PharmaSUG 2026 - Paper SS-261

Closing the Loop: Validating AI-Generated SDTM Mappings using CDISC CORE and Synthetic Data

Pietro Belligoli, Constantin Weberpals, and Yarhy Flores Lopez, Technical University of Munich

ABSTRACT

Generative AI approaches to SDTM mapping often focus on prediction accuracy but lack tight integration with conformance validation, leaving specification and logic errors undetected until late in the data collection or analysis phases. We describe a closed-loop framework that combines generative AI SDTM mapping with automated CDISC CORE conformance validation using synthetic data to enable earlier detection of specification issues. The proposed methodology extracts detailed field-level metadata from CRFs and complementary non-EDC inputs, including data transfer specifications, and generates SDTM variable mappings using large language models grounded in SDTMIG specifications. To proactively assess conformance, the system generates synthetic SDTM datasets based on the proposed mappings and executes CDISC CORE validation rules against these datasets. Validation failures automatically trigger an iterative refinement loop, in which the AI model revises mappings and derivation logic based on specific CDISC CORE rule violations and contextual feedback. This closed-loop process continues until conformance criteria are satisfied or remaining non-conformance is identified as expected and appropriately documented. The framework targets error categories that are difficult to identify through manual review alone, including controlled terminology mismatches, context-dependent Value Level Metadata violations, and logic-based derivation errors. By integrating mapping generation and conformance validation into a single automated workflow, this approach demonstrates the feasibility of shifting CDISC compliance earlier into the specification process, reducing downstream rework and improving the overall quality and reliability of SDTM deliverables.

INTRODUCTION

The transformation of raw clinical trial data into CDISC Study Data Tabulation Model (SDTM) format is a critical step in the regulatory submission pipeline. This process requires detailed mapping specifications that define how source variables from Case Report Forms (CRFs) and other data collection instruments are converted into standardized SDTM domains and variables. Mapping specifications must comply with CDISC SDTM Implementation Guide (SDTMIG) rules, use appropriate Controlled Terminology (CT), and satisfy conformance rules enforced by validation engines such as the CDISC Open Rules Engine (CORE).

Traditionally, SDTM mapping is a manual, labor-intensive process performed by clinical data standards specialists. Mappings are typically authored in specification documents, implemented in SAS or R code, and then validated using tools such as Pinnacle 21 or CDISC CORE only after datasets have been produced. This late-stage validation approach creates an expensive rework cycle: errors discovered during validation require tracing the issue back through code, specifications, and sometimes the CRF design itself.

The emergence of large language models (LLMs) has introduced the possibility of automating the initial generation of SDTM mapping specifications. Recent work in the industry has demonstrated that LLMs can produce first-draft mappings, derivation logic, and code scaffolding when provided with appropriate metadata context. However, most existing AI-assisted mapping approaches treat mapping generation and conformance validation as separate, sequential activities. The AI produces a mapping, which is then manually reviewed and eventually validated, reproducing the same late-detection pattern as the traditional workflow.

This paper presents a closed-loop framework that integrates AI-driven SDTM mapping generation with automated CDISC CORE conformance validation using synthetic data. By generating synthetic SDTM datasets from proposed mappings and immediately validating them against CDISC CORE rules, the

system is designed to detect specification issues before any real data is transformed. Validation failures feed back into the AI model, which revises its mappings in an iterative refinement loop until conformance criteria are satisfied or remaining non-conformance is documented as expected. This approach shifts CDISC compliance from a late-stage quality gate to an integral part of the specification process itself.

BACKGROUND

SDTM MAPPING CHALLENGES

SDTM mapping involves translating source data from EDC systems, laboratory feeds, ePRO instruments, and other sources into standardized tabular datasets organized by CDISC-defined domains, such as DM (Demographics), AE (Adverse Events), LB (Laboratory Test Results), and VS (Vital Signs). Each variable within these domains has defined attributes, including data type, controlled terminology, and derivation rules specified in the SDTMIG.

