Safe-by-Design Development Method for Artificial Intelligent Based Systems

Gabriel Pedroza, Morayo Adedjouma

Safe-by-Design Development Method for Artificial Intelligent Based Systems

Gabriel Pedroza, Morayo Adedjouma

CEA LIST, Department of System and Software Engineering
P.C. 174, Gif-sur-Yvette, 91191, France
{firstname.lastname}@cea.fr

Abstract

Albeit Artificial Intelligent (AI) based systems are nowadays deployed in a variety of safety critical domains, current engineering methods and standards are barely applicable for their development and assurance. The lack of common criteria to assess safety levels as well as the dependency of certain development phases w.r.t. the chosen technology (e.g., machine learning modules) are among the identified drawbacks. In addition, the development of such engineering methods has been hampered by the emerging challenges in AI-based systems design mainly regarding autonomy, correctness and prevention of catastrophic risks. In this paper we propose an approach to conduct a safe-by-design development process for AI based systems. The approach relies upon a method which benefits from a reference AI architecture and safety principles. This contribution helps to address safety concerns and to comprehend current AI architectures diversity and particularities.

***Index Terms*—safe-by-design, AI, safety, engineering**

I. Introduction

The implementation of Artificial Intelligence (AI) based systems has progressed in many aspects. The technology shows increasing levels in tasks automation and adaptation to the context. For instance, for self-driving vehicles, we can cite the DriveMe project on Volvo XC90 series launched in 2013 [1], the Autopilot technology integrated in Tesla vehicles since 2013 [2], and the Waymo project of Google which conducted the world's first self-driving ride on public roads [3] claiming maximum autonomy level (5). Several instances of systems based upon AI technology can be found in the literature, *e.g.*, [4], [5]. The main concerns addressed by those architectures are related to (1) the limits imposed by sensors' detection, (2) the heuristic nature of machine learning (ML) and deep learning (DL) techniques, and (3) the variability and complexity of context scenarios. Whereas current technology and implementations show approach feasibility and provide some solutions to referred concerns, they are mostly committed to improve system's autonomy letting aside other aspects of the engineering process. In general for AI-based systems development, a variety of engineering techniques are applied and combined with almost empirical parameters, choices which are finally tuned to optimize performance [6]. Despite the engineering process has proven certain effectiveness, it also shows some drawbacks regarding safety assessment. In particular, the lack of a comprehensive process to settle common criteria and thresholds for safety evaluation and certification. To our knowledge, the problem is quite hard to solve and no current approach overcomes related issues: an AI-based system should integrate sensor devices with limited - sometimes opaque - detection capabilities ($\leq$90% in average), deploy algorithms to identify and properly react to complex, rather unforeseen, situational scenarios and, on top of that, ensure negligible likelihood of occurrence for critical hazards and disfunctioning. Moreover, whereas for typical development methods, like the V-cycle, the phases and their sequencing are almost static, it is observed that for AI-based systems, the nature of certain engineering phases and their order may vary. Some of the factors for that to occur are the dependency of the engineering process on the AI technology choices, on the knowledge bases - used for learning - and on their maturity (representativeness of data sets, events, phenomena). Although, it is consensual that safety analyses should be conducted as early as possible and all along the life-cycle, the few initiatives on that respect, like ISO/IEC 23053 [7], are still work in progress. Others, like ISO 21448 [8], provide insights on a safety-integrated process for autonomous vehicles without settling generic criteria suitable for other application domains. Consequently, current

DOI reference number: 10.18293/SEKE2019-094

standards landscape is barely applicable to the growing space of AI-based systems, and new safety methods and analysis processes are required to develop them. To tackle the referred issues, the main contribution of this paper is an overall generic iterative (OGI) method to conduct safe-by-design development of AI-based systems relying upon a generic architecture and safety principles.

The rest of the paper is structured as follows. In Section II, the reference AI architecture is introduced. In Section III, some safety relevant aspects to AI-based systems are highlighted. The OGI process for AI-based systems development is described in Section IV, including safety assessment phases. In Section V, the safe-by-design method is applied to the development of an autonomous system. Some related works are explained in Section VI. Finally, a discussion and work perspectives come in Section VII.

