PharmaSUG 2026 - Paper DV-189

Metadata Network, Impact Analysis, and Visualization Across Rawdata, SDTM, ADaM, and TFL

Nana Yang, Tingting Zeng, Yaohui Zhu, Yanan Sui and Mingliang Fan, BeOne Medicines

ABSTRACT

End-to-end traceability across rawdata, SDTM, ADaM, and TFL is an important foundation for lineage explanation, impact assessment, and review preparation in clinical data development. In practice, however, relevant relationships are often distributed across specifications, mapping logic, derivation logic, reviewing outputs, and reporting artifacts, making cross-layer dependencies difficult to understand as a whole and downstream impact scope difficult to assess in time. To address this problem, this paper presents a trace data framework that organizes distributed cross-layer relationships into a queryable metadata network. The framework generates trace segments from three linkage domains, RAW-to-SDTM, SDTM-to-ADaM, and ADaM-to-TFL, and dynamically assembles them into continuous lineage chains according to user-selected scope and filters. Using this framework, users can identify which CRF fields support a given TFL output, analyze the upstream origins of an ADaM derivation, assess downstream impact caused by SDTM changes, and inspect the relationships between CRF forms and reporting TFL targets. To improve practical usability, the framework further provides Sankey-style flow diagrams, trace tables, form-to-output mappings, and downloadable trace datasets for review, communication, and documentation support. Practical implementation demonstrates that once trace relationships are organized into navigable, composable, and reusable trace data, they transcend their role as a governance requirement. Instead, they become a computational layer that actively supports explainable, auditable, and maintainable clinical data development.

INTRODUCTION

In clinical trial data development, the accuracy, interpretability, and consistency of ADaM and downstream TFL outputs directly affect statistical quality, review efficiency, and credibility. Because analysis and reporting development usually spans rawdata/CRF, SDTM, ADaM, and TFL, while also involving mapping, derivation, program dependency, and output organization logic, end-to-end traceability is best understood as infrastructure for issue diagnosis, impact assessment, and review preparation. In practice, however, this traceability is often distributed across CRF design, mapping specifications, derivation descriptions, program logic, and review artifacts, making cross-layer dependencies difficult to express continuously. As a result, when teams need to answer questions such as which CRF fields support a given TFL output, whether an ADaM variable has complete upstream provenance, or which downstream objects are affected by SDTM variables change, they often must manually cross-reference documents, code, and institutional knowledge. Under conditions of evolving analytical requirements, frequent specification revisions, and accelerating review cycles, static documentation and isolated mappings are no longer sufficient for change impact analysis, exploratory diagnosis, or review-ready evidence.

In response to these challenges, this paper presents a trace data framework for the clinical data pipeline. Rather than leaving traceability at the level of descriptive documentation, the framework operationalizes it as structured, queryable trace data and further organizes it into an integrated metadata association network. Within this framework, cross-layer relationships spanning rawdata, SDTM, ADaM, and TFL are represented as composable trace segments and dynamically assembled into continuous lineage chains according to selected scope and filters. The following sections discuss the conceptual model, framework design, interaction and visualization scenarios, implementation experience, and limitations, showing how this approach provides a more stable foundation for explainable, auditable, and maintainable clinical data development.

CONCEPTUAL MODEL: FROM STATIC TRANCEABILITY TO A QUERYABLE METADATA NETWORK

In traditional usage, traceability is more often treated as a principle-level requirement: development processes should be able to trace sources, justify derivations, and support review. However, if these relationships remain only in static tables, scattered mappings, or standalone explanatory documents, they are difficult to extend into analysis-driven use scenarios. The focus of this paper is therefore not simply whether traceability statements exist, but whether these relationships can be organized into a data layer that is computable, composable, and reusable. The significance of a trace data framework lies in enabling this transformation: it represents distributed dependencies as structured trace data that can be programmatically read, filtered, and reorganized while supporting both upstream backtracking and downstream propagation analysis. On this basis, questions that previously depended on manual reconstruction can be addressed more consistently and efficiently, such as tracing from a TFL output back to the ADaM, SDTM, and rawdata fields on which it depends, or starting from a rawdata field or derivation change to assess downstream propagation scope.

