Paper AD08

TFL Automation Through Design, Specify, Execute, and Report: A Seamless End-to-End Framework

Malan Bosman, Clymb Clinical, Bloemfontein, South Africa
Bhavin Busa, Clymb Clinical, Burlington, MA, United States

ABSTRACT

This paper presents a comprehensive framework for automating the full lifecycle of Tables, Figures, and Listings (TFLs) in clinical trials - spanning design, specification, execution, and reporting. The framework integrates multiple complementary tools to streamline each phase of the process.

It begins with TFL Designer, a no-code tool that enables users to design standardized TFL shells and generate structured metadata. This metadata incorporates CDISC standards, including the Analysis Results Standards (ARS), to support traceable automation. Next, siera, an R package purpose-built for metadata-driven programming, consumes this metadata along with ADaM datasets to generate Analysis Results Datasets (ARD) and produce validated outputs in R. Finally, TFL Viewer provides a centralized platform for reviewing outputs, capturing reviewer comments, tracking versions, and supporting collaborative reporting workflows.

Together, these components demonstrate how a metadata-first, standards-driven framework can automate the TFL lifecycle end to end.

INTRODUCTION

Tables, Figures, and Listings (TFLs) are core deliverables in clinical trials, supporting interim analyses, regulatory review, and Clinical Study Report (CSR) development. Despite advances in statistical software and data standards, the TFL lifecycle remains fragmented. Shells are often developed manually, programming logic is duplicated across studies, and review workflows rely on static documents with manually consolidated feedback.

Industry initiatives such as CDISC 360 and the recently launched CDISC 360i program provide the foundation for a metadata-driven, standards-based approach to clinical study design, analysis, and reporting [1, 2]. These initiatives emphasize the prospective use of standards to enable automation across the study lifecycle. Building on this vision, the publication of the CDISC Analysis Results Standard (ARS) introduces a standardized logical model to prospectively define analyses, describe outputs, and store analysis results in a structured, machine-readable form [3, 4]. Early industry presentations and papers have demonstrated how ARS supports automation, traceability, and reuse of analysis results metadata [5, 6, 7].

This forward-looking standards strategy also anticipates emerging CDISC initiatives such as Analysis Concepts, enabling future integration of standardized analysis intent into SAP and downstream automation workflows.

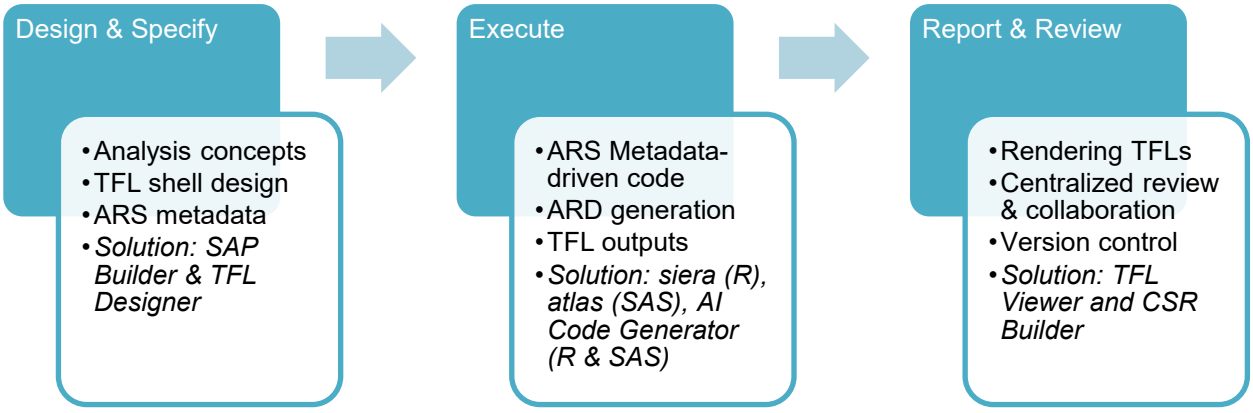
This paper and accompanying presentation are submitted as part of the Application Development and Software Demo stream at the first PHUSE APAC conference. The intent is not only to describe the framework conceptually, but to demonstrate the full, end-to-end process from TFL design and specification, through execution, and into reporting and collaborative review. Attendees will see how metadata captured during design directly drives downstream automation and how multiple execution engines and review workflows integrate seamlessly within a single framework.

