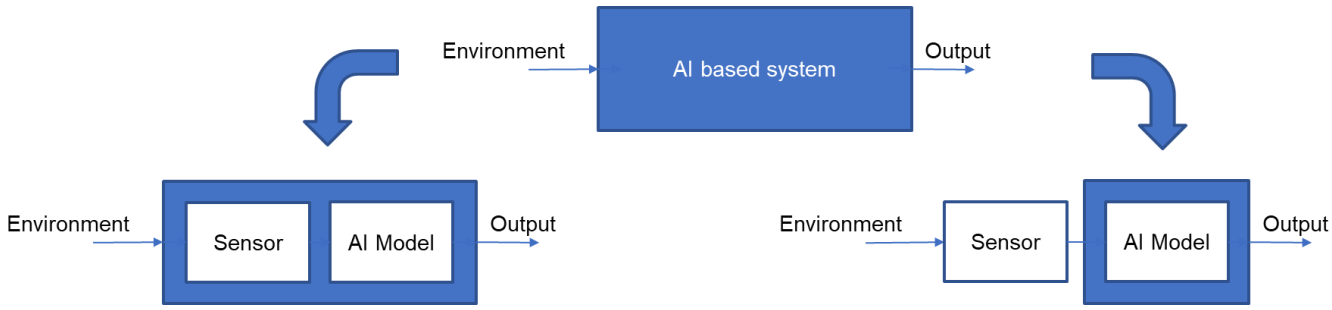


Figure 7 Place of sensors in AI-based systems

NB: Any acquisition system is assimilated to a sensor. This includes virtual ones. For example, image databases gather pictures from a wide variety of camera and their settings.

D.2 Ability of digital data to represent the “real world” described in operational and system specifications

D.2.1 Context on the use of sensors

Digital data from sensors is only a representation of the real-world. It relies on the ability of a given system to sense physical phenomena and record them with a specific format. The sensor has senses, and these senses have abilities but also limitations.

For example, a camera produces an image when reading its array of light sensors. The light sensors deliver an output depending on the quantity of photon hitting them. When the luminosity is very low, the sensors struggle to give a sharp image; when it is too bright, the sensors are saturated. The ability of a camera to detect an object at very long range, given a fixed sensor size, will depend on the optical elements and the ability to zoom in front of it. Its limitations come from the method of sensing (the physical phenomena involved) but also the design of the sensor. A pedestrian at 200 meters in front of a very wide-angle camera may take 3 pixels while with an objective, it will make it 100 meters.

That is why perception in automated driving relies on the use of multiple sensors to “view” the world. As illustrated by Figure 8 (Kumar et al., 2020), automated driving often relies on a combination of camera, LIDAR (Light Detection and Ranging) as well as other sensors to reconstruct a realistic model of the world view, with the relevant obstacles and objects to detect. LIDAR can be used in night conditions, while RGB camera are limited to conditions with sufficient levels of luminosity.

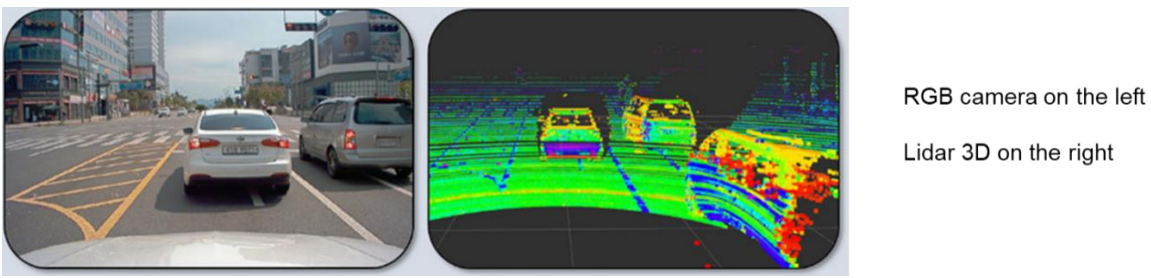
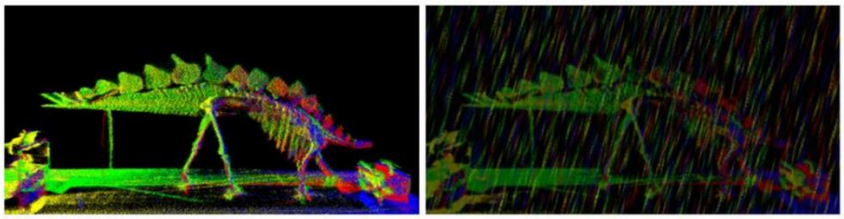