To support this kind of cross-layer analysis, this paper adopts the metadata association network as its central concept. The network does not treat RAW-to-SDTM, SDTM-to-ADaM, and ADaM-to-TFL as isolated fragments, but instead organizes them as continuous dependency chains that can be stitched together under standardized rules. The same underlying chain data can produce row-level trace records for precise retrieval and export, while also being derived into source-target relationship pairs for Sankey-style visualization and hierarchical exploration. For this reason, traceability is transformed from static explanation into an interactive and reusable metadata network.

FRAMEWORK DESIGN

THREE FOUNDATIONAL TRACE SEGMENTS

The framework takes three trace segments as the starting point of a unified trace foundation and uses them to decompose cross-layer dependencies into semantically clear building blocks that can later be assembled into continuous lineage. These three foundational segments differ in functional role, metadata origin, and key output fields, yet together form the core structure of the unified trace foundation. Specifically, the **RAW-to-SDTM segment** primarily preserves source context by expressing relationships among CRF forms, rawdata dataset names and field variables and SDTM variables, and therefore corresponds to the collection and standardization mapping layer. The **SDTM-to-ADaM segment** focuses on derivation and analytical transformation by organizing intermediate variable transformation relationships from SDTM variables and domains to ADaM variables and domains. The **ADaM-to-TFL segment** then connects variables in analysis datasets with report blocks, outputs, and sections, corresponding to the layer of output organization, program use, and reporting presentation. This design is not a simple concatenation of three segments. Instead, it represents a logical integration that respects different data semantic domains and maintenance origins. Stable dependency fragments are first extracted separately, then connected across layers through unified standardization. This approach preserves both the heterogeneity of multi-source metadata and the continuity required for cross-layer analysis. In the current implementation, these segments are stored and transferred through unified fields. This standardized naming allows the system to avoid maintaining a separate dependency logic for each view and instead construct different levels of output forms around the same basic data structure of trace foundation.

END-TO-END ASSEMBLY LOGIC

Within the assembly layer, the three trace segments are further standardized and merged into continuous lineage chains. The fundamental objective is to place dependency information from different metadata domains into single, ordered upstream-to-downstream expressions. A complete chain, for example, may begin with a CRF form and rawdata field, continue through an SDTM domain.variable and an ADaM dataset.variable, and extend to a TFL block and final output. This chain-based representation preserves a dependency order that is understandable to humans while remaining suitable for programmatic processing. Cross-layer relationships in clinical projects, however, are not always neat one-to-one mappings. Some ADaM variables depend on combinations of multiple SDTM variables; some TFL blocks directly use SDTM without a full ADaM transition; and some study-specific logic introduces unusual variable combinations, parameter substitutions, or partially aligned relationships. Assembly logic must

therefore do more than simple concatenation. It must handle cross-layer jumps, combined-variable matching, missing sources, and missing targets. In the current implementation, such cases are still incorporated into a unified chain-building process so that the final output can explicitly mark states such as “source exists without target” or “target exists without complete source,” thereby retaining analytical value.

LAYERED ORGANIZATION AND A UNIFIED TRACE ENGINE

From a system perspective, the framework can be abstracted into three layers: presentation layer, service layer, and trace assembly layer. **Presentation layer** receives user-selected filters and returns statistics, graphics, tables, and exportable results. **Service layer** normalizes study scope, filtering logic, caching strategy, and response shaping. **Trace assembly layer** performs segment extraction, standardization, chain assembly, and relationship derivation. This layered structure enables the same trace foundation to support overview, analysis, and change impact assessment scenarios and so on. The value of a unified trace engine lies not in merely collecting multiple functions together, but in preventing different interfaces and analysis scenarios from maintaining similar but inconsistent dependency interpretation logic. Once overview, deep-dive analysis, impact analysis, and exported review all rely on the same underlying trace chains, the relationship expressions seen by teams remain consistent across tasks, which is especially important for review and communication.

Figure 1 illustrates the overall organization of trace data framework, including relationships among the presentation layer, service layer, and trace assembly layer, and shows how the three foundational linkage segments, RAW-to-SDTM, SDTM-to-ADaM, and ADaM-to-TFL, jointly support overview, analysis, and change impact assessment scenarios.

