GenAI for Automated Form Filling:

generating data-aware executable artefacts deployable at scale

`exadaps`: **e**valuation of e**x**ecutable **a**rtifacts for **da**ta **p**roce**s**sing

Submission date	Apr 30, 2026
Team leader	Michael Accetto
Contact email	michael.accetto@dtsc.be
Domain of application	Industry 4.0
Scientific theme	Generative & Multimodal AI
Project name	`exadaps` – **e**valuation of e**x**ecutable **a**rtifacts for **da**ta **p**roce**s**sing
Original form title	TReC 2026 Project Proposal Submission Form
Updated	Apr 30, 2026

1 Team leader and team composition

Profile of the team leader(s) and expected team composition

Michael Accetto serves as TRACEFORM-AI project lead within the DTSC BRAIN Lab. He has delivered a production RAG system for BNP Paribas Fortis in a regulated-industry environment and leads the Heerlen cross-border partnership with CIED on document intelligence. His expertise covers agentic AI systems, regulated-document processing and the industrial development and deployment of data driven solutions.

Aditya Mukhopadhyay holds an MSc and enters the doctoral track on TRACEFORM-AI. His work addresses hierarchical prompt evolution for automated form filling, with a focus on separating stochastic synthesis, where LLMs analyse and generate, from deterministic execution through compiled filling programs.

Potential team members identified?	Yes
Team members identified	Academic co-leadership. An academic co-leadership with a group active in compound AI systems or document intelligence is being explored. Natural candidates include UCLouvain and UNamur NLP teams. This configuration would strengthen the academic anchoring of the proposal. We welcome participants from natural language processing, compound AI systems, document intelligence, compliance and audit practice, regulatory law, and UX design for accountability interfaces.

2 Abstract

Administrative and regulatory form filling remains a major bottleneck in bureaucracy-heavy industries. Many organisations already hold the required information in structured systems, but still spend substantial time transferring that information into heterogeneous external forms. This is especially common in compliance, QA, ESG, supplier qualification and certification workflows,

where the same underlying business data must be repeatedly reformatted for different customers, auditors or public bodies.

Classical automation works when the target form is stable and machine-readable, but breaks down when fields are visually defined, annotations are missing or unreliable, or the form arrives as a scan or image. Conversely, asking a language model or vision-language model to fill the form directly may handle more variation, but gives limited guarantees of reproducibility, auditability and reuse. This is problematic in regulated settings, where outputs must be traceable and errors must be explainable.

The technical problem addressed by this project is therefore: given an empty form F, a knowledge map M, and labelled examples of correct filling behaviour, can a compound AI system generate a deterministic program P_F that reliably produces filled forms, F_{filled}. The program must decide where answers go, which values from M should be selected, whether the output is text or a tick mark, how the value should be placed, and how formatting constraints should be respected.

The project lead will provide a prepared benchmark dataset containing forms with labelled fields, expected filling-box answers, reference outputs and evaluation metadata. The benchmark will combine synthetic forms with adapted public form-understanding material, including modified FUNSD-style cases in which answer regions are removed or masked. All instances follow the same task: infer the correct filling behaviour and generate a program able to produce the completed form.

The evaluation function μ is central to the research design. It measures value selection, field placement, output type, formatting, visual coherence, program validity and execution success. It also returns traces describing failures such as wrong mappings, missing fields, invalid code, incorrect tick marks or poor placement. These traces allow participants to test iterative improvement: generate P_F, execute it, evaluate it, feed the errors back into the system, and generate a better program.

The two-week camp is scoped around a prototype rather than a deployable product. Researchers will build an initial AI system that predicts filling behaviour across multiple forms, generates executable programs, benchmarks them, and repeats the process to improve transferability to new forms. A stretch goal is form recognition: identifying whether a new form belongs to a known family and applying the appropriate previously generated program.

3 Background information and problem statement

Regulatory and compliance workflows often end with a deceptively simple task: filling a form using structured data that already exists elsewhere. In practice, this requires identifying the fields of the form, selecting the relevant datum from a business knowledge map M, and inserting the value in the correct visual location. Forms may arrive as native PDFs, scanned PDFs, images, or other weakly structured formats, making the task difficult to solve with fixed rules.