Figure 8 Comparison of what RGB camera and LIDAR see, from Kumal et al.

However, using only a LIDAR is not sufficient, as this type of sensor is also subject to a certain number of limitations. For instance, Figure 9 (Dorazio, 2018) illustrates that several environmental conditions can disturb LIDAR-based perception, by producing noise.



Lidar on the left

Lidar and rain on the right

NB : these are simulated results

Figure 9 Pictures of what LIDAR see in different environmental contexts

For this reason, in order to cover for each sensor insufficiency, it is part of design best practices for vision-based systems to rely on sensor fusion, at it provides enhanced availability of the systems, as well as improved reliability.

D.2.2 Gaps on sensors technology and how to solve them

The use of sensor fusion is an answer in order to fill the “gaps” between different technologies and mitigate their respective limitation.

For example:

- If there is an obstacle in front of the object of interest, the photon won’t go through it, but a radar may be able to “see” it with the Doppler effect;
- A thermal and infrared camera may be coupled to a visible light (RGB) camera to perform with low luminosity environments;
- The Doppler effect may change sound recorded by the microphone.

	Type	Param1	Param2	Param3	Limitations			Advantages
Caméra	RGB	Global shutter	Résolution	Focale	nuit	brouillard	éblouissement	good resolution low cost eye safe
	N&B	Rolling shutter	Compression	Ouverture	nuit	brouillard	éblouissement	
	Thermique		Fréquence	"optique"	brouillard		saturation thermique	
	TOF (depth)							
	Stéréo'				nuit	brouillard	éblouissement	
	IR				brouillard		éblouissement IR	
	NIR				brouillard		éblouissement NIR	
Lidar	2D	Standard spindle	Longueur d'onde	Range	portée	brouillard	éblouissement	medium/long range works at night resolution
	3D	Flash		Ouverture	portée	brouillard	éblouissement	
		Solid-state		Fréquence	portée	brouillard	éblouissement	
Radar	SAR/2D	Longueur d'onde	Fréquence	Ouverture	ouverture	portée		long range obstructions price
Sonar /US		Longueur d'onde	Fréquence	Ouverture	brouillard	vitres	portée	
Microphone								

Figure 10 List of sensors with their parameters, pros and cons

Sensor fusion nowadays rely on different types of architectural patterns that achieve various expectations.

D.2.3 Use of sensors fusion

Sensor fusion typically includes three levels of abstractions, as illustrated by Figure 11 (Debie et al., 2019):

- Sensor-level abstraction processes the raw sensor data. If multiple sensors are used to measure the same physical attribute, the data can be combined at this level. For sensors measuring different attributes, the data is combined at a higher level;
- Feature level abstraction extracts feature from various independent sensors to produce individual feature vector representations;
- Decision level abstraction classifies the various features and uses the resulting data to make decisions about the environment and identify any necessary actions that need to be carried out.

Each data/sensor fusion paradigm (statistical, probabilistic, and knowledge-based) can be used at different levels of processing abstractions. In addition to these three basic levels, hybrid models can be implemented. For example, data from two different sensors can be combined to produce a single feature set and the resulting classification model used at the decision level. Or, the results of feature extractions and decision level classifications of multiple modalities can be used for training to refine the decision level classification algorithms for other modalities.

