Engineering Dependable AI Systems

Morayo Adedjouma, Christophe Alix, Loic Cantat, Eric Jenn, Juliette Mattioli, Boris Robert, Fabien Tschirhart, Jean-Luc Voirin

Engineering Dependable AI Systems

Morayo ADEDJOUMA*†, Christophe ALIX‡, Loic CANTAT†, Eric JENN§†
Juliette MATTIOLI‡, Boris ROBERT§†, Fabien TSCHIRHART†, Jean-Luc VOIRIN¶

* CEA, France - † IRT SystemX, France - ‡ Thales, France - § IRT Saint Exupéry, France - ¶ Thales DMS, France
morayo.adedjouma@cea.fr, {eric.jenn, boris.robert}@irt-saintexupery.com, {fabien.tschirhart, loic.cantat}@irt-systemx.fr,
{christophe.alix,juliette.mattioli}@thalesgroup.com, jean-luc.voirin@fr.thalesgroup.com

This work has been supported by the French government under the "France 2030" program,
as part of the SystemX Technological Research Institute

***Abstract*—If AI algorithms are now pervasive in our daily life, they essentially deliver non-critical services, i.e., services which failures remain socially and economically acceptable. In order to introduce those algorithms in critical systems, new engineering practices must be defined to give a justified trust in the capability of the system to deliver the intended services. In this paper, we give an overview of the approach that we have put in place to reach this goal in the framework of the French Confiance.ai program. Based on the needs of the industrial partners of the program, we propose a model-based analysis framework capturing the two dimensions of the problem: the one related to *the development and operation of the system* and the one related to *the trust in the system*.**

***Index Terms*—Artificial intelligence, Systems engineering and theory, System verification, Mission critical systems**

I. Introduction

A. A long journey for deploying AI in an industrial solution

Artificial Intelligence (AI) faces challenges on different levels that need to be addressed to speed up the adoption and deployment of AI in industry. Towards that goal, we need a SoSE workbench and methods that support trustworthy AI at both the component and system levels, whether the specifications come from regulation, societal concerns, safety or security, etc.

With an allocated budget of €45M for the period 2021-2024, The French program "Confiance.ai" (www.confiance.ai) is a collective endeavour of 9 major French industrial (Air Liquide, Airbus, Atos, Naval Group, Renault, Safran, Sopra Steria, Thales and Valeo) and academic partners (CEA, IRT Saint Exupéry, IRT SystemX, INRIA). Its target is the creation of an engineering workbench enhanced by methods and tools that will enable the integration of AI into products or services considered as critical, i.e. where accidents, failures or errors could have serious consequences for people and/or valuable assets. The program is organized in 7 projects: EC1, which builds an integrated environment for the development and deployment of trustworthy AI (the "workbench" mentioned previously); EC2, focused on engineering processes and methods; EC3, focused on the characterization and qualification of AI components; EC4, focused on the design of AI components; EC5, focused on data, information and knowledge engineering; EC6, focused on on Integration, Verification, Validation and Qualification (IVVQ) activities; EC7, focused on embeddability issues. The results presented below have been achieved essentially in projects EC1, EC2 and EC6 in close collaboration with the other projects of the program.

Based on the observation that, in some areas, too few AI Proof of Concept (PoC) are reaching production level, a survey was sent to all industrial partners of Confiance.ai to identify their current practices and needs concerning trustworthy AI.

B. Capturing the industrial practices and needs

Each industrial partner received questionnaires dedicated to their respondent's business profile. Each question corresponded to a specific engineering phase as well as to an object of interest (algorithmic engineering, data engineering...) for a total of 140 different questions, most of them being open-ended. 52 questionnaires were returned, from which 3583 responses to individual questions were extracted. The following graph highlights the global distribution of answers according to the respondent's business profile.