Common sources of mapping errors include incorrect domain assignment, misapplication of controlled terminology, inconsistent derivation logic for timing variables (such as EPOCH assignment), missing or mismatched Value Level Metadata (VLM) for findings domains, and cross-variable inconsistencies such as LBTESTCD values that do not correspond to their expected LBTEST labels or units. These errors are often subtle and context-dependent, making them difficult to detect solely through manual specification review.

CDISC CORE

The CDISC Open Rules Engine (CORE) is an open-source validation engine developed under the CDISC Open-Source Alliance (COSA). CORE executes standardized conformance rules against SDTM datasets in a consistent, transparent, and reproducible manner. Rules cover structural requirements (variable presence, data types, label compliance), controlled terminology usage, cross-domain consistency, and logical constraints specified in the SDTMIG.

CORE can be invoked both as a command-line tool against XPT files and as a Python library against in-memory data structures. This programmatic interface is what enables integration into an automated pipeline. The rules engine produces structured output identifying each rule violation by rule ID, affected domain, variable, and record, providing the contextual information necessary for targeted remediation.

LLMs FOR SDTM MAPPING AUTOMATION

Recent PharmaSUG, PHUSE, and CDISC Interchange publications have explored the use of LLMs for clinical data standards tasks, including generating SDTM mapping specifications, assigning controlled terminology, and scaffolding SAS and R code. These approaches typically provide the LLM with CRF metadata, SDTMIG specifications, and example mappings as context, then prompt the model to generate mapping specifications for new source data.

While LLMs have shown strong performance on structural mapping tasks, such as assigning source fields to correct SDTM domains and variables, they remain prone to errors in areas requiring precise application of CDISC rules: controlled terminology mismatches, incorrect derivation logic for calculated fields, and hallucinated variable names or codelists. Without a mechanism to validate the generated mappings against authoritative CDISC rules, these errors propagate downstream and are only caught during late-stage validation.

METHODOLOGY

ARCHITECTURE OVERVIEW

The closed-loop framework consists of five interconnected components that operate in sequence within an automated pipeline:

(1) Metadata Extraction ingests eCRF definitions and non-EDC source documentation (e.g., data transfer specifications) to produce a structured representation of source fields and their attributes. (2) AI-Driven Mapping Generation uses an LLM grounded in SDTMIG specifications to produce candidate SDTM

variable mappings and derivation logic. (3) Synthetic Data Generation creates realistic SDTM datasets based on the proposed mappings. (4) CDISC CORE Validation executes conformance rules against the synthetic datasets. (5) Iterative Refinement feeds validation failures back to the AI model for targeted mapping revision. Figure 1 illustrates the data flow between these components.

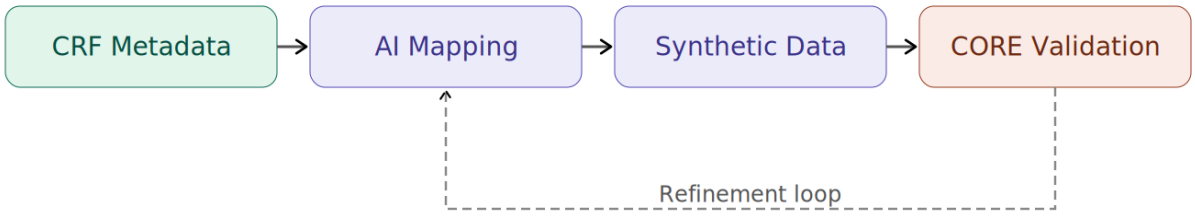

Figure 1. Closed-Loop Framework Architecture

CRF METADATA EXTRACTION

The first stage extracts structured, field-level metadata from clinical data collection instruments. The system processes CRF-annotated documents (aCRFs), EDC export specifications, and data transfer agreements. For each source field, the extractor captures the field name, data type, associated codelist or allowable values, collection context (visit, form, section), and any conditional or branching logic that governs when the field is populated.