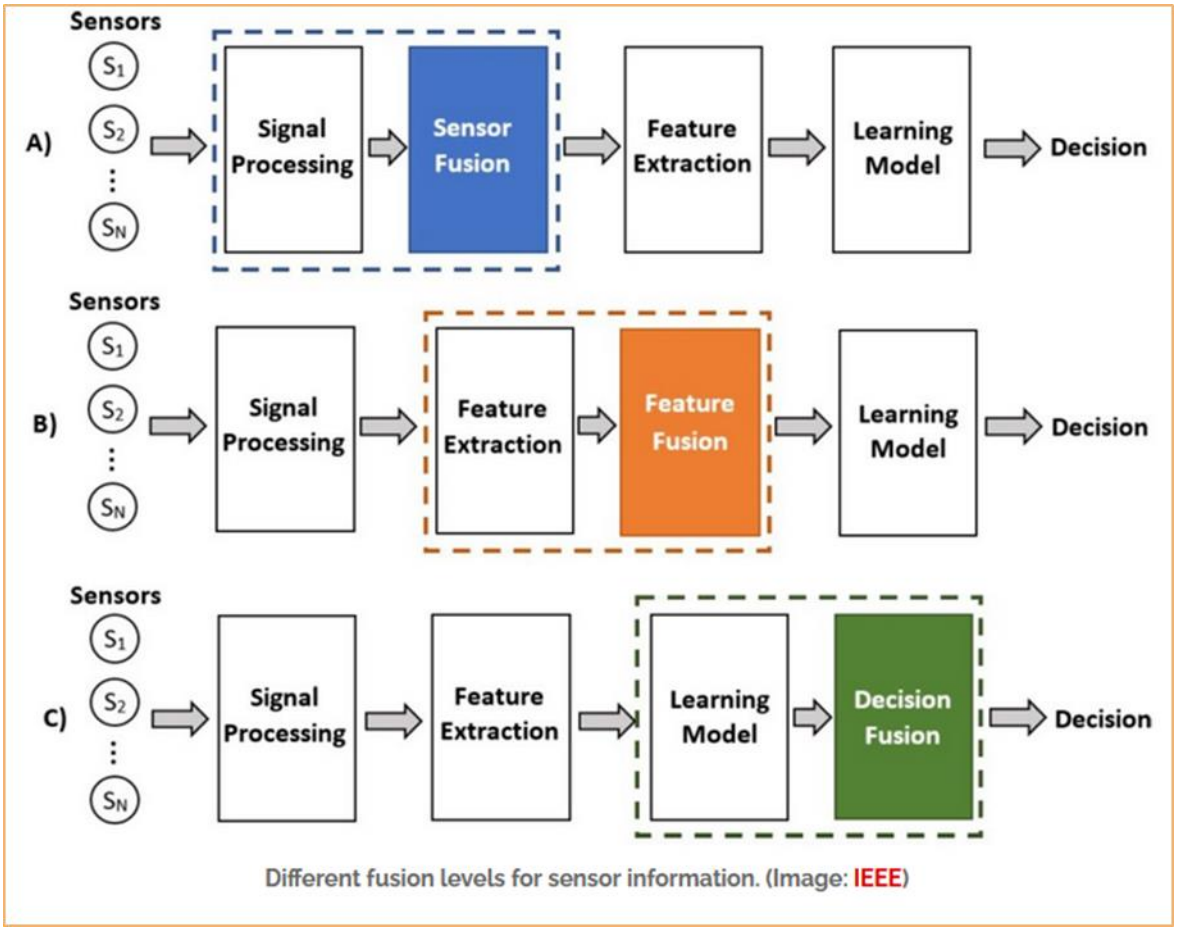


Figure 11 Different fusion levels for sensor information

There are five levels of fusion, as illustrated by Figure 12 (Singh & Tripathi, 2014):

- Level 0: source preprocessing is the lowest level of data fusion. It includes signal conditioning and fusion at the signal level. In the case of optical sensors, it can include fusion at the level of individual pixels. The goal of preprocessing is to reduce the quantity of data while maintaining all of the useful information needed by the higher levels.
- Level 1: object refinement uses the preprocessed data from the previous level to perform spatio-temporal alignment, correlations, association, clustering or grouping techniques, state estimation, the removal of false positives, identity fusion, and the combining of features that were extracted from images. Object refinement results in object classification and identification (also called object discrimination). The output is produced in consistent data formats that can be used for situation assessment.
- Level 2: situation assessment establishes relationships between the classified and identified objects. Relationships include proximity, trajectories, and communications activities and are used to determine the significance of the objects in relation to the environment. Activities at this level include the prioritization of significant activities, events, and any overall patterns. The output is a set of high-level inferences that can be used for impact assessment.

