Reimagining SDTM Mapping with Agentic AI and Automation

Nathan Johnson, eClinical Solutions, Mansfield, MA, USA

ABSTRACT

Transforming clinical trial data into SDTM standards remains one of the most time-intensive and manual steps in data preparation. This session introduces a next-gen end-to-end automation approach to streamlining this process with human-in-the-loop supported intelligence, including an agentic AI-driven method for determining appropriate transformation logic. In this session, explore the workflow and process for this modern method of mapping automation, including how the framework operates, and the components required of underlying clinical data technologies to facilitate this transformation approach. Join this session to learn how automating specification creation can maintain transparency and oversight while reducing manual effort, accelerating SDTM preparation and enabling data teams to focus on higher-value tasks.

INTRODUCTION

Mapping clinical trial data to the Study Data Tabulation Model (SDTM) is a core step in regulatory data preparation. That critical transformation activity requires careful interpretation of protocol intent, deep familiarity with heterogeneous source data, and consistent application of SDTM conventions. The industry has made real progress in automating parts of this workflow. Even so, SDTM mapping remains one of the most resource-intensive stages of the clinical data lifecycle.

Many recent automation efforts have focused on execution efficiency through metadata-driven transformations and specification-based code generation. These approaches can shorten programming time and improve consistency in how code is produced. In practice, however, questions about what a variable means, what the protocol is asking for, or which of several valid SDTM IG interpretations should be used still frequently stretch review cycle timelines.

This paper builds on work presented at PHUSE US 2025, which introduced a guided, human-in-the-loop user experience for automated SDTM mapping. That work emphasized transparency, traceability, and expert oversight as requirements for trustworthy automation. Over the past year, we implemented that approach and applied it in real mapping workflows. The experience gained through that implementation shaped the observations and evolution described here.

CONTEXT AND PRIOR WORK

In prior work, we examined two common limitations in SDTM automation. First, metadata-driven automated code generation still depends on fully-authored specifications, so much of the effort shifts upstream rather than disappearing. Second, end-to-end approaches that function as black boxes are difficult to validate and are rarely trusted in regulated environments. We proposed a guided user experience to bridge this gap: AI-driven suggestions surfaced to experts, adjudicated in context, and documented with an audit trail.

The approach centered on interactive specification development, confidence-based guidance, and traceable decision tracking. The goal was practical: minimize the effort spent on routine mappings while keeping experts closely involved where judgment is required. [1]

After PHUSE US 2025, this guided automation approach was implemented and used across multiple mapping efforts. The expected gains in execution were realized, but extended use exposed a remaining pain point: execution can be automated to a high degree, but the hardest part of SDTM mapping is often the reasoning that determines what the specification should say.

OBSERVATIONS FROM IMPLEMENTATION: WHERE AUTOMATION STILL FELL SHORT

Implementation of the guided approach produced clear benefits. Specification-driven code generation reduced manual programming effort and significantly accelerated initial dataset production. The guided review workflow also changed how teams spent their time: reviewers could focus on uncertain or high-impact items rather than combing through every line of code.

Over repeated use, however, the same friction kept resurfacing during downstream review and quality control. SDTM mappings still went through multiple rounds of revision. These changes were rarely about syntax or defects in generated code. They traced back to interpretation: what the protocol intended, what a source field represented in practice, or how to choose among multiple technically valid SDTM options or ambiguous guidance.

This meant that automation reduced the effort to produce SDTM datasets, but did not remove the effort needed to refine specifications. Review discussions continued to center on transformation intent, and they often resulted in specification updates after automated generation had already occurred. Observational metrics from this period were directionally consistent: programming effort decreased, and coding issue resolution decreased with it, but the number of review iterations did not change materially. The remaining churn reflected reinterpretation of transformation logic rather than correction of execution.

These findings helped clarify where existing automation tends to plateau. If the reasoning that drives mapping logic remains implicit, decisions are revisited later in the cycle, when change is disruptive and expensive.

REFRAMING SDTM MAPPING AS A REASONING PROBLEM

This implementation experience prompted a reassessment of the SDTM mapping problem space. Many automation efforts, including our own previous work, implicitly treat mapping as a workflow execution problem. Teams define and approve a mapping specification, then execute that specification in code reliably and efficiently. We can increase the reliability and efficiency of that workflow through automation. That framing has delivered measurable gains in speed and consistency, but it has not solved the problem.

During implementation, we learned that execution is rarely the primary source of friction. The most arduous and time-consuming work happens earlier, during specification authoring and review. Teams interpret protocol objectives, reconcile those objectives with heterogeneous source data, navigate ambiguous SDTM guidance, and choose among multiple technically valid representations. These steps require careful judgment and awareness of in- and cross-study context. They are often collaborative, as multiple subject matter experts are required. These are also the points at which disagreement, escalation, rework (and delays) most frequently occur.

Thus, the problem to be solved in STDM mapping is not primarily a coding problem. It is a reasoning problem supported by coding. When reasoning remains implicit, especially when it is distributed across emails, meetings, and individual expertise, automation cannot eliminate the churn that causes delay. When decisions during review only are reflected in coding changes, but not formalized in the specification, problems are more likely to resurface and repeat.

Reframing mapping as a reasoning problem shifts the objective of automation. The goal is no longer only to execute specifications faster. This has largely been demonstrated, and execution automation is now an exercise in optimization and scale. But this is all downstream of a more fundamental capability need: making reasoning explicit by supporting, simulating, automating, and documenting expert judgment.

AGENTIC, HUMAN-IN-THE-LOOP REASONING FOR SDTM MAPPING

To operationalize this reframing, we explored an agentic approach designed specifically to support the reasoning activities that precede the execution phase of mapping. In this paper, we use *agentic* to describe systems that evaluate alternatives, apply constraints, and produce justified recommendations rather than simply execute predefined instructions.