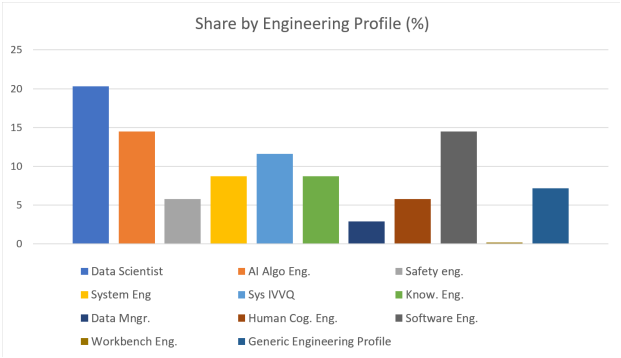

Fig. 1. Global distribution of answers according to the respondent's business profile

Organized in eighteen items, we started by trust and confidence issues, and general questions about the current and future use of AI solutions. Then, a number of objects of interest were addressed, namely the ODD (Operational Design Domain) specification, data, machine learning solutions, knowledge-based solutions, and some transverse aspects:

human-computer interaction, safety and software interoperability. All engineering phases were examined, starting by lifecycle management, followed by seven phases: requirements, design, algorithmic engineering, assessment, deployment, supervision/monitoring and certification. The last item contained a few additional issues not included in the previous ones.

- **Trust and confidence**: Dealing with general considerations about trust/confidence, the questionnaire emphasizes the need to share a common view of the terminology used, of the available technologies and methods. However, most bibliographic references mentioned are norms and standards. The main expected services are linked to certification and validation on representative use cases, with supporting methods and tools.
- **Generalities**: Many partners have already implemented AI components in a solution, but only a small number of them have been up to operational use. However, an increasing number of applications are envisaged, to improve existing system or to provide new functionalities: e.g. perception, decision support, question answering, mission support, command and control systems, etc. Foreseen applications are both batch and streaming/real time, with optimism on the level of autonomy that can be reached in the next five years.
- **Operational Design Domain (ODD)**: Since the ODD concept is relatively new, there is a lack of methods and tools for its definition. Several partners use in-house solutions for its formalization, but with no confidence in their methodology regarding the necessary components and properties they may consider. ODD is also deemed important for the operational phase since the system must detect in operation when it exits the ODD limits to perform the adequate fallback actions. However, the partners have no clue on how to deal with this challenge.
- **Data engineering**: Many tools are mentioned for various aspects of the data lifecycle: feature engineering, data augmentation, training and validation, data engineering in general. Many off-the-shelf commercial tools are used as well as specific in-house tools developed by the partners often using Python, for example with Scikit-learn. A number of data engineering methodologies are mentioned but in most cases the tools play the role of methodologies.
- **Machine Learning**: Industrial partners use commercial and academic tools such as Tensorflow, Keras, Scikit-learn, Pytorch... and sometimes in-house products for their Machine Learning (ML) developments. In upstream phases, no methodology seems to be used, but once engineers move to practical implementation, they are empowered by the richness of the current supply.
- **Knowledge-based approaches**: Knowledge-based approaches are today little used by industrial partners. Only few mentions of ontology tools such as Protégé are reported.
- **Human-Computer Interaction**: The collaboration between AI systems and human requests AI systems to be reliable, reactive, to reduce the worker's cognitive load, to be reproducible, transparent, in short to behave as a trustworthy co-worker. The major risk identified is that an AI system reproduces the same errors over and over again, leading to the famous saying "trust takes the stairs up and the elevator down".
- **Safety**: The main safety concerns raised by industrial partners are the potential errors of an AI system and their consequences. Guaranteeing precision and generalization, characterizing the uncertainty of answers are obvious needs. Several standards have been designated, depending on the industrial domain and reveal that industrial partners almost systematically work in standard controlled environment.
- **Lifecycle management**: Lifecycle management is an activity that seems to be little addressed and, sometimes, relegated to a single tool (MLFlow and Git are often used for the model) and sometimes ignored.
- **Requirements engineering**: Requirements are expressed in terms of performance, robustness, explainability, interpretability. When dealing with embedded systems, the usual constraints are mentioned: size, power consumption, real time performance. The methods and tools used for requirements engineering of AI components are the ones currently used by industrial partners for traditional components.
- **Design**: Formal methods, processes and methodologies and the supporting tools for AI components and AI-based systems are not often used in current practice. However, when companies integrate components from third parties (i.e. suppliers) they tend to ask for guarantees of some sort in the contracts.
- **Deployment**: Deployment for embedded systems is the main concern since deployment on large-scale infrastructures or cloud does not seem to be a subject of concern.
- **Supervision/Monitoring**: Generally speaking, the process of integrating the monitoring of AI-based systems does not seem mature. Various open-source tools are frequently mentioned (MLFlow, Kubernetes, Tensorboard, Prometheus, Grafana) as well as tools delivered with AI platforms (Azure, AWS Sagemaker), sometimes in-house tools monitoring specific system indicators are mentioned.
- **Certification**: It seems clear to everyone that certification of AI component will have a major impact on the current practices, for methodologies as well as for tools used in the development chain, essentially because there is little or no use of certification tools and methods for AI systems at the moment. Engineers are keen to apply such changes if the added-value is demonstrated. However, they ask to have some freedom of choice, not to be constrained by the requirements of certification tools and methods.