- Level 3: impact assessment evaluates the relative impacts of the detected activities in level 2 to support a situation analysis. In addition, a future projection is made to identify possible near-term vulnerabilities, risks, and operational opportunities. The future projection includes an evaluation of the threat or risk and a prediction of the anticipated outcome.
- Level 4: process refinement is used to improve Levels 0 to 3 and to support sensor and general resource management. Initially, this was a manual task to achieve efficient resource management while accounting for task priorities, scheduling, and controlling available resources. While the goals have not changed, modern systems increasingly supplement manual analysis with AI and ML tools.

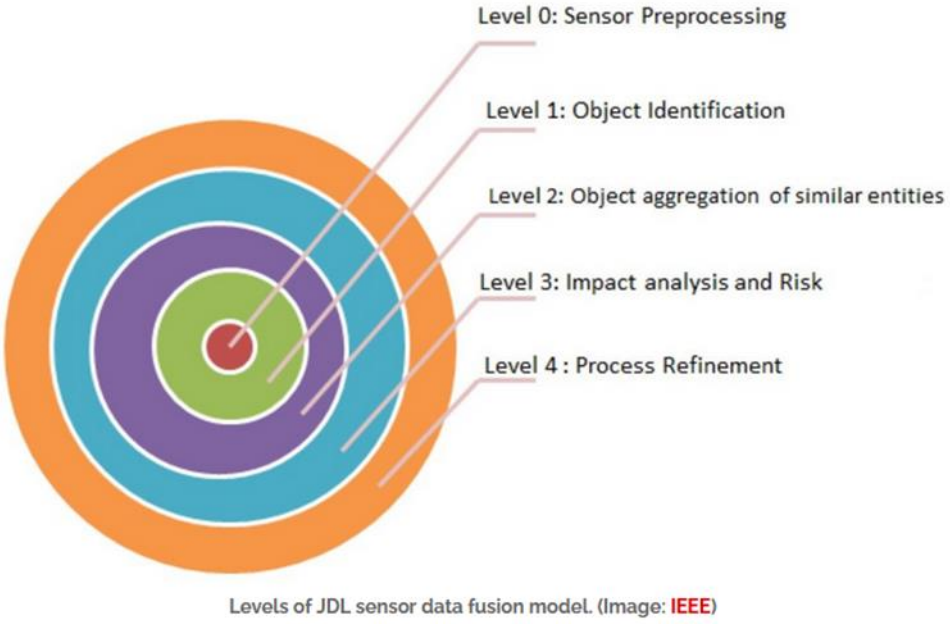


Figure 12 Levels of data fusion models

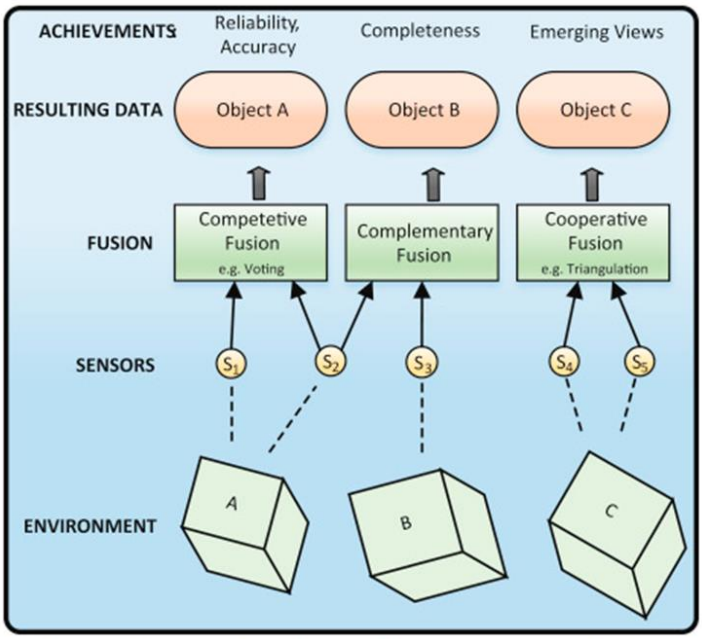
D.2.4 Use of sensors architectural patterns