This distinction matters in the context of SDTM mapping because multiple outcomes may be technically correct. For example, a source variable may plausibly map to more than one SDTM endpoint depending on protocol intent, or the expression of an SDTM variable may be open to more than one interpretation that can be achieved through several acceptable transformation strategies. Human experts resolve these ambiguities by evaluating context, weighing previous experience, and measuring against regulatory expectations. An agentic system is intended to participate in that evaluative reasoning process.

This extends automation into specification creation in a way that more realistically maps to the experience of human experts. The system proposes candidate mappings and transformation logic based not just on rules or patterns from previous study specifications, but also on protocol content and SDTM IG recommendations. It surfaces alternative options, highlights ambiguity, articulates the underlying assumptions that led to each recommendation, and ranks the results. Human experts then review proposals and accept or modify recommendations, applying judgment and expert reasoning where required. The system thus serves as context-aware decision support, augmenting expert reasoning rather than replacing it.

Transparency remains a central design consideration. Each proposed transformation is accompanied by contextual signals and an explicit rationale. The user finalizes each recommendation by accepting it, modifying it, or rejecting it and providing their own entry. That result is stored alongside the specification for traceability and future model improvement. It is at this point in the process that collaboration occurs, and the specification eventually becomes a record not just of final mapping decisions, but a record of reasoning and justification.

By structuring and documenting reasoning at the point of specification creation, this approach aims to reduce late-stage reinterpretation. Ambiguity is resolved earlier in the process, when change is less disruptive and collaboration is more efficient.

END-TO-END WORKFLOW AND ARCHITECTURAL CONSIDERATIONS

Supporting reasoning-aware automation requires capabilities beyond workflow orchestration and deterministic code generation. It requires a clear separation between reasoning and execution layers, each with distinct responsibilities.

At a high level, the process begins with ingestion of protocol artifacts, source data, and standards metadata. These inputs provide the contextual foundation necessary to evaluate transformation intent. The reasoning layer then applies transformation patterns, constraints, and learned heuristics to generate candidate mappings. Rather than producing a single opaque result, the system surfaces alternatives, indicates relative confidence, and flags cases that warrant explicit human review.

A guided interaction model, leveraging the guided user experience principles we presented last year [1], allows teams to adjudicate these proposals. User decisions, modifications, and overrides are captured directly within the system, along with supporting rationale. This structured capture of intent differentiates reasoning-aware automation from traditional specification authoring tools.

Once specifications are finalized, execution proceeds through deterministic, reproducible code generation and transformation. Keeping reasoning and execution distinct in this process is intentional: reasoning is inherently context-sensitive and may involve probabilistic models, including large language models; execution, by contrast, remains deterministic, traceable, and auditable. Separating these layers allows decision support to evolve while preserving regulatory robustness in dataset production. Human experts remain ultimately accountable for mapping decisions, and the benefits seen in automated code generation can still be realized.

EARLY OUTCOMES, LIMITATIONS, AND FUTURE DIRECTIONS

Early use of a reasoning-oriented, agentic approach suggests benefits beyond execution-focused automation alone. By addressing ambiguity during specification creation and capturing intent explicitly, teams can reduce late-stage revisions and focus review on substantive questions rather than mechanical validation. This has the potential to achieve the promise of shortened SDTM preparation timelines and focus on higher-value oversight and analysis.

Important limitations remain. The observations presented here are based on limited real-world use rather than controlled experimentation or scaled adoption. Clinical data variability, evolving standards, and study-specific nuances will continue to challenge any automated system. Agentic reasoning does not eliminate the need for expert judgment. However, it does provide a structured way to apply that judgment consistently and document it clearly.

Future work will focus in several areas. First, expanding the coverage of transformation patterns to increase high-confidence recommendations from the agent. Second, improving the reasoning layer to take advantage of more powerful large language models. Third, expanding agentic reasoning to include not only standard transformations but also generation of novel functions including agent-authored code. Finally, continuing evaluation of workflow governance and validation considerations in regulated settings. Incremental adoption, conservative deployment, and continued emphasis on traceability will remain essential to building trust.

CONCLUSION

Experience with end-to-end SDTM automation shows that execution efficiency is achievable and necessary, but not sufficient. The most persistent sources of effort and variability sit upstream, where teams interpret intent and decide what the specification should contain.

By extending automation into this reasoning layer through an agentic, human-in-the-loop approach, teams can address ambiguity earlier, preserve transparency, and maintain regulatory confidence. The ultimate goals remain: accelerate

SDTM preparation, reduce manual effort while maintaining quality, and enable teams to spend more time on higher-value work. Reimagining SDTM mapping in this way shifts the focus of automation from reproducing human task execution to simulating and supporting subject matter expert reasoning.

REFERENCES

[1] Johnson, N., Mallarapu, V. (2025). A Guided User Experience for Automated Mapping. PHUSE US Connect 2025.

ACKNOWLEDGMENTS

We acknowledge Steve Linberg for his contributions to system development, evaluation, and manuscript review, and we thank all team members who participated in the development and implementation of this work.

The authors acknowledge the use of ChatGPT (GPT 5.2), a large language model developed by OpenAI, to assist with drafting, structuring, and refining the narrative of this paper. All ideas, interpretations, and conclusions presented remain the responsibility of the authors, who reviewed and validated the content for accuracy, completeness, and alignment with regulatory and professional standards.

CONTACT INFORMATION

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

Author Name: Nathan Johnson
Company: eClinical Solutions LLC
Address: 603 West St., Mansfield MA, USA 02048
Work Phone: +1 216 245 5166
Email: njohnson@eclinicalsol.com
Website: www.eclinicalsol.com