All these issues highlight today's roadblocks that need to be addressed to bridge the gap between Proof of Concepts and actual deployment of dependable (system of) systems

involving AI algorithms. We consider that those issues can only be tackled by a deep revamping of current engineering practices.

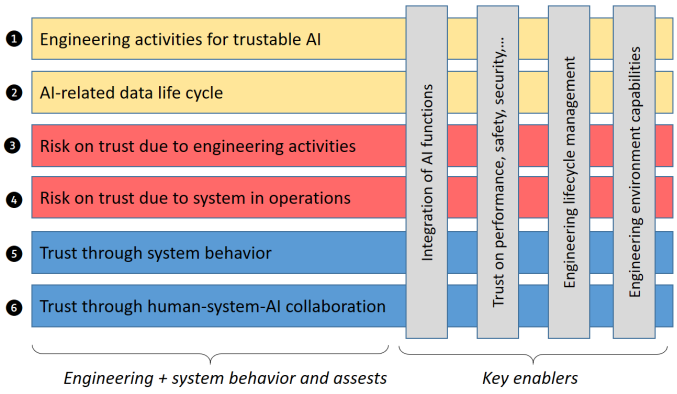

Fig. 2. Global view of the analysis framework

II. Engineering dependable industrial AI-based systems

A. Dependable AI-based systems

The operational exploitation of AI is relatively recent. It was determined by the spectacular improvements of algorithms and the hardware components executing those algorithms. Initially exploited for non-critical tasks showing no or a very low level of risk, building an AI system was essentially a matter of combining *ad-hoc* engineering practices with the objective of providing "usable" results in the most cost-effective way. When it comes to industrial critical systems, several additional constraints must be considered. First processes must be rationalized, justified, made reproducible, optimized, etc. Second, processes must ensure that the overarching properties of the system under design are actually satisfied with the appropriate level of confidence: (i) the defined intended behavior of the system is correct and complete with respect to the desired behavior, (ii) the implementation of the system is correct with respect to its defined intended behavior, under foreseeable operating conditions, (iii) any part of the implementation that is not required by the defined intended behavior has no unacceptable safety impact [1].

In the context of Confiance.ai program, we propose to reconcile these two approaches, namely learning from ad-hoc engineering practices on the different use cases on the one hand, and structuring reproducible processes for achieving appropriate level of confidence on the other hand. To achieve this reconciliation, we need an analysis framework able to capture and organize engineering contexts, constraints, activities, data, lifecycle, etc. concurrently under different viewpoints, in order to build a global and comprehensive model. Each viewpoint enriches others in an iterative and incremental, multi-viewpoint analysis.

B. Capturing and modeling the engineering processes of dependable AI-based systems

To ensure that AI-based systems will possess the necessary "Trust Properties", specific System Development activities (elaborated by project EC2) and IVVQ activities (elaborated by project EC6) are required. These activities will use a chain of elementary methods and tools, specified or recommended by projects EC3, EC4, EC5 and EC7.

The roles and the relations between the main artefacts produced by these activities are depicted in Fig. 3. The System Development Activities will produce Engineering Items. The IVVQ Activities will propose various Strategies to generate the Evidences showing that these Engineering Items possess the Trust Properties. The Trust Environment (produced by Confiance.ai project EC1) will orchestrate these System Development Activities and IVVQ Activities, and will store or reference the produced Engineering Items and Trust Evidences.