II. Generic Reference AI-based Architecture

The specification of a process development for AI-based systems should consider the following particularities:

- *Engineering process dependent on AI technology:* Subsystems or components implementing ML/DL modules are based upon parameters which may require to be set during conception, design, implementation and validation phases. For those subsystems and components, a learning phase should be introduced whereas for other typical (non-ML-based) subsystems and components, the learning phase is non-existing.
- *Engineering process dependent on knowledge bases:* The learning phases strongly depend upon target objectives and external knowledge bases (KBs). For many complex AI-based systems, an iteration on design parameters may be necessary after a validation campaign, *e.g.*, to adjust detection ranges and accuracy. Detailed requirements cannot be elicited before knowing the effectiveness of knowledge bases, ML/DL techniques, and parameters choices.
- *Knowledge bases maturity:* Due to previous aspects, the AI-based systems development shall not only depend upon ML/DL techniques and development methods, but also on building up knowledge bases (KBs) and making them evolve so as to improve coverage and accuracy of objects, events, and phenomena detection. Referred KBs shall be useful for detection and for high level reasoning, *e.g.*, reasoning based upon intuition [9].

We propose to integrate the previous specificities from early stages of design. Since a huge diversity of architectures and process cycles currently exist, the specification of a development process should be as generic and comprehensive as possible. We find suitable to first introduce a generic AI-based architecture aiming to cover most of them. The Figure 1 illustrates the coarse domains composing our reference architecture which are briefly described in the following items and in section IV.

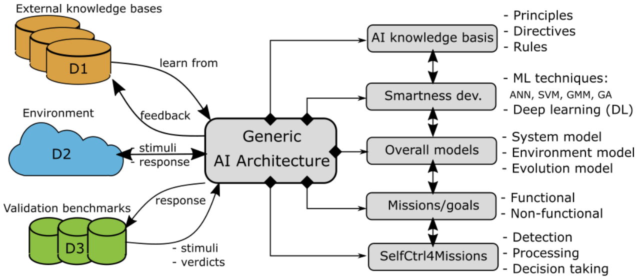

Fig. 1. Generic reference AI architecture

- *External knowledge sources:* this domain comprises KBs required for training ML/DL modules, for their implementation and validation. At the beginning, these external KBs are not part of the system, but they are integrated as a logical component during the engineering process, for instance, after generation of training sets (*e.g.*, via the feature vectors).
- *Missions/goals:* this domain of the architecture covers the fulfillment of functional and non-functional goals and missions of the system. Certain missions and goals may not require the deployment of ML/DL modules. Thus, they can be deployed by components and subsystems relying upon typical technology.
- *AI knowledge basis:* this domain includes components for the fulfillment of principles, directives and rules which guide the AI-behavior. The modules pertaining to this domain provide a basis upon which missions and goals can be accomplished by the system.
- *Smartness development:* the smartness of the system relies, at least, upon two layers of reasoning. The first layer is in charge of detection of external objects, phenomena, and scenarios. The second layer includes monitoring of system status, self-positioning and reacting to external conditions according to missions and AI-principles. The development of system smartness depends on AI techniques like ML and DL.
- *Overall models:* the development of autonomy and smartness demands an understanding of environmental and internal system elements. As for external elements, a model of the environment is to be settled. Among others, this model allows the interpretation of external stimuli. As for internal elements, the system should be able to have a comprehensive model of itself. Among others, this model allows to asses system self-status. Both models capture the current capabilities of the overall system. To ensure certain independence, the AI-system should be able to learn

from new stimuli and situations and it should be able to integrate and deploy new capabilities.

- *Self-control:* this domain includes the functions deployed to realize autonomy during missions and goals accomplishment. A typical functional path comprises sensors→controllers→actuators that respectively support detection, processing and decision-taking.