Sensor relationships and sensor fusion architectures can be classified based on the relationships between multiple sensors in a system, as presented in Figure 13 (Galar & Kumar, 2017, p. 1):

- Complementary sensors provide information that represents different aspects of the environment and can be combined to produce more complete global information. In a complementary implementation, the sensors operate independently and do not directly depend on each other but can be combined to give a more complete image of the phenomenon under observation. For example, combining information from a rotational speed sensor and a vibration sensor can provide enhanced information about the condition of motors and gearboxes. Or, in the case of vision systems, images of the same object from two different cameras or a camera and LIDAR sensor can be combined to provide a more complete “picture” of the environment.
- Redundant or competitive sensors are used to provide information about the same target, and their outputs are combined with increasing the reliability or confidence of the output. For example, if the field of view of two cameras overlaps, the overlapping area is classified as redundant sensing. In the

case of competitive sensing, each sensor measures the same property; in the case of two cameras, both would have the same field of view. There are two cases of competitive configurations: fusion of data from different sensors; or fusion of data from a single sensor measured at different points in time. A special case of competitive sensor fusion can be used when monitoring critical parameters, called fault-tolerant fusion. Fault-tolerant designs are typically based on modular designs such as N+1 redundant architecture.

- Cooperative sensor fusion combines the inputs from multiple sensor modalities, such as audio and visual, to produce more complex information than the individual inputs. Combining two cameras with different viewpoints can be used to synthesize a three-dimensional representation of the environment. Cooperative sensor fusion is complex, and the results are sensitive to the accumulated accuracies in all the included sensors. While competitive sensor fusion can increase accuracy and reliability, cooperative sensor fusion can decrease accuracy and reliability.



Competitive, complementary, and cooperative sensor fusion.

Figure 13 Competitive, complementary and cooperative sensor fusion

These different categories have several pros and cons for AI-based systems. “Complementary sensors” relations are helpful to produce data for AI-based systems or reliability purposes while “competitive sensors” have an edge on safety-related topics.

D.2.5 Sampling and combinatorial aspects

The System Specification must include the right amount of constraints in order to be able to choose wisely the sensor set-up and settings, but not over-constrained to have some choice. Among these constraints from sensors, we have to consider topics related to resolution, sampling and refresh rate; as well as topics related combinatorial, volume and likeliness.

D.2.5.1 Topics related to resolution, sampling and refresh rate

- Appropriate resolution of sensors:
  - High resolution may create unwanted effects, such as the Moiré effect (Pritchard, 2009) in Figure 14:
  - The picture size of a 4K camera is much bigger than a Full HD camera (four times more pixels);
- Appropriate method of down sampling:
  - An inadequate down sampling applied to adapt to a model input size may erase important features;
- Appropriate refresh rate for video-based systems:
  - A low refresh rate may not be suitable for real-time critical systems, while a high refresh rate may burden a network requiring important processing time.