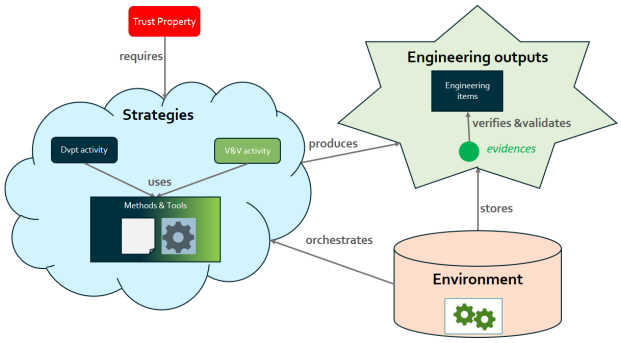

Fig. 3. Roles and relations between artefacts in Confiance.ai

The Trustable AI analysis framework designed by the Confiance.ai program will allow to elaborate the strategies for System Development Activities and for IVVQ Activities and will contribute to the specification of the Trust Environment.

The approach consists in:

- Defining analysis viewpoints, formalized in a modeling tool by a meta-model containing the definition of involved concepts and semantic relationships between these concepts. As of today, the analysis framework implementation is derived from the Capella toolbox[1]. Capella has been chosen to develop the first version of the tool so as to leverage our experience on this technology and its underlying system development methodology (ARCADIA). However, our approach is independent from the modeling technology, and other tools may well be used.
- Consolidating the methodological outputs of the Confiance.ai projects by analyzing their various aspects: engineering context, constraints, activities, data, lifecycle, etc.
- Formalizing the analyzed methods in a modeling tool, according to the meta-models of the considered viewpoints. The modeling will help ensuring that all methods are compatible with each other, and that the Confiance.ai

[1] See https://www.eclipse.org/capella

program will produce a consistent end-to-end process allowing the design of dependable AI-based systems.

The overall structure of the analysis framework is depicted on Fig.2.

The viewpoints for System Development (left part of the figure) are the following:

- 2 generic viewpoints:
  - **Engineering activities for trustable AI** (layer 1): Define the tasks to perform so as to specify, design, produce, deploy and operate an appropriate and trustable solution to a well understood need, involving AI techniques. An example of engineering activities viewpoint limited to ML algorithm engineering is shown Fig. 4.
  - **AI-related data life cycle** (layer 2): Identify major data required/produced by AI engineering, when they are produced/used, and how they evolve with time
- 2 viewpoints dedicated to risk on trust (ie. risk that the trust on the capability of the system to deliver the expected service is reduced or lost):
  - **Risk on trust due to engineering** (layer 3): identify major sources of bias or errors brought by other engineering activities to inputs and outputs of AI engineering and data
  - **Risk on trust due to system during operation** (layer 4): identify major sources of bias or corruption brought by other system components interacting with AI components in Operation
- 2 viewpoints dedicated to trust development and support:
  - **Trust through system behavior** (layer 5): define major system capabilities needed to ensure Trust in Operation
  - **Trust through Human/System/AI Collaboration** (layer 6): define expectations of humans stakeholders, their role and workshare with System AI, in delivering the expected services in a trustable manner.

4 transverse system viewpoints are also identified (right part of Fig.2):

- Integration of AI functions: characterize and address specific concerns related to integrating one or more AI functions together in system target context; deliver guidance on how to manage each concern,
- Trust on Performance, Safety, Security: define main needs, contributions and obstacles regarding Trust applied to AI decision performance, safety, and security of the global solution including AI,
- Engineering Lifecycle Management: define processes to revisit engineering choices and decisions according to evolution of context, environment and needs,
- Engineering Environment Capabilities: define the tooling support required to make trustable AI systems engineering feasible, scalable, efficient and secure.

Consistency of the content in all those viewpoints shall be checked.

The next section focuses on VV aspects in relation with these engineering viewpoints.

III. Coupling the engineering and V&V concerns

Verification and validation activities are essential contributors to trust. Indeed, those means are essentially aimed at providing evidences that the system will realize the intended function. Towards that goal, we propose to establish a clear, traceable, auditable, and *as formal as possible* relationship between the *engineering items* produced by an AI system engineering activity, the properties that those items must satisfy, and the activities providing evidences that those properties are actually satisfied. In the Confiance.ai program, this relationship is captured by means of *Assurance Cases* [4].