III. Safety Concerns for AI-based Systems

Safety is one of utmost relevant concerns when designing AI-based systems. However, it is also a vast and complex subject considering the multiple applications domains where they can be deployed and their related specificities. The main safety related activity affected by the specificities of AI-based systems is the hazard analysis at the concept phase of the system. Indeed, the implementation of ML/DL components rise new concerns regarding emerging categories of hazardous events. Referred hazardous events exhibit an increased risk level (which may even be catastrophic) due to the machine overtaking over former human-based activities. Along with more autonomy, the transfer of duties to AI-based algorithms implies no further human interaction as safety barrier in case of hazards. In addition, certain typical safety criteria like redundancy do not suffice anymore to ensure expected levels of availability and also accuracy. The specific aspects addressed in this paper are described in line. Some of these aspects have already been highlighted in emerging safety standards like ISO 21448 [8] (see Figure 2).

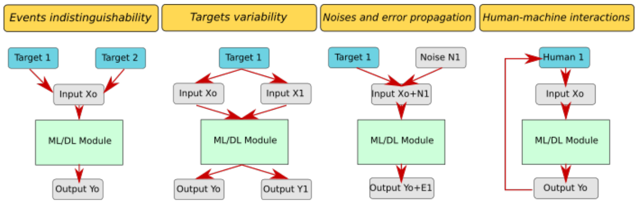

Fig. 2. Hazard events related to ML modules

- *Events indistinguishability.* This issue is mainly due to physical limits of sensors. In particular, because real and virtual images are practically indistinguishable. For instance, the detection of a stop signal can be easily faked by a photo of the same. Light rays and other natural electromagnetic signals used for detection can be reproduced by different objects.
- *Targets variability.* Many objects, elements and phenomena to be detected by AI-based systems exhibit certain variability. Although AI-based systems should cope with referred variability, questions arise if the system may face unforeseen situations beyond its limits. For instance, sudden deterioration or disfunctioning of context signs may lead to wrong detection of objects.
- *Noises and error propagation.* Along with variability issues, the noises added by the background and environmental phenomena increase hazards impact, in particular in case of dismissed/miscalculated noises and errors propagated through the system. Environmental conditions (*e.g.*, solar winds, snow) may impose additional constraints to system operation and behave as background noises.
- *Human-machine interactions harmonization.* The operation of AI-based systems in real environment is quite novel. Whereas risks related to machine-to-machine or machine-to-environment interactions can be assessed during design phases, predict the outcomes of human-to-machine interactions is far more complex [10]. Although an harmonization phase can be conducted, complex human reactions (*e.g.*, psychological, societal, political) are difficult to assess.

IV. Safe-by-Design Method for AI Systems

In this section, we introduce a method to develop the architectural domains specified in section II. The safety aspects described in section III are integrated into the cycle. The method is illustrated in Figure 3 and is described in the following subsections. The method is iterative and several phases can be conducted in parallel even if interdependencies may appear.

A. AI-based systems development method

The main phases of the OGI process are briefly described in the following items.

1) *Missions and goals.* This phase comprises the specification of missions and goals the system should accomplish. The specification mostly targets functional requirements of a typical engineering phase. However, as for safety-critical systems, it also includes the management of non-functional requirements.
2) *AI principles structuring.* This phase covers the specification and structuring of AI principles upon which the system relies. As for typical requirements stages, the formalization and validation of consistency between AI principles is a major stake. This principles are processed afterwards in phase 4.
3) *De-compositional analysis.* This phase is dedicated to decompose, refine and structure missions and goals up to obtain a layer including detailed functions. This phase mostly follows a typical process of design refinement. However, notice that a subset