FRAMEWORK OVERVIEW

The proposed framework organizes the TFL lifecycle into three connected phases:

1. **Design and Specify**: Define TFL shells, analysis intent, and structured ARS metadata

2. **Execute:** ARS metadata-driven code, generate analysis results datasets (ARD) and TFL outputs using metadata-driven automation
3. **Report and Review:** Render ARD for display / TFL outputs in RTF/PDF formats, review, collaborate, and manage versions centrally



CDISC ARS serves as the single source of truth, linking specification, programming logic, and reported results across all phases. While the overall framework also supports upstream automation of ADaM specification and programming, these components are out of scope for this paper and demonstration, which focus specifically on the design, execution, reporting, and review of TFL outputs.

Figure 1 below illustrates the end-to-end workflow supporting the proposed TFL automation framework. The schematic shows how ARS-aligned metadata flows from design and specification through execution, reporting, and review, and how the individual solutions: SAP Builder, TFL Designer, TFL Code Generators (siera, atlas, and AI Code Generator), and TFL Viewer work together to operationalize the framework. While the primary focus of this paper and demonstration is TFL automation, the workflow also highlights how the same metadata-driven architecture extends downstream to support CSR generation and statistical report writing.

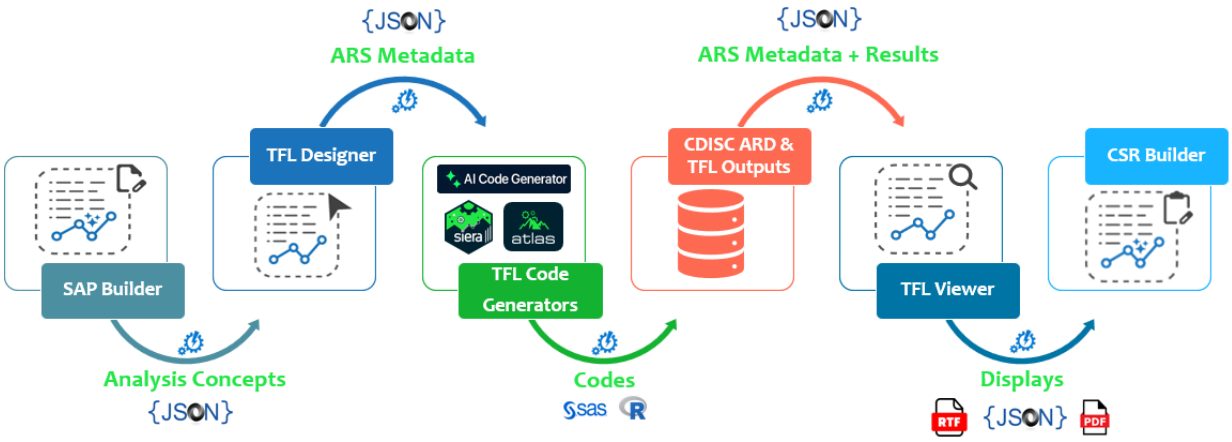


Figure 1: End-to-end solution architecture supporting metadata-driven TFL automation and reporting.

1. DESIGN AND SPECIFY: METADATA-DRIVEN TFL AND SAP DEFINITION

The framework begins with TFL Designer, a web-based, no-code application offered in Community and Enterprise editions. The TFL Designer Community Edition provides open access to standardized TFL templates and ARS-aligned metadata capabilities, enabling early adoption of metadata-driven design [8].

The Enterprise Edition extends these capabilities with study-level configuration, governance, collaboration, integrations, and AI-enabled features.