An Assurance Case provides a structured argument to justify certain properties (sometimes called "claims") about the system under design, based on evidences concerning both the system and the environment in which it operates. The objective is to demonstrate as rigorously as possible that if some evidences are provided then some claim is justified. This argument cannot be as rigorous as a mathematical demonstration[2], simply because it does not refer to mathematical entities and mathematical properties. Nevertheless, the objective is to make it as close as possible to what a mathematical demonstration would be. In particular, terms and properties must be defined as precisely as possible, hypothesis and assumptions must be clearly stated, etc.

A claim concerns the satisfaction of some property by the system (e.g. system-level properties such as safety, security, or item-level properties such as "completeness", "consistency", etc.). Building an argument consists to decompose the initial claim into sub-claims deemed easier to justify in a divide-and-conquer approach, down to the point where claims can be directly justified by showing evidences. During the construction of the argument, claims and properties may concern the whole system or some engineering items produced and used to engineer the system. As for the design of the system itself, building the Assurance Case of the system is not an ideal and strict top-down process that would start from some top-level property (e.g., "the system performs the intended function") and would be progressively decomposed into more and more primitive properties applicable to more and more primitive items. It is rather a combination of top-down and bottom-up approaches.

Assurance Cases are not a new practice. They actually derive from the now-well established practice of *safety cases* that are already required in some industrial domains (e.g. in the automotive domain with the ISO 26262 [2]). For the interested reader, the genesis of Assurance Cases is very well described in [5].

As stated earlier, the introduction of AI in industrial practices strongly changes well-established practices, in particular those related to certification [3], and many initiatives are currently working on updating or defining new practices addressing the specificity of AI (e.g., EUROCAE WG114, ISO/TC204

[2]It is not strictly *deductive.*

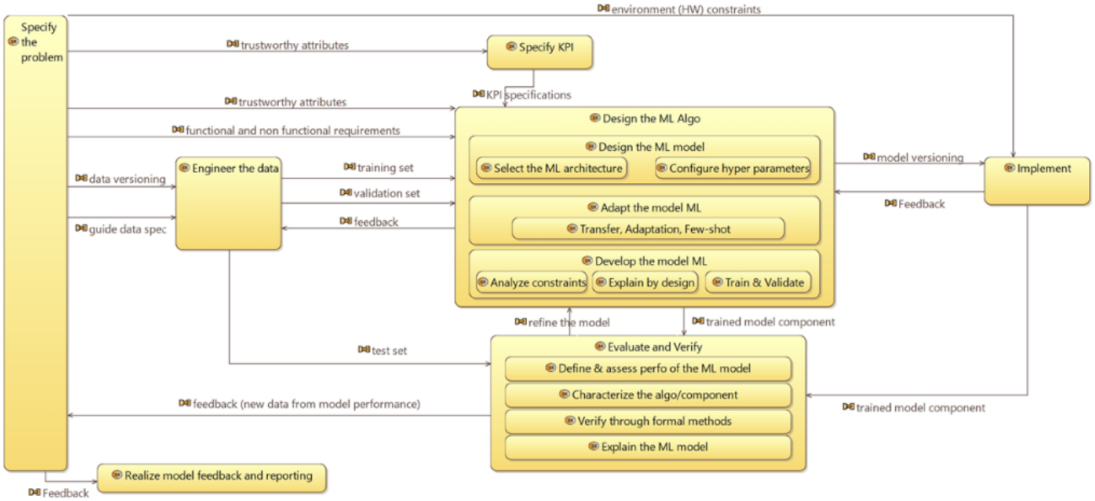

Fig. 4. Illustration of a 1st Capella model for ML Algo Engineering

WG14). Basically, most current practices are based on recommendations established on the basis of historical record, but as stated by Rushby *et al* in [5], "[...] one reason for looking beyond guidelines and toward assurance cases is to admit new methods of assurance [...] and new kinds of systems [...], so relevance of the historical record becomes unclear. "

It is worth noting that, in the context of Confiance.ai, Assurance Cases are strictly considered as a means to formalize the argumentation and build trust. It is a reference model from which other, specific representations, can be extracted. This includes, in particular, representations that will eventually be required by certification authorities. Our Assurance Cases are

- Not a verification and validation plan, but it may be used to build it.
- Not a Certification Standard, but it may be used to build one and, at least, it may be used to determine the activities to be carried out to comply with the existing ones. Traceability between the objectives / recommendations of standards and our Assurance Case is not yet addressed.