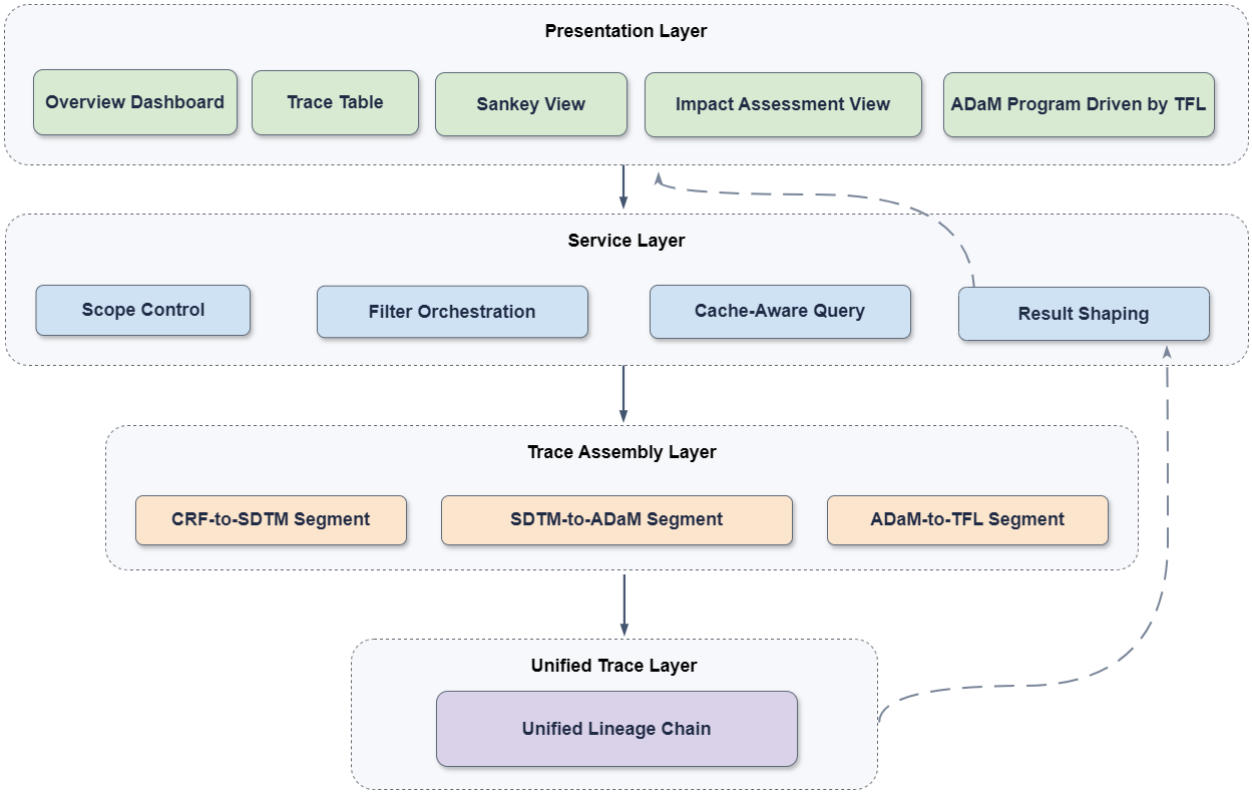

Figure 1. Overall Architecture of the Trace Data Framework

QUERY OPTIMIZATION AND CACHE-AWARE SERVICES

As trace data span multiple levels and continue to grow in volume, performance becomes a critical factor in practical interactive use. Existing module design indicates that the service side adopts a two-part strategy combining persistent cache and filter-oriented query organization. On one side, persistent cache

reduces repeated assembly cost. On the other, structured filters control analytical scope so that queries first contract at the most strongly constrained level and then propagate conditions across the remaining lineage segments. Once trace relationships are organized as a network, the system naturally faces a larger combinational space. Without control over query order, scope narrowing, and result shaping, users can easily receive results that are too broad, too noisy, too time-consuming, and difficult to interpret. By contrast, structured filtering allows analysis to proceed in an ordered way around dimensions such as CRF form, field, SDTM, ADaM, and TFL output, making the trace data explorable and interactive without becoming unwieldy.