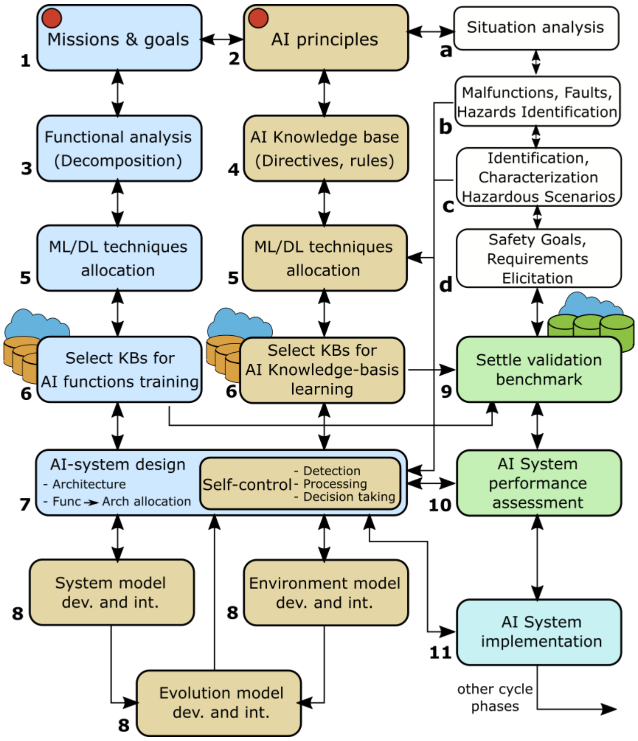
Missions & goals
1
AI principles
2
Situation analysis
a
Malfunctions, Faults, Hazards Identification
b
Identification, Characterization Hazardous Scenarios
c
Safety Goals, Requirements Elicitation
d
Functional analysis (Decomposition)
3
AI Knowledge base (Directives, rules)
4
ML/DL techniques allocation
5
ML/DL techniques allocation
5
Select KBs for AI functions training
6
Select KBs for AI Knowledge-basis learning
6
Settle validation benchmark
9
AI-system design
- Architecture
- Func → Arch allocation
Self-control
- Detection
- Processing
- Decision taking
7
AI System performance assessment
10
System model dev. and int.
8
Environment model dev. and int.
8
AI System implementation
11
Evolution model dev. and int.
8
other cycle phases

Fig. 3. Method for AI systems development

of functions and blocks are to be carried out by ML/DL-based modules as described in phase 7.

4) *AI knowledge basis structuring.* The structured AI principles in phase 2 are first decomposed into a set of high level directives which are afterwards refined up to obtain a set of rules including specific system behaviors. The elicited rules play the role of policies which help to guide - or even enforce - system behavior when needed.
5) *Allocation of ML/DL techniques.* This phase is dedicated to allocate concrete ML/DL techniques and modules to the functions and blocks elicited in phase 3, "*De-compositional analysis*", and to the behaviors elicited in phase 4, "*AI knowledge basis structuring*". The allocation should (1) accomplish the missions and goals of the system and (2) ensure the compliance with rules refined from AI principles and directives.
6) *Knowledge bases selection.* Once the allocation of ML/DL techniques is finished, the KBs for training the respective modules are selected. Of course, building up new or dedicated KBs may be necessary. The KBs are the basis to generate outside-world and system models as well as the evolution model by performing the training tasks (see phase 8).
7) *Detailed AI architecture design.* The detailed design of the AI-based system comprises at least three tasks or sub-phases. The first one consists in proposing the architecture to support the functions specified in previous phases 4 and 5, including allocations. In the second sub-phase, a distribution of functions over the support architecture is conducted. This task covers the exploration of the design space. In the third and last sub-phase, the first layer of intelligence is developed by training the ML/DL modules, *i.e.*, the self-control functions for detection, processing and decision taking.
8) *Overall models development and integration.* The second layer of intelligence is developed in this phase: the AI-system should be able to learn from new stimuli and situations. The models of the external world and the system itself have been partially developed and integrated during previous training tasks. Upon outside-world and system models, an evolution model should be elaborated and integrated relying upon (1) specific principles governing AI autonomy, (2) techniques and metrics to assess and filter new stimuli and situations (*e.g.* adaptive decision making [11]) and (3) techniques to integrate new filtered knowledge so as to grow up system and environment models.
9) *Settle validation benchmark.* After the allocation of ML/DL techniques, a benchmark for validation can be settled. The benchmark includes a set of target objectives used to assess performance and emit verdicts. The target objectives are refined from missions, goals, AI principles and other requirements *e.g.*, safety requirements. The benchmark also includes the data sets selected or generated to test the performance of ML/DL modules. When applicable, the validation benchmark can be based upon typical validation and verification techniques like testing, simulation, and formal verification.
10) *AI system performance assessment.* The system performance is evaluated relying upon the validation benchmark. An iterative process starts in order to fulfill the target objectives non satisfied after the first validation campaign.
11) *AI system implementation.* Along with typical technology components, the components including trained ML/DL modules and their parameters are deployed in this phase.