The methods identified in the Assurance Case to provide evidences are normally captured by verification and validation activities of the engineering process. Therefore, the overall process is the following:

- The design / development / deployment / etc. workflow is established and the artefacts involved in this - workflow are described
- Claims concerning the properties that those artefacts must possess are expressed
- Assurance cases are built to show how those properties will be assessed
- At the "bottom" of the argumentation (the leaves of the tree), one find evidences
- Evidences are brought thanks to some dedicated verification and validation activity
- Those activities are inserted in the workflow to give the complete picture.

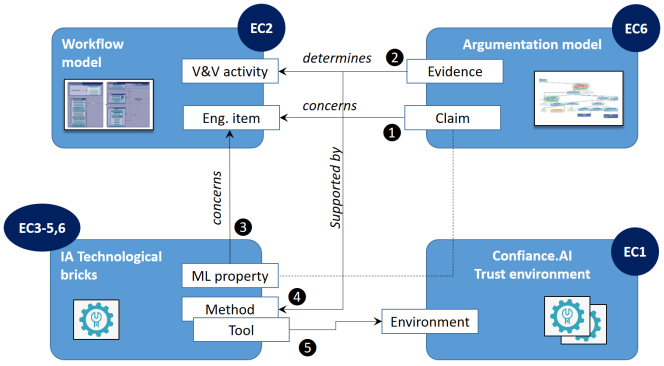

Fig. 5. From the workflow to the Assurance Cases and backward

The navigation between the Assurance Case model and the engineering process model is illustrated on Figure 5.

The Assurance Cases developed in Confiance.ai are fairly generic for they need to be applicable in different contexts (embedded system, production lines,...), industrial domains (aeronautics, space, automotive, etc.), and for different types of applications involving different types of sensors and algorithms, with different level of criticality.

The Assurance Cases capture this diversity using "strategies". For instance, the same claim about a given property P may be decomposed (or justified) in different manners, or strategy, according to the level of criticality of the system. This way, a claim about temporal determinism may be achieved empirically by means of measurements performed on the actual system in one strategy, and may be performed using complex static analysis techniques based on formal methods (e.g., abstract interpretation) in another strategy. In this example, the confidence on the engineering item will (normally) depends on the strategy.

At the end of the day, the objective is to use the model to help making the optimal choice considering (i) the additional confidence brought by the method on the capability of the system to perform its intended function, and (ii) the cost of