Complementary non-EDC sources, such as central laboratory data specifications, ePRO vendor transfer files, and external data feeds, are handled by a configurable parser that maps vendor-specific metadata schemas to a common internal representation. This normalization step is critical because SDTM mappings must account for all data sources that contribute records to a given domain, not only EDC-collected fields.

The output of this stage is a JSON-structured metadata catalog that serves as the primary input context for the LLM in the subsequent mapping stage.

AI-DRIVEN SDTM MAPPING GENERATION

The mapping generation component uses a large language model (in the current implementation, Gemini 3.1 Pro) prompted with the extracted metadata catalog, the relevant SDTMIG domain specifications, controlled terminology codelists, and a set of mapping conventions that encode organizational standards. The prompt instructs the model to produce, for each target SDTM variable, a mapping specification that includes the source field(s), derivation logic (if applicable), controlled terminology mapping, and any Value Level Metadata conditions.

To improve mapping quality and reduce hallucination, the system employs a retrieval-augmented generation (RAG) approach. SDTMIG text, controlled terminology tables, and example mappings from prior studies are embedded and indexed. At inference time, the most relevant specification passages and examples are retrieved and included in the prompt context, grounding the model's output in authoritative CDISC documentation.

The model's output is a structured mapping specification in JSON format that defines, for each SDTM domain and variable, the source-to-target transformation. This structured output serves as both the human-readable specification and the input for synthetic data generation.

SYNTHETIC DATA GENERATION

Rather than waiting for real study data to be transformed, the system generates synthetic SDTM datasets directly from the proposed mapping specifications. The synthetic data generator creates records that exercise the mapping logic by producing values that span the expected range for each variable, including boundary conditions, null handling, and controlled terminology values.

For findings domains such as LB, VS, and EG, the generator produces records with realistic combinations of test codes, test names, result values, and units drawn from the controlled terminology specified in the mapping. For event domains such as AE and MH, the generator creates records with appropriate timing variables, severity grades, and coded terms. Cross-domain relationships (such as subject identifiers, visit sequences, and reference dates from DM) are maintained to ensure that cross-domain validation rules execute correctly.

The generator is designed to produce data that deliberately probes common failure modes: it includes edge-case timing values, mixed-case terminology entries, and intentional gaps in optional fields to ensure that validation rules are exercised comprehensively.

CDISC CORE VALIDATION

The synthetic datasets are validated against CDISC CORE rules for the applicable SDTMIG version. The system invokes the CORE engine programmatically via the Python library interface, enabling validation to run within the pipeline without writing intermediate XPT files.

The validation produces a structured results object containing, for each rule violation: the CORE rule ID, the severity level, the affected domain and variable(s), the record identifier(s), and a description of the violation. This structured output is essential for the refinement loop, as it provides the AI model with specific, actionable feedback about what aspect of the mapping produced a non-conformant result.

ITERATIVE REFINEMENT LOOP

When CDISC CORE validation produces failures, the system enters an iterative refinement cycle. The validation results are formatted into a structured prompt context that includes the original mapping specification, the specific rule violations detected, and the CDISC CORE rule definitions for the violated rules. The LLM is then asked to analyze the violations and propose revised mappings that address the identified issues.

The refinement prompt is designed to encourage targeted revisions rather than wholesale regeneration. For each violation, the model must explain its diagnosis of the root cause and specify the minimal change to the mapping specification required to resolve it. This approach preserves correct portions of the mapping while addressing specific conformance gaps.

After the model produces revised mappings, the synthetic data generation and CORE validation steps are repeated. This loop continues until one of three exit conditions is met: (1) all CORE rules pass, (2) a configurable maximum number of iterations is reached, or (3) remaining violations are classified by the model as expected non-conformance requiring documentation rather than mapping changes. The third condition is important because some CORE rules may legitimately fire for certain study designs or data patterns, and these need to be documented in the Clinical Study Data Reviewer's Guide (cSDRG) rather than resolved through mapping changes.