USER INTERACTION, VISUALIZATION

INTERACTION VIEWS FOR DIFFERENT TASKS

From a task organization perspective, interaction views can be grouped into three types. **Overview views** emphasize study-level CRF usage, form/field coverage, and overall connectivity density. They primarily answer questions such as which forms, datasets, and outputs are most densely connected within a study, typically using filters such as study, domain, form, and output group, and they mainly produce usage summaries that help users quickly understand trace distribution across the study. **Analysis views** support deeper exploration around a specified output, form, variable, or dependency level. They focus on questions such as what the exact lineage path is for a given output or variable, often using TFL program name and SDTM or ADaM variable name as key filters, and they produce detailed trace records, lineage detail, and path-browsing outputs. **Impact-assessment views** start from upstream SDTM variables changes and examine downstream propagation. They focus on which downstream objects are affected by a given upstream change and produce outputs such as impacted ADaM variables and TFL targets, thereby supporting change assessment and review prioritization. Figure 2 shows an example of CRF usage and form/field coverage.

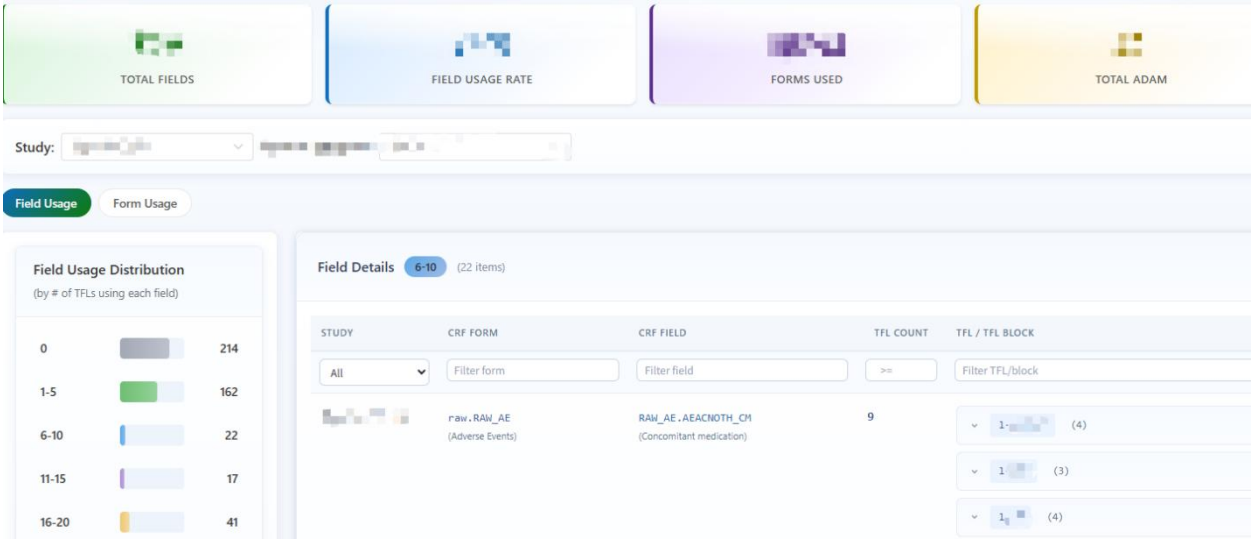

Figure 2. Example of Overview Analysis of CRF usage and form/field coverage

VISUALIZATION AS AN ANALYTICAL INTERFACE

In terms of dependency representation, Sankey-style flow diagrams are particularly well suited for expressing the overall structure of cross-layer connectivity. By displaying dependency traversal as hierarchical flows, these diagrams help users quickly identify which upstream objects feed a particular output or which downstream targets may be affected by an upstream change. By contrast, row-level trace tables are better suited to precise localization: users can directly inspect form, field, dataset, variable, block, and output information for each chain and use these records for review, communication, and evidence retention. The significance of visualization lies not merely in presenting results, but in providing a unified interface for analytical exploration, anomaly diagnosis, version change review, and human

review activity. Under a human-in-the-loop model, visualization should therefore be understood as a work surface that is reviewable, discussable, and revisitable. Figure 3 presents an example of this unified interface through a Sankey-style network view from rawdata to T-DM.