B. Integration of safety aspects into the OGI cycle

The assessment of the specific safety aspects of ML/DL components, introduced in Section III, demands dedicated principles. Referred safety aspects are integrated into the OGI cycle (see Figure 3) as follows:

a) **Situations analysis.** This activity corresponds to a typical operational situation analysis as in hazard analysis [12]. However, special attention is given to situations in which autonomous functions operate and are at stake.

b) **Malfunctions, faults and hazards related to ML/DL modules.** Along with classical malfunctions, faults and hazards, the aspects referred in subsection IV-A are considered as sources of new potential malfunctioning and hazardous behaviors. For their identification, each safety aspect is related to the architecture parts potentially impacted. For instance, regarding *events indistinguishability*, ML/DL functions, blocks and components carrying out detection are to be considered; for object *variability* and *noises and error propagation*, the reasoning layer used for environment interpretation is in cause, and for the *human-machine interaction harmonization*, the decision-taking reasoning layer is targeted. In addition, the heuristic nature of ML/DL algorithms as well as their dependency to external KBs demand the definition of a malfunctioning matrix including false positives and false negatives occurrence. The autonomy of an AI-based system can be assessed based on its ability to cope with multiple components malfunctioning and hazards (accidental, misuse, natural). Concretely, for each component C_i and target T_j with background B_j the probability of error should be minimized:

$$P[ErrorC_{(i,j)}] := P[FalsePos_{(i,j)}] + P[FalseNeg_{(i,j)}],$$
$$FalsePos_{(i,j)} := \cup_j \{C_i(Accept, B_j)\},$$
$$FalseNeg_{(i,j)} := \cup_j \{C_i(Reject, T_j)\}.$$

Other factors of components failure can be considered, in particular the failure rate λ_i along with the probability of failure over time $P[FailC_{(i,t)}] = \lambda_i e^{-\lambda_i t}$. Thus, the overall probability of component disfunctioning can be calculated by:

$$P[DisfC_{(i,j,t)}] = \omega_1 P[FailC_{(i,t)}] + \omega_2 P[ErrorC_{(i,j)}],$$
$$\omega_1 + \omega_2 = 1.$$

c) **Identification and characterization of hazardous scenarios.** Along with classification of typical hazardous scenarios, the hazardous scenarios involving ML/DL components should be characterized. Notice that, the typical scenarios referred in ISO 26262 [13] as "reasonably foreseeable misuse" are no longer under human control and consequently should also be reclassified. To do so:

- Scenarios involving ML/DL components are defined as combinations of external stimuli, system response (internal stimuli) and malfunctions.
- For each scenario S_k, subsets of KBs are selected to evaluate the performance of involved ML/DL components ($\{C_i\}$):
  - Data/features characterizing legitimate targets T_j (true positives)
  - Data/features characterizing targets' backgrounds B_j (true negatives)
- The likelihood of each scenario $P[S_k]$ is estimated by combining the disfunctioning probabilities $\{P[DisfC_{(i,j,t)}]\}$ of involved components $\{C_i\}$ according to the architecture structure and relying upon basic probability theory.

d) **Safety goals elicitation.** Elicitation of safety goals based upon ASIL levels is rather coarse and thus inadequate considering the current nature of hazardous scenarios $\{S_k\}$. Since a huge diversity and complexity of hazardous scenarios are possible, three cases are identified from which detailed requirements can be elicited:

- The hazardous scenario S_k can be associated to a concrete behavior which is formalized and monitored. A monitoring formula ϕ including safety threshold θ can thus be elicited. For instance, in the case of autonomous vehicles, a formula for minimal safety distance between them is settled: $\phi \geq \theta$. In this case, the detailed requirements are elicited in terms of a maximum threshold violation: $P[\phi < \theta] \leq \delta$.
- The hazardous scenario S_k can be associated to a concrete behavior but deriving a safety monitoring formula is not evident. If the behavior can be formalized and a validation test-bench can be settled, then performance tests are conducted. In this case, the detailed requirements can also be elicited in terms of maximum probability of error given by $\{P[DisfC_{(i,j,t)}]\}$.
- The hazardous scenario or the associated behavior are complex. Conducting performance tests is unfeasible. In this case, the behavior can be formalized and a validation test-bench can be settled relying upon simulation. For sophisticated simulation test-benches, the requirements can be elicited as in previous case, *i.e.*, via $\{P[DisfC_{(i,j,t)}]\}$. Otherwise, safety requirements may depend upon simulation scenarios.

V. OGI Method Tool Support and Evaluation

The modeling and safety assessment framework of the OGI method is implemented in Sophia, a model-based toolset integrated with Eclipse Papyrus editor for UML/SysML models [14]. Sophia uses Papyrus extension mechanisms to support safety and reliability analyses like HARA, FMEA, FTA. We evaluate the OGI method on an ongoing industry-academy experimentation of autonomous shuttles deployment on a sensitive site, as described in [12]. Due to lack of space, the evaluation only focus on selected method phases. A critical mission of the autonomous shuttle is to adopt a safe reaction in presence of an obstacle on its trajectory. The selected scenario S_k corresponding to the mission is presented in the sequence diagram in Figure 4. The scenario S_k involves the *Perception* (*i.e. ContextDetection* and *SelfDetection*), *Navigation* and *Motion control* functions of the system. In the first exchanges, the shuttle detects an object on its route, and is able to brake at safe distance to avoid an accident. In the second iteration, although the system detects an obstacle on its path, it is not recognized as a person due to malfunctioning. Consequently, the shuttle does not take appropriate decision to avoid the collision. Figure 5 presents an excerpt of the architecture design showing the allocation of the scenario functions. The *SelfDetection* function (see *Detected*() in Figure 4) is realized by an inertial measurement unit (IMU), a relative GPS (RGPS) and a ML/DL component for data fusion (see *Fusion()* in Figure 4), the latter computes the vehicle dynamics e.g. speed, acceleration, momentum (see *VehiculeDynamics()* in Figure 4), thus settling the model of the system itself. The *ContextDetection* relies upon a set of LIDARS and Cameras, and upon an ML/DL data fusion component which structures a model including speed and position of objects within the surrounding vehicle's environment. The *Navigation* function is also an ML/DL component that interprets the scene and take a decision according to safety and AI-based directives (see *SafetyAssessment()* in Figure 4). The decision is afterwards sent to an ordinary *Motion control* component for action-taking. Decision Trees, Support Vector Machines, Gaussian Mixture Models, Fuzzy Logic, and other unsupervised machine learning are example of techniques to be used for the ML/DL components. In order to acquire their capability of detection, scene interpretation and decision taking, the ML/DL components have been trained with dedicated KBs prior to architecture's definition. The shuttle process development follows the phase 1 to phase 7 shown in Figure 3. Notice that, in the hazardous scenario in Figure 4 (failure in person's detection), since the system was neither able to perform a collision avoidance maneuver, it presupposes that an intelligence layer is missing: the evolution model to face unknown situations is still to be constructed (see Figure 3, Phase 8). The evaluation of the hazardous scenario is conducted based upon our proposed safety assessment method. To do so, an existing hazard analysis of the shuttle [12] is considered which covers the phases a, b, c, d of the OGI cycle (see Figure 3). The concerning fault in the scenario is the non recognition of a person as a such. The overall probability of the hazardous scenario $P[S_k]$ (see section IV) can be calculated in terms of the probability of disfunctioning for each component $P[DisfC_{(i,j,0)}]$ as follows:

$$\begin{aligned} P[S_k] := {} & \lambda_{LIDAR}\lambda_{Camera} + P[DisfECU_{CD}] \\ & + \lambda_{IMU}\lambda_{RGPS} + P[DisfECU_{SD}] \\ & + P[DisfECU_{Navigation}] + \lambda_{MotionControl}. \end{aligned}$$

A detailed requirement for the scenario can now be specified as $P[S_k] \leq \theta$ which allows to settle a target for the validation benchmark and assess the performance of the system (covering phase 9 and phase 10 in Figure 3).