TFL Shell Design and Standardization

TFL Designer enables users to design mock shells using configurable templates aligned with industry standards, including the U.S. FDA Safety Tables, Figures, and Listings Integrated Guide [9], JPMA ADaM-based analysis and reporting guidance [10], and CDISC eTFL Portal templates [11]. The free TFL Designer Community version 2.0 provides access to a curated library of pre-built templates (n=109 TFL templates), while the Enterprise edition extends this library to more than 500 templates along with additional metadata and governance capabilities. Templates and metadata are organized by therapeutic area, reporting events (CSR, DSUR, IB), and study-specific requirements. Shells and associated metadata are created, edited, and versioned within the platform, eliminating reliance on disconnected Word or Excel files.

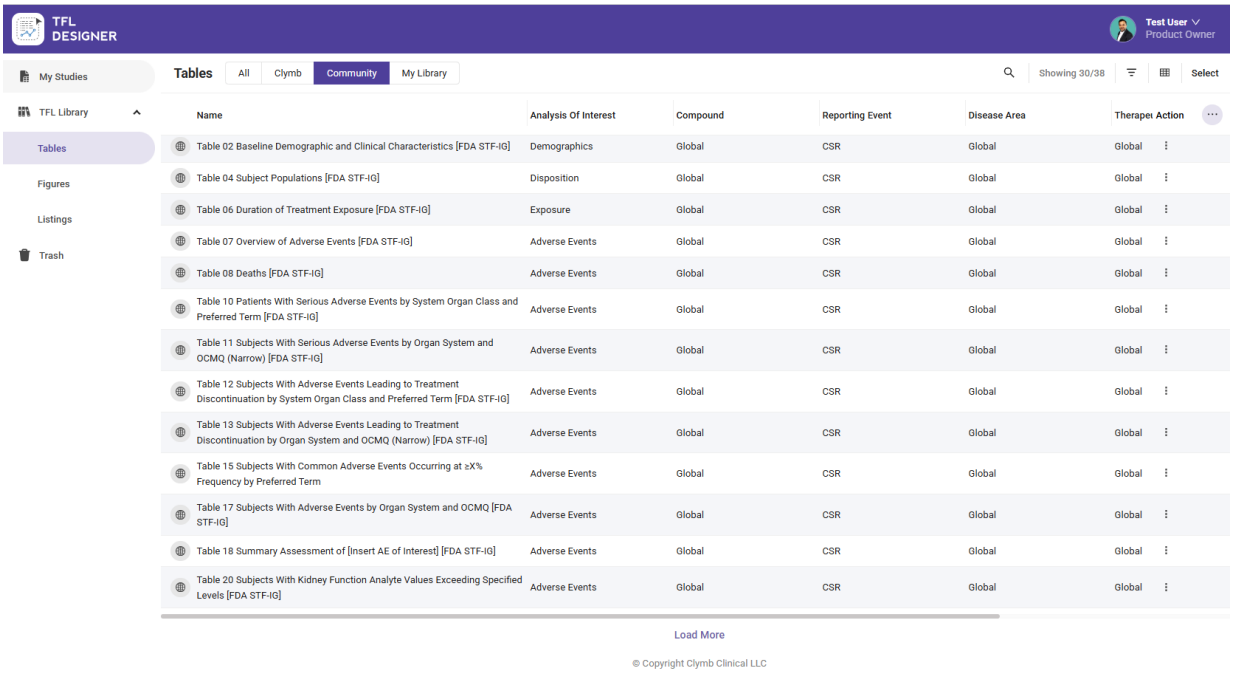


Figure 2: TFL Designer Community version - library of pre-built TFL templates.

ARS-Aligned Metadata Capture

Beyond layout and formatting, TFL Designer captures analysis-level metadata aligned with the CDISC ARS logical model, including:

- Analysis sets and subject populations
- Data subsets and selection criteria
- Groupings and stratification factors
- Analysis methods and statistical operations

Prospective capture of this metadata ensures that analysis intent, programming logic, and outputs remain linked throughout the lifecycle [3, 4, 5, 6].