This project frames Automated Form Filling (AFF) not as direct document generation, but as AI-assisted program generation. A compound AI system S_θ receives an empty form F and a structured knowledge map M. Its task is to generate a form-specific executable program P_F:

$$P_F = S_\theta(F, M)$$

Once generated, P_F is executed to produce the completed form:

$$F_{\text{filled}} = P_F(F, M)$$

The generated program P_F encodes how fields are identified, how values from M are mapped to

those fields, and how the values are visually inserted into the document, including text, tick marks, bounding boxes, formatting and approximate text size. The compound AI system is used for the difficult synthesis step; the final filling operation is performed by a deterministic artefact that can be executed, inspected, tested and reused.

The quality of P_F is measured by an evaluation function μ, which scores the filled output against labelled reference metadata and returns both scores and traces. These traces can then be fed back into S_θ, creating a self-improving loop for program generation.

During the two-week Research Camp, participants will prototype and benchmark this approach using a prepared AFF dataset provided by the project lead. Proposed strategies include VLM-based form analysis, execution-based repair, and prompt optimisation using frameworks such as DSPy and GEPA. The expected outcome is a reusable benchmark and prototype harness for VLM-guided generation of deterministic form-filling programs. While AFF is the concrete use case, the broader research contribution is the study of how compound AI systems can generate executable, auditable data-processing artefacts from weakly structured inputs, provided a robust evaluation method.

4 Research objectives

The project explores Automated Form Filling as AI-assisted program generation. Rather than asking a model to fill a document directly, participants will build a system that, given a form F and a structured knowledge map M, generates a form-specific program P_F capable of producing the completed form.

Objective 1: Predict how a form should be filled

Participants will develop an AI system that analyses individual forms and predicts where each answer should go. This includes selecting the right datum from M, identifying the correct field or answer box, deciding whether the output should be text, a tick mark, or another mark, and estimating placement details such as bounding box, text size, and visual alignment.

The project lead will provide a dataset of documents with labelled fields and expected filling-box answers, so that participants can focus on modelling. The evaluation protocol will also be offered in the form of a code repository with all participants.

Objective 2: Generate executable programs that produce filled forms

Participants will use the predicted field mappings to augment the AI system with a generation step. The output will be a deterministic program P_F for each form or form family. When executed, P_F should take the empty form and the knowledge map as input and produce the filled form. The project leader will challenge participants by proposing different strategies to iterate over the output of P_F to improve it and the system with it.

This objective tests the core hypothesis of the project: that VLMs and compound AI systems can be used for the synthesis step, while the final form-filling operation is handled by an executable artifact that is deterministic, inspectable, and reusable.

Objective 3: Benchmark, repair, and improve transferability

The generated programs will be evaluated against synthetic benchmark forms and a modified subset of FUNSD where answer fields have been removed. Evaluation will measure whether values are selected correctly, placed in the right field, visually aligned, formatted correctly, and reproducible across runs.

The benchmark will return both scores and traces. These traces will be fed back into the AI system

to adapt the instructions contained and improve later program generation, creating a self-improving loop. Participants will repeat the process across multiple forms to test transferability: whether experience on earlier forms improves performance on new forms.

As a stretch objective, participants may also build a form-recognition module that identifies the form type and applies the appropriate previously generated program when available.

5 TRAIL Factory brick

Reusable brick planned? Yes

The TRAIL Factory brick will be a reusable self-improving executable-generation harness. Its purpose is broader than form filling: given a task dataset, an evaluation function μ, and scoring metadata, the brick can generate programs for data-processing tasks.

In the Research Camp, AFF is the concrete benchmark: the system receives an empty form F and a structured knowledge map M, then generates an executable P_F that produces the filled form. However, the reusable contribution is the general loop: generate a candidate program, execute it in a controlled environment, score the output against reference metadata, extract failure traces, and feed those traces back into the AI system to improve the next generation. The same architecture can later be adapted to other structured transformation tasks, such as document completion, report generation, data migration, validation workflows, compliance checks, or domain-specific ETL pipelines.