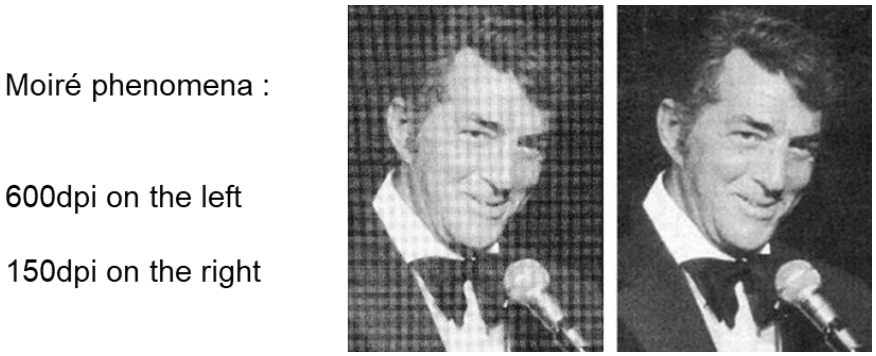


Figure 14 Illustration of the Moiré phenomenon

D.2.5.2 Topics related to combinatorial, volume and likeliness

In order to train the model, the accessibility of a dataset and its volume must be addressed:

- An insufficient dataset volume will lead to an incomplete learning phase;
- Volume must be combined with an adequate representativity/variety/quality of data:
  - An Inadequate representativity will lead to biases and/or poor results;

The operational requirements shall be explicit regarding the objectives (needs) of the systems and the environment it will navigate into. This shall allow for an adequate dataset specification.

D.2.6 Conclusion of Part 3

AI-based systems rely on sensors to collect the data needed for their learning and their operations. These sensors are diverse and each category come with their pros and cons. In order to achieve the Intended Purpose, it is needed to build an association of sensors that may be inspired by architectural patterns.

The nature of sensors has impact on the data, and the data impact the AI-component. To fulfil its role of intermediate between the Operational Specification and the Data Specification, the System Specification has to take into account the typology of sensors used for the AI-based system when describing technical elements. In addition, it shall include requirements on the way to configure the sensors in order to capture relevant data in accordance with the Design Intent.

E. Part 4 – Identification of system-level elements to address ML choices

This part has been identified as within the scope of the System Approach, but shall be developed with the adequate knowledge, resources and profiles to be relevant and we have not succeeded in reaching them yet. This part is complementary to the section C above, related to data, and is dedicated to identify design choices which can be used to guide the learning methods. An example of related topics using a bounding box or segmentation in a detection problem will be pulled by the operational and system needs.

This chapter has to be developed in a future work to complete System Approach.

F. Part 5 – Requirements for embeddability in AI engineering

We invite the readers to check the EC7 project that may have developed elements that are aligned with this part of the System Approach dedicated to embeddability in AI engineering.

G. Part 6 - Requirements for trustworthiness attributes implementation

The instantiation of generic attributes on a given system is part of the System Approach. This instantiation relies on the Operational Specification.

We invite the readers to check the EC2.15 project that may have developed elements that are aligned with this part of the System Approach dedicated to trustworthiness attributes.

H. Part 7 - Requirements for monitoring implementation

We invite the readers to check the EC3 project that may have developed elements that are aligned with this part of the System Approach dedicated to monitoring implementation.

I. Conclusion

Thanks to the Operational Approach developed in the deliverable 218A, we aim to identify and characterize operational needs which can only be managed by an AI-based system (could not be reached by a conventional software).

With the System Approach developed in this deliverable, we aim to gather, at system level, artefacts that are required for the AI-component implementation and the coverage of operational needs. It is highly dependent on the realization of the Operational Specification that is used as input, and remains at a subsystem level close to the boundaries of the AI component. These activities should serve as a bridge between the needs of the Operational Specification and the requirements of the ML & Data Specifications.

The System Approach should raise awareness on several activities that result from the new paradigm of AI-based systems:

- First the fact that AI-based systems learn their transfer function, unlike systems where the transfer function is specified. This shift in paradigm explains why in the context of AI-based systems, the inputs and outputs have to be the target of System Specification.
- Secondly, the importance of properly classifying the inputs and outputs for the systems, as they will guide the technical elements required for the Data Specification. On the one hand, Expected and Harmful Environmental Conditions; on the other hand, Required Intended Behaviors and Unwanted or Disturbing Behaviors. These four categories of inputs and outputs give a rigorous vision on the way to consider the elements needed for dataset specification in order to correctly identify expected correlations.
- Furthermore, to take advantage of these elements, they must be captured thanks to an appropriate combination of sensors whom advantages and limitations must be considered, as they will drive the type and quality of data that will gathered. Architectural patterns for sensors have several advantages: sensors can be combined by using architectural patterns. Technical elements description has to consider this sensors configuration in order to reflect inputs that the AI-based system can interpret.