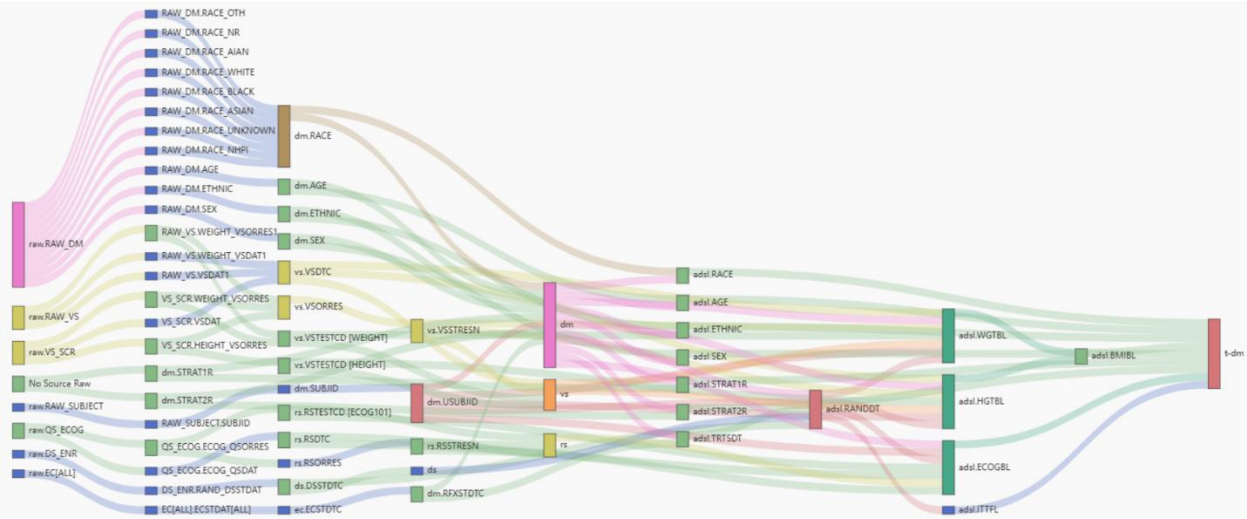

Figure 3. Example of Sankey-Style View of the Metadata Association Network from Rawdata to T-DM

CHANGE IMPACT ANALYSIS AS A CORE INTERACTION SCENARIO

Among these interaction scenarios, change impact analysis directly demonstrates the operational value of trace data. Under traditional approaches, when upstream specifications or CRF mappings change, teams often rely on manual judgment to determine whether downstream ADaM derivations, TFL outputs, or review artifacts are affected. This process is not only time-consuming, but also vulnerable to omissions because of differences in experience and breaks in information continuity. Within this framework, the metadata network forms the infrastructure supporting detection, assessment, notification, and decision-making. From this perspective, change impact analysis proceeds through four interconnected steps. **Detection** identifies changes in upstream objects, such as additions to or deletions from SDTM variables, and its practical value lies in rapidly identifying triggers that require evaluation. **Assessment** evaluates which ADaM variables, TFL blocks, and outputs may be affected along this dependency network; its core action is to use lineage chains to calculate propagation paths and impact scope, thereby reducing manual investigation time. **Notification** presents affected areas in a structured way to reviewers, creating a shared starting point for collaboration. **Decision Support** then uses propagation scope, impact depth, and object significance to determine priorities for subsequent updates, validation, and focused review, thereby improving resource allocation and review efficiency.