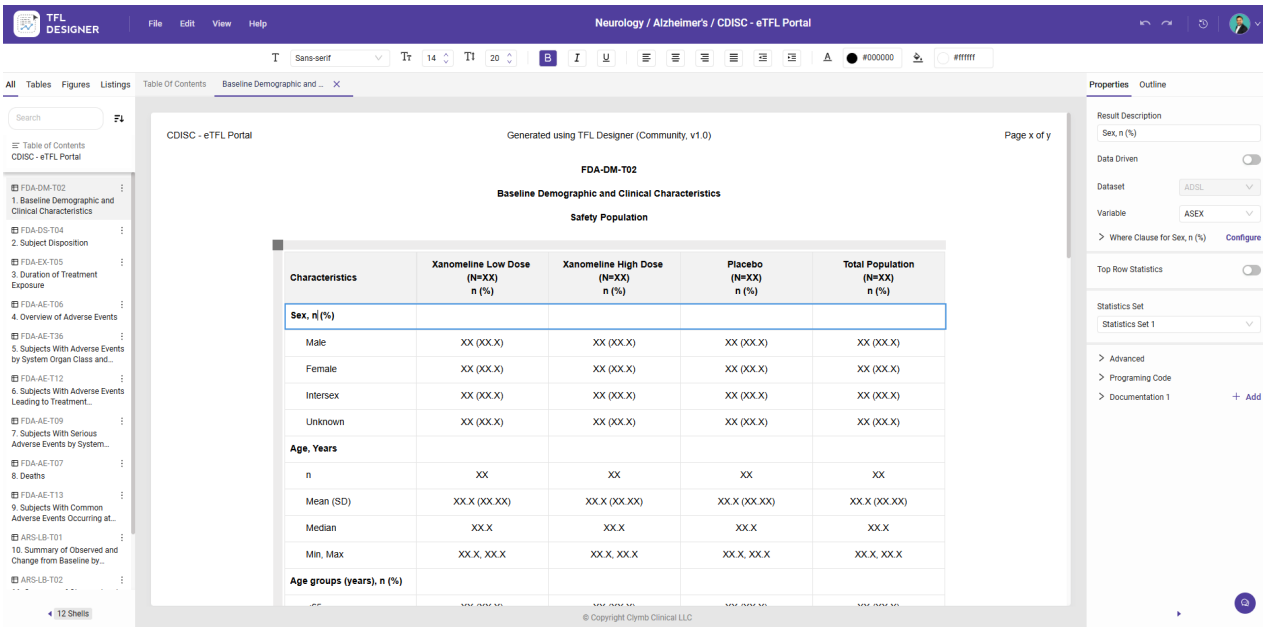


Figure 3: Metadata-driven TFL shell design in TFL Designer Community version. This figure shows how TFL shells are created using standardized templates and enriched with ARS-compliant metadata describing analysis sets, groupings, where clause, analysis methods, programming code context, and output structure.