To prevent context bloat and endless iterative loops, the framework implements a triage layer that deduplicates cascading CDISC CORE errors by root cause before feeding them back to the LLM. A single mapping error, such as an incorrect controlled terminology value for LBTESTCD, can trigger multiple downstream violations across related variables (LBTEST, LBORRESU, LBSTRESC), and presenting all of these individually would consume prompt context without improving the diagnostic signal. The triage layer groups violations by their likely root cause and presents the LLM with a consolidated set of issues to resolve. Additionally, the framework incorporates a "circuit breaker" mechanism that automatically quarantines persistent failures after a configurable number of unsuccessful correction attempts. Quarantined violations are flagged for human-in-the-loop review, at which point a domain expert can determine whether the issue reflects a genuine mapping error requiring manual intervention or an expected non-conformance that should be classified and documented for the cSDRG. This safeguard ensures that the automated loop does not cycle indefinitely on issues that exceed the LLM's reasoning capacity while preserving a structured path to resolution.

TARGET ERROR CATEGORIES

The closed-loop framework is designed to detect several categories of SDTM mapping errors commonly encountered in practice and difficult to identify solely through manual specification review. This section describes these error categories and illustrates how the framework's combination of synthetic data generation and CDISC CORE validation is positioned to address them.

CONTROLLED TERMINOLOGY MISMATCHES

LLMs generating SDTM mappings may produce variable values that are semantically correct but do not exactly match CDISC Controlled Terminology (CT) submission values. For example, a vital signs mapping might use a test code derived from the CRF field label rather than the standardized CDISC CT value, or an adverse event severity term might use a synonym rather than the canonical codelist entry. CDISC CORE includes rules that validate variable values against the applicable CT codelists, and the refinement loop can correct these mismatches by referencing the authoritative CT provided in the RAG context.

VALUE LEVEL METADATA VIOLATIONS

Findings domains such as LB, VS, and EG require that certain variable values (such as units, methods, and specimen types) are consistent with the specific test code in each record. For instance, a hemoglobin result should carry units of "g/dL" rather than "g/L," and a body temperature measurement should use "C" or "F" rather than a unit appropriate for a different vital sign. These Value Level Metadata constraints are context-dependent, and the correct value depends on the combination of test code and variable, making them particularly prone to LLM errors and difficult to catch through flat specification review. CDISC CORE rules that check TESTCD-to-unit and TESTCD-to-method relationships are designed to catch these violations when executed against synthetic data that includes realistic combinations of test-level values.

DERIVATION LOGIC ERRORS

Derived SDTM variables such as --DY (study day), EPOCH, and duration fields require precise calculation logic that accounts for edge cases, including same-day events, missing dates, and partial date imputation. LLMs are known to produce derivation logic that works for typical cases but fails at boundaries; for example, an EPOCH derivation that incorrectly handles subjects whose treatment start date coincides with an observation date. By deliberately generating synthetic data with boundary-condition records, the framework can expose these edge-case failures through CORE rules that check the consistency of derived variables against their source values and cross-domain relationships (such as SE domain elements and domain-level EPOCH values).

DISCUSSION

The proposed framework addresses a fundamental gap in current AI-assisted SDTM mapping workflows: the separation between mapping generation and conformance validation. By integrating these activities into a single closed-loop process, the framework is designed to eliminate the costly cycle of producing datasets, discovering validation failures, tracing issues back to specifications, correcting mappings, regenerating datasets, and re-validating.

The synthetic data generation component is central to this approach. Without synthetic data, CDISC CORE rules cannot execute because they require actual data records to validate against. The quality of the synthetic data directly affects the validation coverage: data that does not exercise edge cases or probe known failure modes will not trigger the rules that detect those failures. This motivates the generator's deliberate inclusion of boundary-condition values and controlled terminology edge cases.

An important design consideration is how to handle expected non-conformance. Not all CORE rule violations indicate mapping errors; some rules may legitimately fire for certain study designs, therapeutic area conventions, or sponsor-specific practices. The framework accommodates this by supporting the classification of violations as expected and by facilitating their documentation for inclusion in the Clinical Study Data Reviewer's Guide (cSDRG). This mirrors industry-standard practice, in which validation findings are reviewed and documented rather than universally resolved.