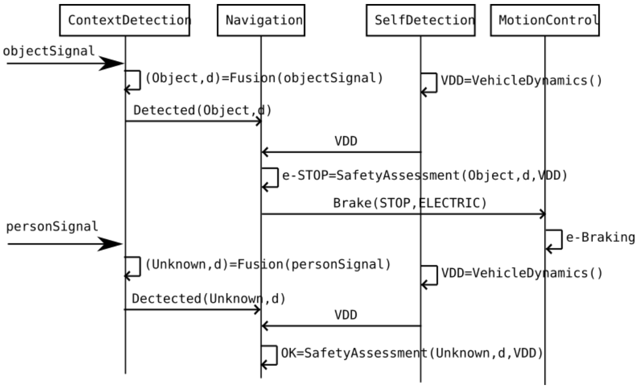

Fig. 4. Shuttle safety critical scenario

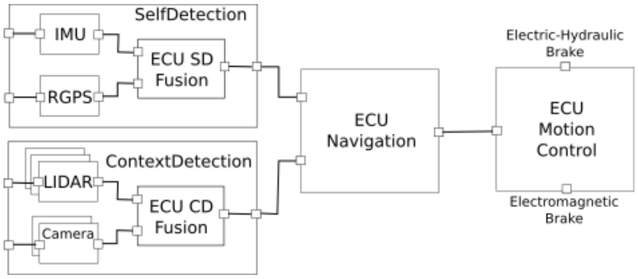

Fig. 5. Excerpt of shuttle architecture design

VI. Related Work

Several works have been presented to identify and integrate safety during AI systems development by industry and academy. In [15], a survey of main AI safety problematics is provided, among them, wrong or cost-ineffective objective functions, and ineffective learning phases. In [16], a preliminary study is conducted discussing different aspects of AI systems' safety. The authors propose the application of classical formal verification principles to AI systems design, like randomized formal methods for training, in combination with design space exploration guided by safety criteria. The authors in [17] present a design strategy for autonomous architectures which maps safety requirements over a global control module. A lot of proposals exist addressing the optimization of ML/DL based modules performance, *e.g.*, [18]. All previous cited approaches either only cover specific phases of the engineering cycle or are quite specific to a given application domain. On the contrary, the work in [11] presents a holistic approach for autonomous systems development addressing similar safety concerns as in this paper, and also relying upon a layered architecture. Unfortunately, it does not propose any criterion for evaluation of hazardous scenarios nor for safety goals elicitation. The theoretical perspective for a safe AI cycle presented in [19] is quite aligned to the one in this paper, with regard to the integration of KBs into the engineering loop targeting data and software diversity. Being a theoretical work, the elicitation of precise system requirements is nonetheless not covered.