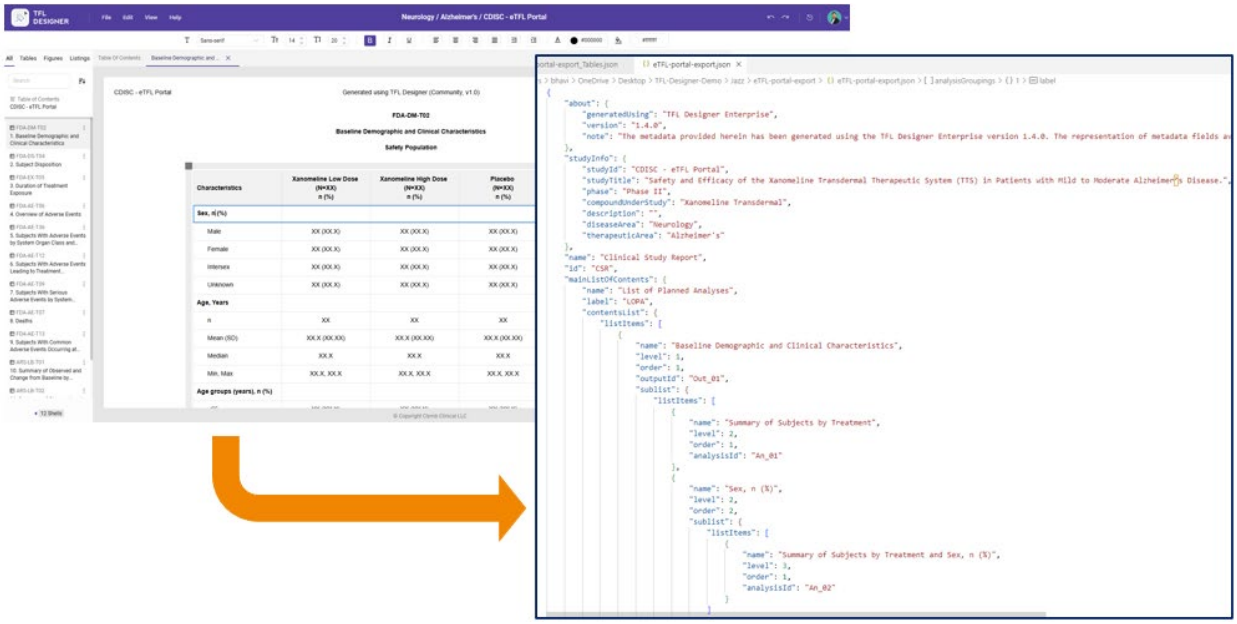


Figure 4: Extraction of CDISC ARS-aligned metadata from TFL Designer in JSON format. Metadata defined during the Design and Specify phase is ingested in the execution and reporting phases, reducing duplication and improving traceability.

Expansion to SAP Specification with SAP Builder

As an expansion of the Design and Specify phase, **SAP Builder** is being rolled out as an AI-enabled capability that builds on the same structured metadata foundation. SAP Builder reuses analysis definitions, populations, endpoints, and methods captured during TFL design to assist with the development and maintenance of Statistical Analysis Plan (SAP) content, improving alignment between the SAP and downstream deliverables while reducing duplication and manual effort.

2. EXECUTE: METADATA-DRIVEN TFL CODE GENERATION AND ANALYSIS

The Execute phase supports multiple, complementary automation paths, allowing organizations to leverage existing programming ecosystems while adopting metadata-driven workflows.

R-Based Execution with siera

At the core of execution is **siera**, an open-source R package (on CRAN, approved by Pharmaverse and CDISC Open-Source Alliance organizations) designed for metadata-driven programming. siera ingests ARS metadata together with ADaM datasets to generate meta-programmed R scripts which, when executed, produce Analysis Results Datasets (ARDs) [7]. The generated code is transparent and reviewable, supporting validation and controlled adaptation.