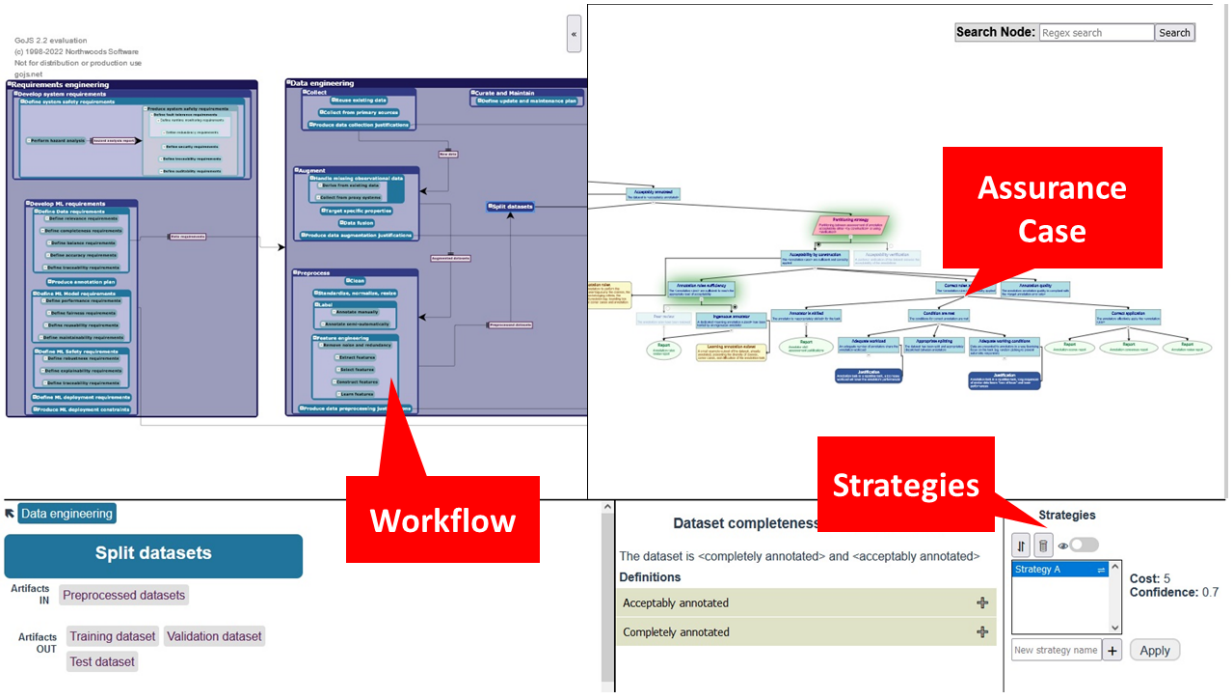

Fig. 6. Engineering and Assurance Case tooling

implementing the method.

To reach this objective, we are currently developing a tool that allows navigating between the engineering workflow and the Assurance Case. A screenshot of the tool interface is given on Figure 6. It shows the *workflow* (on the top left), the *Assurance Case* (on the top right), and the applied *strategy* (on the bottom right). Thanks to this tool, the user will eventually be able to (i) build a V&V strategy considering the risk level associated with errors affecting engineering item, cost and trust indicators associated with the production of evidences, and to (ii) display the resulting engineering workflow including V&V activities.

IV. Conclusion: From the Model to the Workbench

Adoption of AI in our industry raises many technical and non technical challenges and a huge effort is currently deployed to address these challenges and facilitate the early industrial adoption of AI in a *cost-effective* and *safe* way.

In the context of the Confiance.ai program, the most critical of those challenges have been identified, covering two main aspects: *dependability* to provide the capability to give a justified trust on the capability of the system to deliver the expected service, and *industrial efficiency* in order to ensure that dependability will be achieved in a cost-effective way.

The Confiance.ai program addresses most of the dimensions of the problem, from the provision new ML algorithms and techniques addressing the various dimension of trust, including explicability, fairness, temporal determinism, etc. But the engineering practices themselves must also be updated to account for the specificities of AI. Towards that goal, we propose a "Trustable AI Engineering Definition Framework" to build system development and V&V workflow integrating explicitly the various dimension of trust. The framework relies on a model-based approach involving a series of 10 viewpoints capturing the various aspects of system development including those related to data engineering, risk analysis, etc.

As of today, a meta-model capturing and organizing the concepts supporting the different viewpoints has been developed, and its implementation in the Capella environment is on-going.

In the next phase, our objective is (i) to populate the engineering model using the methods and tools developed or recommended by the different projects (EC3,4,5,7), (ii) to build the complete V&V argumentation with clear links between the workflow activities, the engineering items, evidences, and activities providing evidences, and (iii) bring together related methodology elements and adequate tooling to support collaboration of all engineering disciplines for trustworthy AI-based products over their life cycle. The models obtained in this first phase will be used to evaluate, validate and possibly correct the approach.

References

[1] Holloway, C. M. Understanding the Overarching Properties. Tech. Rep. NASA/TM-2019-220292, NASA, July 2019.

[2] ISO. 26262 – Road vehicles – Functional safety, 2011.

[3] Mamalet, F. *et al*. Machine Learning in Certified Systems. Tech. rep., IRT Saint Exupéry, Mar. 2021. https://hal.archives-ouvertes.fr/hal-03176080.

[4] Rushby, J. Assurance and Assurance Cases. In *Dependable Software Systems Engineering*, vol. 50 of *NATO Science for Peace and Security Series*. Oct. 2017, pp. 207–236.

[5] Rushby, J., Xu, X., Rangarajan, M., and Weaver, T. L. Understanding and evaluating assurance cases. Research Report NASA/CR–2015-218802, NASA, Sept. 2015.