A practical consideration is computational cost. Each iteration of the loop requires LLM inference, synthetic data generation, and CORE validation execution. However, because these operations are performed against synthetic data at the specification stage, rather than against full production datasets, the data volumes are modest, and each iteration is expected to complete in minutes rather than the hours or days typical of traditional rework cycles.

LIMITATIONS AND FUTURE WORK

Several limitations of the current framework design should be acknowledged. First, the quality of synthetic data is bounded by the mapping specification itself; if a mapping omits a variable entirely, the synthetic data will not include it, and the corresponding CORE rules will not fire. Second, CDISC CORE rule coverage continues to evolve; the framework's effectiveness depends on the comprehensiveness of the available rule set for the target SDTMIG version. Third, some mapping decisions are inherently context-dependent and require knowledge of sponsor conventions, protocol-specific definitions, or therapeutic area practices that cannot be fully captured in CDISC specifications alone. The framework's design accommodates this by allowing human reviewers to intervene at any point in the loop. Fourth, the current design does not yet incorporate Define-XML generation and validation, which represents an important additional layer of conformance checking.

Future work will focus on implementing and evaluating the framework across multiple domains, therapeutic areas, and study designs. Additional planned extensions include Define-XML validation support, integration with metadata repositories, and evaluation of different LLM architectures for the mapping and refinement stages.

CONCLUSION

This paper has described a closed-loop framework that integrates AI-driven SDTM mapping generation with CDISC CORE conformance validation using synthetic data. The framework proposes shifting conformance validation from a late-stage quality gate to an integral, automated component of the specification process. By detecting mapping errors before real data is transformed, the approach aims to reduce downstream rework, improve specification quality, and provide a structured mechanism for documenting expected non-conformance. The combination of LLM-based mapping generation, synthetic data probing, and iterative CORE-driven refinement represents a practical path toward tighter integration of AI automation and CDISC compliance in the SDTM development lifecycle.

REFERENCES

1. Clinical Data Interchange Standards Consortium (CDISC). 2022. Study Data Tabulation Model Implementation Guide: Human Clinical Trials v3.4. Available at: https://www.cdisc.org/standards/foundational/sdtmig/sdtmig-v3-4

2. Clinical Data Interchange Standards Consortium (CDISC). 2025. CDISC Open Rules Engine (CORE). GitHub Repository. Available at: https://github.com/cdisc-org/cdisc-rules-engine

3. CDISC Open-Source Alliance (COSA). 2024. CORE Conformance Rules for SDTM. Available at: https://www.cdisc.org/cosa

4. Ganapathy, R. 2024. "Introducing sdtm.oak." Pharmaverse Blog. Available at: https://www.lexjansen.com/phuse-us/2025/pp/PAP_PP31.pdf

5. Thukral, A., Bhardwaj, S. 2025. "Clinical Data Transformation: AbbVie's AI Journey." PharmaSUG 2025 Proceedings, Paper SI-180. Available at: https://pharmasug.org/proceedings/2025/SI/PharmaSUG-2025-SI-180.pdf

6. Rolo, D., Louw, B. 2023. "Can We Do It Better? Real-Time Validation of SDTM Mapping Is Superior to Double Programming." PharmaSUG 2023 Proceedings, Paper DS-193. Available at: https://pharmasug.org/proceedings/2023/DS/PharmaSUG-2023-DS-193.pdf

7. Jansen, L. 2025. "Running the CDISC Open Rules Engine (CORE) in BASE SAS." PharmaSUG 2025 Proceedings, Paper SD-044. Available at: https://pharmasug.org/proceedings/2025/SD/PharmaSUG-2025-SD-044.pdf

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Pietro Belligoli; Constantin Weberpals; Yarhy Flores

Technical University of Munich (TUM)

E-mail: pietro.belligoli@tum.de; constantin.weberpals@tum.de; yarhy.flores-lopez@tum.de

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.