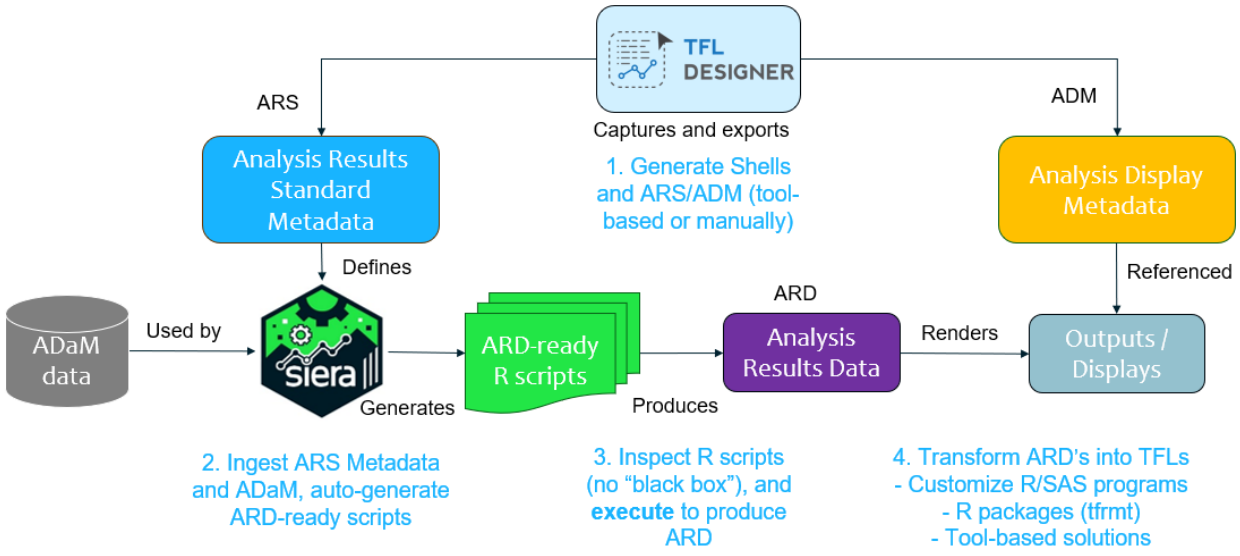


Figure 5: siera meta-programming workflow from ARS metadata to ARD generation. This figure illustrates how ARS metadata is translated into executable R code and combined with ADaM datasets to produce structured Analysis Results Datasets.

SAS-Based Execution with atlas

To support SAS-centric environments, the framework also includes **atlas**, a SAS-based automation engine that consumes ARS-aligned metadata to dynamically select macros from a SAS macro library and generate executable SAS programs for individual TFLs. This approach enables organizations to leverage existing, validated SAS infrastructure while benefiting from standardized metadata and reduced manual programming [8].

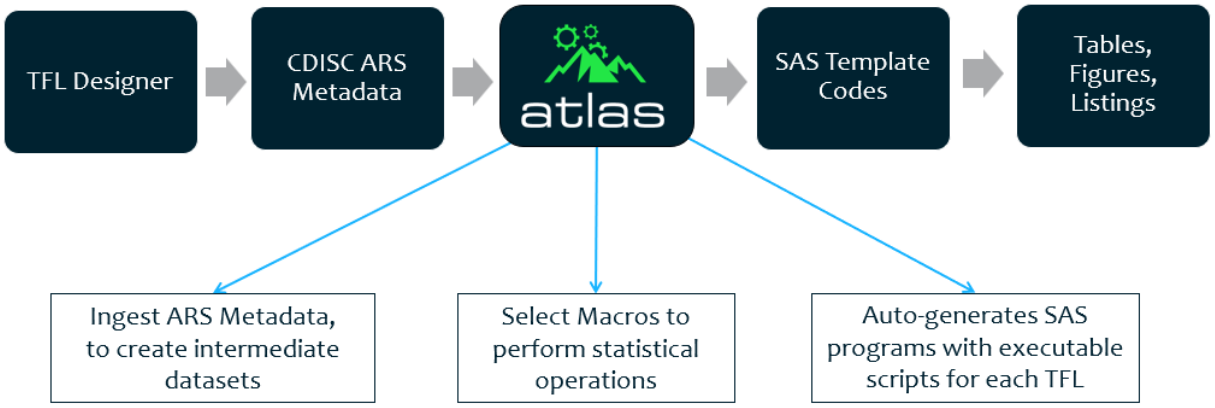


Figure 6: SAS-based TFL execution using atlas. ARS metadata is used to parameterize SAS macro calls, enabling automated generation of validated, executable TFL programs within existing SAS infrastructures.

AI-Assisted Execution with AI Code Generator

The Execute phase is further extended through the **AI Code Generator**, available as part of TFL Designer Enterprise. This LLM-driven capability uses structured ARS metadata, study context, and standardized analysis definitions to assist with statistical programming code generation, complementing both siera and atlas while maintaining transparency and reviewability [12].