The brick will include the executable-generation harness, a benchmark template, evaluation and trace formats, baseline generators, repair loops, documentation, and an initial AFF dataset demonstrating the approach. It will support both scalar scores and richer textual traces, enabling systems such as DSPy/GEPA-style optimizers to improve program generation from feedback rather than from manual prompt tuning alone.

The intended users are AI researchers working on compound AI systems, program synthesis, document intelligence, evaluation and self-improving workflows; TRAIL teams seeking reusable research infrastructure; and industrial innovation teams that need auditable automation rather than opaque one-shot model outputs. In a company setting, the brick could become the basis for a process where repeated data-processing tasks are converted into validated, deterministic executables that are cheaper, safer and easier to audit than repeated generative-AI calls.

6 Project dataset

The project lead will provide a prepared benchmark dataset for Automated Form Filling before the Research Camp. The dataset will contain forms with labelled fields, structured knowledge maps M, expected filling-box answers, and reference annotations indicating where and how each answer should be inserted. The forms will cover text fields, tick boxes, dates, numeric entries, repeated sections, and simple tables, with variation in layout quality and visual complexity.

The benchmark will combine purpose-built synthetic forms with adapted public form-understanding material, including a modified subset of FUNSD, RVL CDIP and XFUND. All instances will follow the same task structure: given an empty form F and a knowledge map M, the system must predict which data from M belongs in each field, determine the required output type and visual placement, and generate a program P_F capable of producing the filled form.

For each form, the dataset will include labelled field metadata such as field type, visual anchor or bounding box, expected value source in M, formatting constraints, and reference placement. In the modified FUNSD cases, answer regions will be removed or masked so that the task becomes genuine form filling rather than form understanding alone.

The dataset will be split into development and held-out evaluation sets. The development split will support system construction, prompt optimisation, and execution-based repair. The held-out split will test whether the generated programs transfer to unseen forms or form families. The benchmark will return both scores and traces, covering value selection, placement, formatting, output type, program validity, execution success, and visual coherence. These traces will be fed back into the AI system to support iterative improvement.

7 Detailed work plan

Before the camp, the project lead and co-lead will prepare the AFF benchmark dataset, the task specification, the initial evaluation procedure and example forms. This preparation is meant to make the two-week work feasible: participants will not need to collect data or define the problem from scratch, but can start immediately from labelled forms, knowledge maps, reference answers and benchmark tooling.

Period	**Activities and milestone**
Day 1	Align the team on the research framing and execution target. Participants will review the dataset, the definition of F, M and P_F, the evaluation metadata, and the expected output format. The team will then be split into three coordinated tracks: form analysis and VLM prompting; program generation and deterministic execution; evaluation, traces and optimisation.
Days 2–3	Focus on the first end-to-end baseline. The team will implement a minimal pipeline that reads an empty form F and knowledge map M, predicts answer locations and output types, and produces a simple filled form. This first baseline may be direct VLM prediction rather than full program generation. Its role is to establish the benchmark interface and expose the main failure modes early.
Days 4–5	Focus on program generation. Participants will define a restricted representation for P_F, for example a small Python subset, JSON workflow or domain-specific instruction format. The generated program must be executable, deterministic and inspectable. By the end of Day 5, the target milestone is a working path from form and knowledge map to generated program, executed filled form and evaluation result.
Days 6–7	Focus on field-level evaluation and trace generation. The benchmark will score whether the system selected the correct value from M, placed it in the right field, used the correct output type, respected formatting constraints and produced a visually coherent result. The evaluator will also return traces describing concrete errors such as missing fields, wrong mappings, invalid programs, poor placement or failed execution. The end-of-week milestone is a first benchmark table comparing direct filling with generated-program filling on the development split.
Days 8–9	Focus on iterative improvement. Execution traces and benchmark feedback will be fed back into the AI system to repair failed programs and improve generation prompts. Participants will test whether VLM-based form analysis, execution-based repair and DSPy or GEPA-style prompt optimisation improve program validity, placement accuracy and transfer to new forms.