VII. Discussions and Perspectives

This paper presents an overall iterative generic (OGI) method for development of AI-based systems. The method is based upon reference architecture domains dependent upon KBs, environment model, validation benchmarks, among others. In addition, the method integrates assessment activities to tackle specific safety-critical aspects of such systems. The assessment provides an enhancement of the typical hazard analysis method to infer safety goals. In particular, it yields the disfunctioning likelihood of an AI-based component considering the typical failure rate added up with the error probability of ML/DL modules. By applying the OGI method to the autonomous shuttle, a multi-factor uncertainty was identified intervening at different levels of AI systems design. A first uncertainty comes from the accuracy and maturity of external KBs which impact the learning process and performance of ML/DL components. The use of fine-grained approaches, like data diversification [19], is promising to overcome this issue. A second uncertainty is the difficulty to apprehend the infinite usage-scenarios space resulting from a "continuum" of possible environmental-operational contexts, variants and configurations. This often leads to poor system-context specifications not representative enough to characterize the scenarios space. A current trend for this shortcoming is to settle enough specificities, from a safety point of view, to define categories of emblematic scenarios [20] useful in particular for benchmarking but without guarantee of coverage exhaustiveness. The third uncertainty factor comes from the performance limits of AI-based components, *e.g.*, sensors blinding or other still unveiled environment events. Interpretation and decision-taking layers are also at stake when arbitration algorithms face contradictory directives and must solve them in critical scenarios, *e.g.*, between safety requirements and AI-knowledge basis (principles, directives, rules). As of today, there is no ethical solution that has reached consensus on this sensitive issue. Finally, since AI-based systems may still be unable to properly react to changing environments (as in the case study in section V), deploy new capabilities in real time is of utmost importance to ensure true systems intelligence and autonomy. In future work, we plan to conduct larger-scale application of the OGI method on others safety-critical AI-based systems to strengthen our conclusions and enforce approach validity. We are also assessing the applicability of other standard-preconceived methods, like FMEA and FTA, in the context of AI-based systems. We would further like to complete the OGI method to cover latter stages of the development cycle, *i.e.*, testing and validation.

References

[1] Volvo. (2018) Drive me project. [Online]. Available: https://www.volvocars.com/intl/about/our-innovation-brands/intellisafe/autonomous-driving/drive-me

[2] Tesla. (2018) Tesla autopilot. [Online]. Available: https://www.tesla.com/autopilot

[3] Google. (2018) Waymo, google project. [Online]. Available: https://waymo.com

[4] S. Dersten *et al.*, "An analysis of a layered system architecture for autonomous construction vehicles," in *(SysCon) 2015 Proceedings*, April 2015, pp. 582–588.

[5] A. Yavnai, "Distributed decentralized architecture for autonomous cooperative operation of multiple agent system," in *AUV'94 Proceedings*, July 1994, pp. 61–67.

[6] G. Pedroza *et al.*, "A speaker verification system using svm over a spanish corpus," in *2009 Mexican International Conference on Computer Science*, Sep. 2009, pp. 381–386.

[7] *Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML).* ISO, 2019.

[8] (2019) Road vehicles - safety of the intended functionality.

[9] A. Rauber *et al.* (2019) Transparency in algorithmic decision making. [Online]. Available: https://ercim-news.ercim.eu/images/stories/EN116/EN116-web.pdf

[10] F. M. Favaro, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in California," *PLOS ONE*, vol. 12, pp. 1–20, 2017.

[11] A. Aniculaesei *et al.*, "Towards a holistic software systems engineering approach for dependable autonomous systems," in *SEFAIS '18 Proceedings*, 2018, pp. 23–30.

[12] M. Adedjouma *et al.*, "Representative safety assessment of autonomous vehicle for public transportation," in *ISORC 2018 Proceedings*, 2018, pp. 124–129.

[13] *ISO 26262: Road Vehicles - Functional Safety*. ISO, 2011.

[14] M. Adedjouma and N. Yakymets, "A framework for model-based dependability analysis of cyber-physical systems," inpress 19th IEEE International Symposium on High Assurance Systems Engineering (HASE) 2019, Hangzhou, China, January 3-5, 2019, 2019.

[15] D. Amodei *et al.*, "Concrete problems in ai safety," *CoRR*, 06 2016.

[16] S. A. Seshia and D. Sadigh, "Towards verified artificial intelligence," *CoRR*, vol. abs/1606.08514, 2016.

[17] C. Molina *et al.*, "Assuring fully autonomous vehicles safety by design: The autonomous vehicle control AVC module strategy," *(DSN-W) 2017 Proceedings*, pp. 16–21, 2017.

[18] C. Huang *et al.*, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231 – 240, 2006.

[19] R. Ashmore *et al.*, "Rethinking diversity in the context of autonomous systems," in *SSS 2019 Proceedings*, 2019, pp. 175–192.

[20] A. Jang *et al.*, "A study on situation analysis for asil determination," vol. 3, 01 2014.