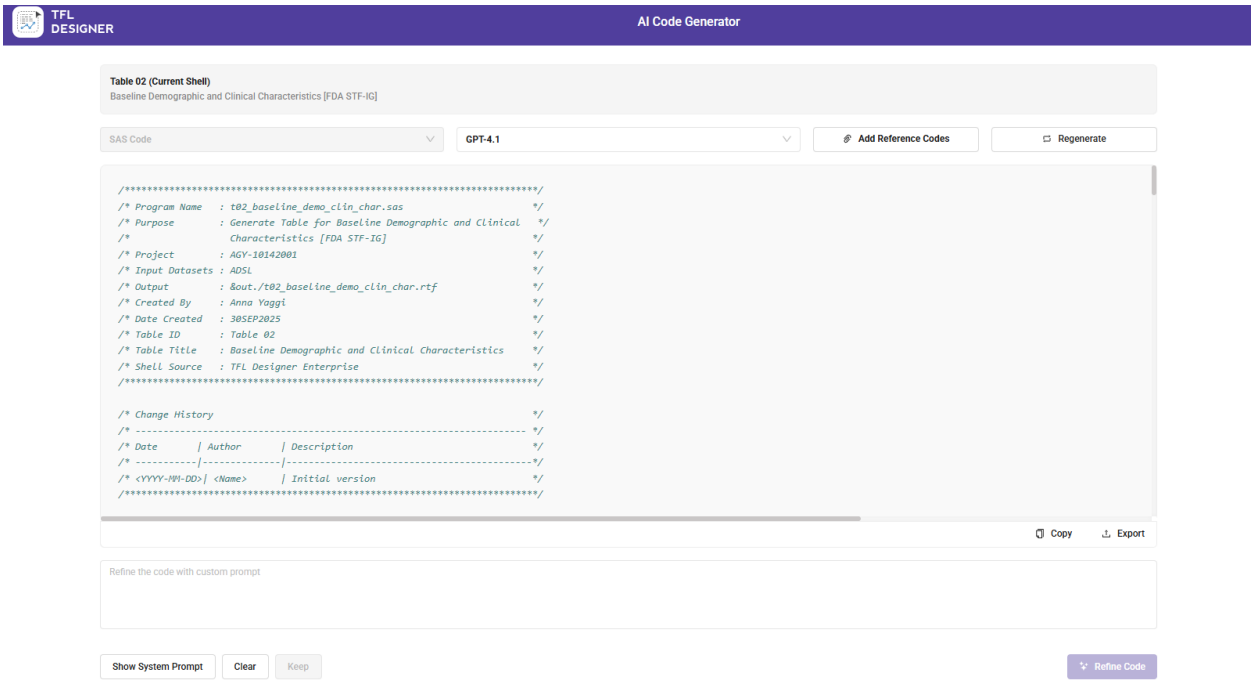


Figure 7: AI-assisted code generation using ARS and analysis display metadata. This figure shows how large language models use structured metadata and study context to assist with statistical programming while keeping code (both R and SAS) inspectable and aligned with company standards.

3. REPORT AND REVIEW: CENTRALIZED COLLABORATION WITH TFL VIEWER

Following execution, outputs move into the Report and Review phase using **TFL Viewer**, a centralized, web-based platform designed to replace fragmented document-based review workflows [13, 14].

TFL Viewer ingests ARD in JSON or TFLs in RTF and PDF formats and organizes them into a structured study workspace. Reviewers add comments directly within the platform, track resolutions, and compare versions. A complete version history is maintained, providing an audit trail suitable for regulatory inspection.