Period	Activities and milestone
Days 10–11	Focus on transferability. The team will repeat the generation process across multiple forms and form families, including synthetic forms and modified FUNSD-style cases where answer regions have been removed or masked. The goal is to measure whether the system improves beyond one-off form solving: learned prompts, representations or filling primitives should help on unseen forms.
Day 12	Reserved for optional extensions and consolidation. If the core pipeline is stable, participants may add a form-recognition module that identifies whether a new form belongs to a known family and applies the appropriate previously generated program. Otherwise, the day will be used to strengthen evaluation, repair or documentation.
Day 13	Focus on packaging the TRAIL Factory brick. The team will clean the repository, finalise the dataset schema, document the benchmark procedure, freeze baseline configurations and prepare example generated programs. The benchmark should be usable by a future researcher without requiring access to proprietary data or industrial systems.
Day 14	Used for the final demonstration and presentation. The demo will show an unseen or held-out form, a knowledge map, the generated program P_F, the executed filled form, the evaluation score and the trace explaining successes and failures. The final deliverables will be the AFF benchmark dataset, the prototype harness, baseline results, generated-program examples, trace-based evaluation tools and a short roadmap for follow-up research.

8 Bibliographic references

1. Khattab, O. et al. (2023). DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. arXiv:2310.03714.
2. Opsahl-Ong, K. et al. (2024). Optimizing Instructions and Demonstrations for Multi-Stage Language Model Programs. arXiv:2406.11695.
3. Agarwal, R. et al. (2024). GEPA: Genetic Prompt Evolution for LLM-Compound Systems. arXiv.
4. Yuksekgonul, M. et al. (2024). TextGrad: Automatic Differentiation via Text. arXiv:2406.07496.
5. Jaume, G., Ekenel, H. K. and Thiran, J.-P. (2019). FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In ICDAR Workshops.
6. Park, S. et al. (2019). CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In NeurIPS Workshops.
7. Mathew, M., Karatzas, D. and Jawahar, C. V. (2021). DocVQA: A Dataset for VQA on Document Images. In WACV.
8. Xu, Y. et al. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In KDD.
9. Xu, Y. et al. (2022). LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In ACM MM.
10. Kim, G. et al. (2022). OCR-free Document Understanding Transformer (Donut). In ECCV.
11. Lee, K. et al. (2023). Pix2Struct: Screenshot Parsing as Pretraining for Visual Language Understanding. In ICML.

12. Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In NeurIPS.

13. Nissenbaum, H. (2010). Privacy in Context: Technology, Policy, and the Integrity of Social Life. Stanford University Press.

14. Selbst, A. D. et al. (2019). Fairness and Abstraction in Sociotechnical Systems. In FAT*.

15. Koshiyama, A. et al. (2022). Towards algorithm auditing: managing legal, ethical and technological risks of AI. Journal of Banking Regulation.

16. European Commission (2024). Regulation on Artificial Intelligence (EU AI Act), Regulation (EU) 2024/1689.

9 Multidisciplinarity, confirmation, and additional comments

Multidisciplinarity between STEM & SSH? Yes

STEM-SSH interface

Work by Nissenbaum (2010) on contextual integrity, by Selbst et al. (2019) on fairness in ML systems, and by Koshiyama et al. (2022) on AI auditing, provides the conceptual foundation for treating regulatory-form filling as a socio-technical system. The SSH track of this project operationalises these frameworks into concrete traceability metrics and audit protocols.

Presence confirmation I/We agree and confirm that the Team Leader will be present for the full duration of TReC'26 if the project is selected (August 24th - September 4th, 2026, Lausanne, Switzerland).

Additional comment

This project aligns with three TReC 2026 scientific themes: Natural Language Processing, Generative and Multimodal AI, and Trustworthy AI. Domain coverage spans Industry 4.0 and Manufacturing, through traceability and compliance workflows, and Education and Public Policy, through administrative automation and audit frameworks.

This project is not a continuation of previous TReC editions. No prior camp has addressed hierarchical prompt evolution for regulatory form filling with the breadth of baselines and metric consistency proposed here.

Research scope and industrial boundary: The research-camp deliverable is strictly the public evaluation framework: benchmark corpus, metric harness, baseline implementations, accountability instrument. Any industrial application remains outside the scope of this proposal. The DTSC team brings practical grounding through its regulated-industry experience, but the workshop output is designed for open community use without product-side entanglement.

Submission IP 91.178.100.194

Submission ID 6533195624915285735

Last update date