This workflow depends on traceability already having been expressed as continuous and queryable data chains. Only when upstream changes can be stably mapped to a downstream dependency network can impact analysis move beyond tacit judgment and manual investigation into a structured, repeatable analytical process. In this sense, trace data are not a by-product of change impact analysis, but they are its computational basis. Another important value of change impact analysis lies in improving cross-role communication. Programmers, reviewers, and study teams often focus on different aspects of the same change: whether derivations need adjustment, whether outputs are affected, and whether review priority should shift. Without a unified trace view, these judgments are difficult to anchor on the same factual basis. The metadata association network provides a relatively neutral and shared dependency representation layer that allows different roles to form a common understanding around the same result set. The endpoint of interaction and visualization is therefore not merely clearer display, but the conversion of change impact analysis into executable decision support. Figure 4 illustrates this with a concrete example in which variables are removed from an upstream SDTM domain. The system traverses the dependency graph to identify all affected variables in the corresponding ADaM dataset (e.g., ADAE), including both direct dependencies and indirect impacts through derivation chains. Beyond

the SDTM-to-ADaM linkage, the full traceability chain extends upstream to CRF collection and downstream to TFL outputs and remains maintained and queryable within the platform.

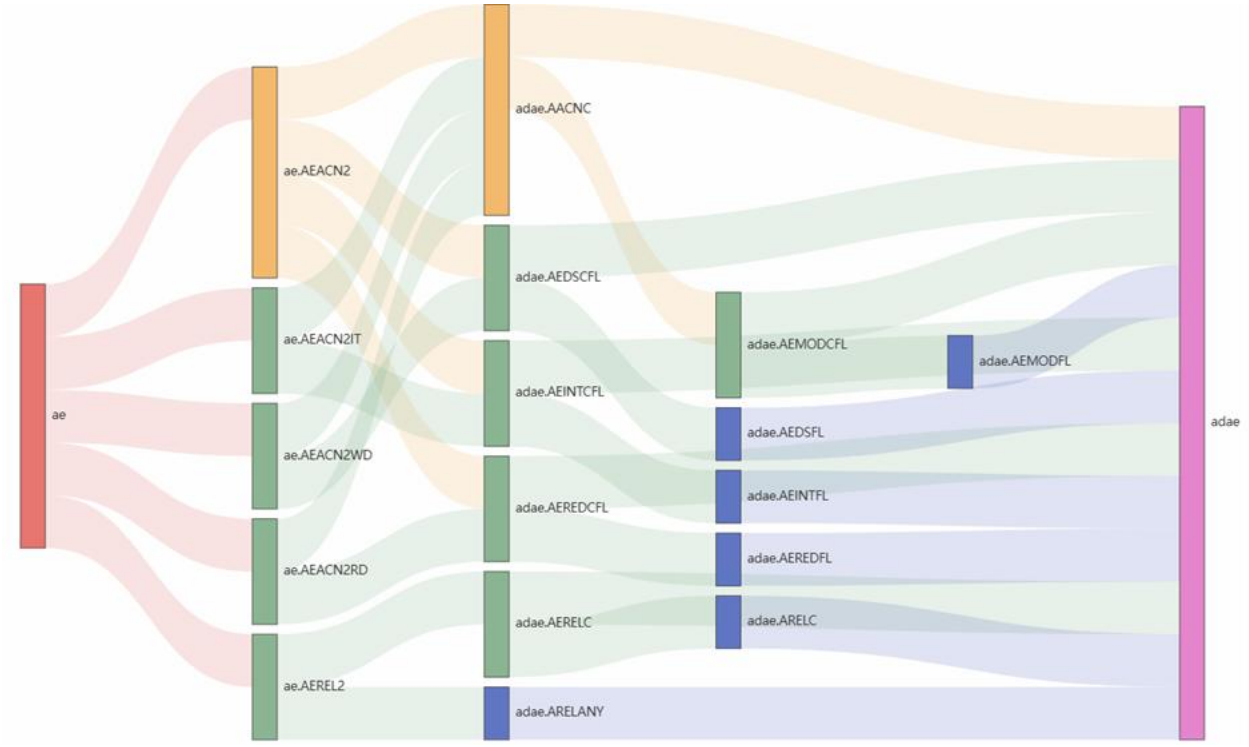

Figure Note: The above figure contains the SDTM variables for action taken (e.g. AEACN2IT/AEACN2WD/AEACN2RD for Drug Interrupted, Withdrawn or Dose Reduced) or causality (e.g. AEREL2) to specific drug, the directly impacted ADaM variables (e.g. AEINTCFL/ AEDSCFL/AEREDCFL/AERELC) and indirect impacted ones (e.g. adae.AEMODCFL/ARELC/AEMODFL) through derivation chains.