This System Approach should also tackle additional topics, that could not be developed in this deliverable. Indeed, the System Specification is expected to help in the identification of system-level elements to address ML choices (choices of algorithms or objective functions, for instance), as well as provide several requirements on the topics of embeddability, trustworthiness attributes and monitoring implementation.

The System Specification shall ensure the traceability of each design choice and renouncement at system level in order to grant the adequation of Dataset Specification and ML Specification with the Intended Purpose.

J. Bibliography

REPORT on the proposal for a regulation of the European Parliament and of the Council on laying down harmonised rules on Artificial Intelligence (Artificial Intelligence Act) and amending certain Union Legislative Acts, Report-A9-0188/2023, European Parliament (2023). https://www.europarl.europa.eu/doceo/document/A-9-2023-0188_EN.html

Debie, E., Fernandez Rojas, R., Fidock, J., Barlow, M., Kasmarik, K., Anavatti, S. G., Garratt, M., & Abbass, H. (2019). Multimodal Fusion for Objective Assessment of Cognitive Workload: A Review. *IEEE Transactions on Cybernetics*, *PP*, 1–14. https://doi.org/10.1109/TCYB.2019.2939399

Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS, 52021PC0206, European Commission, COM/2021/206 final (2021). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

Galar, D., & Kumar, U. (2017). Chapter 1—Sensors and Data Acquisition. In D. Galar & U. Kumar (Eds.), *eMaintenance* (pp. 1–72). Academic Press. https://doi.org/10.1016/B978-0-12-811153-6.00001-4

ISO 21448:2022 Road vehicles Safety of the intended functionality (International Standard Published 21448:2022; Version 1). (2022). https://www.iso.org/fr/standard/77490.html

ISO/IEC DIS 5338: Information technology - Artificial intelligence - AI system life cycle processes (Cobaz 5338; Version 1). (2023). https://cobaz.afnor.org/notice/NORME/XS142213/XS142213

ISO/IEC/IEEE 15288:2023 Systems and software engineering—System life cycle processes (15288). (2023). https://www.iso.org/standard/81702.html

ISO/IEC/IEEE 24765:2017 Systems and software engineering—Vocabulary (International Standard 24765; Version 2). (2017). https://www.iso.org/standard/71952.html

Kevin Mantissa & Christophe Bohn. (2024). *Methodological Guideline for AI-based System Design at Operational and System Level: Operational Approach* [Methodological Guideline]. Confiance.AI.

Kumar, G. A., Lee, J.-H., Hwang, J., Park, J., Youn, S.-H., & Kwon, S. (2020). LiDAR and Camera Fusion Approach for Object Distance Estimation in Self-Driving Vehicles. *Symmetry*, *12*, 324. https://doi.org/10.3390/sym12020324

Pritchard, G. (2009, December 11). The Print Guide: Moiré. *The Print Guide*. http://the-print-guide.blogspot.com/2009/12/moire.html

Rick Adcock. (2023, November 20). *SEBoK - Applying Life Cycle Processes* [Wiki]. https://sebokwiki.org/wiki/Applying_Life_Cycle_Processes

SEBoK. (2023). The Guide to the Systems Engineering Body of Knowledge (SEBoK) Is a Living, Authoritative Guide of the Systems Engineering Discipline. https://sebokwiki.org/wiki/Guide_to_the_Systems_Engineering_Body_of_Knowledge_(SEBoK)

Singh, D., & Tripathi, G. (2014, March 6). *A Survey of Internet-of-Things: Future Vision, Architecture, Challenges and Services*. 2014 IEEE World Forum on Internet of Things, WF-IoT 2014. https://doi.org/10.1109/WF-IoT.2014.6803174

Title: Methodological_Guideline_for_ AI based System Design at Operational and System Level: System Approach

Keywords:

System approach, sensor, System definition, System engineering, System Design

Our partners