Figure 4. Example of SDTM Change Impact Propagation to ADaM Variables

IMPLEMENTATION EXPERIENCE

Implementation shows that the effectiveness of a trace data framework does not arise solely from connected lineage paths, but from the combined action of several design principles. First, a unified trace foundation allows overview statistics, deep analysis, Sankey flow, and change impact assessment to be built on the same underlying dependency representation, thereby reducing logical divergence and inconsistent results across scenarios. Second, output design must support both computational usefulness and review usefulness. Detailed trace records support precise retrieval and exported evidence, while relationship-pair representations support visualization and path browsing, further improving the framework's suitability across different analytical contexts.

In addition, cache awareness and scope control determine whether the framework can truly support interactive analysis. Without filter-oriented scope narrowing, result shaping, and caching mechanisms, cross-layer trace can quickly lose practical value because of excessive combinational complexity. By contrast, when query scope, response speed, and result clarity remain balanced, the trace framework can move beyond being a one-time analytical tool and become infrastructure for routine review workflows. Finally, the value of human-machine collaboration does not lie in passive confirmation of the terminal result, but in transparency, interpretability, and revisitable process history. Trace data provide not only answers, but dependency evidence that can be reviewed, compared, and discussed, which is precisely what makes high-quality human-in-the-loop operation possible.

LIMITATIONS AND FUTURE DIRECTIONS

Although the trace data framework shows clear strengths in structured traceability and impact analysis, its applicability remains constrained by several practical conditions. The most direct limitation comes from metadata quality and coverage. If upstream specifications are incomplete, mapping records are partial, or program-level dependencies have not been sufficiently extracted, end-to-end trace completeness will necessarily be affected. Another limitation comes from the complexity of cross-layer relationship inference. For complex parameter combinations, special intermediate dataset processing, or non-standard output logic, the system still requires study-specific heuristics and domain knowledge, which means the framework cannot yet operate entirely independently of business context. At the same time, when analytical scope becomes too broad and filtering remains insufficient, both visual and tabular outputs may suffer from information overload and reduced interpretability.

These limitations also define the main directions for future work. One direction is stronger version-aware trace comparison so that the framework can not only represent current dependencies, but also systematically compare structural changes across snapshots. Another is the introduction of more automated alert and recommendation layers so that detection, notification, and preliminary prioritization can move further upstream. Third is expansion of the framework into broader cross-study standardization and AI-assisted review scenarios in order to validate its transferability across therapeutic areas, data patterns, and governance requirements.

CONCLUSION

The central argument of this paper is that once traceability in clinical data development is transformed into structured, queryable trace data, it serves as a practical computational layer supporting lineage explanation, impact assessment, review preparation, and cross-role communication. By connecting Rawdata/CRF, SDTM, ADaM, and TFL through a unified metadata network, the trace data framework enables cross-layer dependencies to be represented in standardized form, dynamically assembled, and presented through multiple output modes. This framework can support summary dashboards, analytical exploration, Sankey visualization, trace dataset export, and change impact assessment, while also providing a stronger evidentiary basis for transparent human-in-the-loop monitoring and higher-quality decision-making. Trace data should therefore be understood not as passive records of an existing workflow, but as active infrastructure for explainable, auditable, and maintainable clinical data development.

RECOMMENDED READING

Luu M., et al. Development of a Web-Based Interactive Tool for Visualizing Breast Cancer Clinical Trial Tolerability Data. JCO Clin Cancer Inform. 2024 Jul;8:e2400007. doi: 10.1200/CCI.24.00007. PMID: 39013121; PMCID: PMC11254331.

Tingting Zeng., et al. (2026), "AI-Driven Intelligent Platforms for ADaM Specification and Code: Empowering Clinical Data Analysis" . Available at https://pharmasug.org/proceedings/2026/AI/PharmaSUG-2026-AI-278.pdf

CONTACT INFORMATION

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

Nana Yang

BeOne Medicines
nana.yang@beigene.com