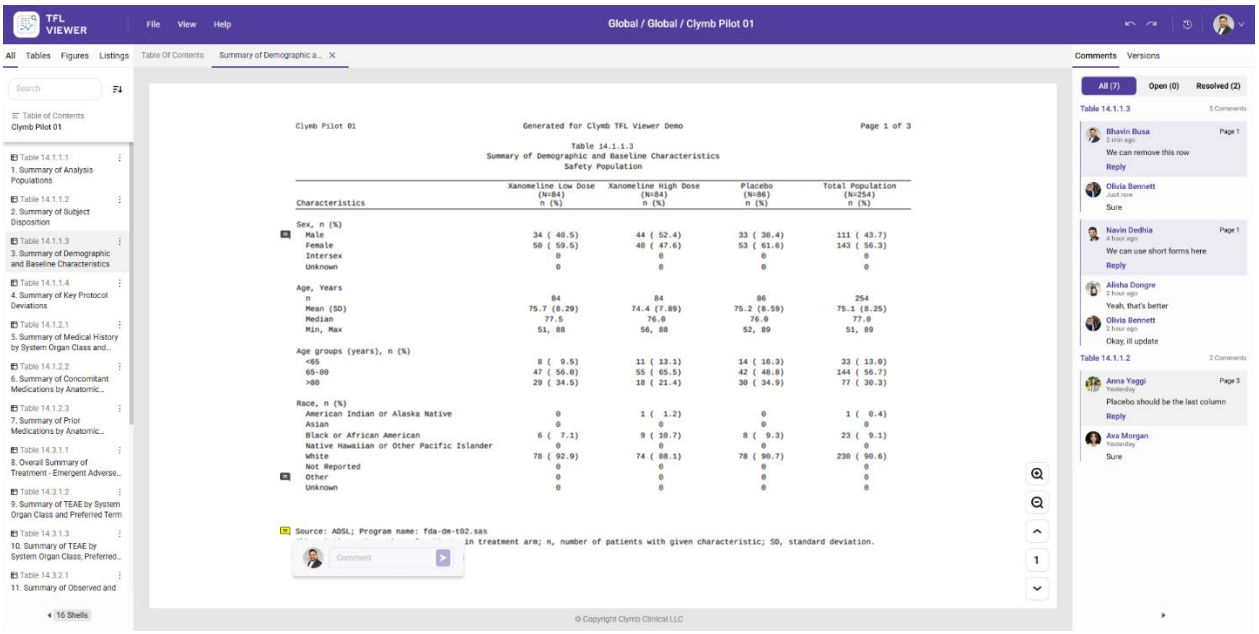


Figure 8: Centralized TFL review and collaboration in TFL Viewer. This figure illustrates intelligent navigation, in-context commenting, version comparison, and audit-ready review workflows supported by TFL Viewer.

DISCUSSION

The framework demonstrates how ARS can function as an operational backbone across the full TFL lifecycle, aligned with the broader CDISC 360 and 360i vision for standards-driven automation from study design through reporting [1, 2, 5]. By shifting effort upstream to structured specification, downstream execution and review become more consistent, traceable, and scalable.

Key observations include:

- Reduced duplication across shells, SAPs, programs, and outputs
- Flexible execution paths supporting R, SAS, and AI-assisted workflows
- Improved traceability linking analysis intent to reported results
- Future readiness for Dataset-JSON, ARM, and AI-enabled reporting

Supporting multiple execution engines within a single metadata-first framework allows organizations to modernize incrementally while maintaining standardization and control.

CONCLUSION

This paper presents a practical, end-to-end framework for automating the TFL lifecycle through Design and Specify, Execute, and Report and Review phases. By integrating TFL Designer, siera, atlas, AI Code Generator, and TFL Viewer around CDISC ARS metadata, the framework enables scalable automation while preserving transparency, traceability, and regulatory confidence.

As clinical trial complexity continues to increase, metadata-driven, multi-engine automation frameworks such as this will be essential for improving efficiency, consistency, and sustainability in clinical reporting.

REFERENCES

1. CDISC 360i Program Kickoff Presentation. Enabling Standards-Driven Automation from Study Design Through Reporting, February 2025. https://www.cdisc.org/sites/default/files/pdf/360i-Program-Kickoff20250218.pdf
2. CDISC 360 Project White Paper, June 2021. https://www.cdisc.org/sites/default/files/2021-06/CDISC_360_Project_White_Paper.pdf
3. CDISC Analysis Results Standard (ARS), Version 1.0, April 2024. https://cdisc-org.github.io/analysis-results-standard/
4. CDISC Analysis Results Standard User Guide, Version 1.0. https://wiki.cdisc.org/display/ARSP/Analysis+Results+Standard+User+Guide+v1.0
5. Busa, Bhavin.; Marshall, Richard.; LeRoy, Bess. '*All You Need to Know about the New CDISC Analysis Results Standards!*' PharmaSUG, May 2023.
6. TFL Designer Community: https://tfldesigner.org
7. Bosman, Malan. '*Advancing TFL Automation: R-Based Meta-Programming Powered by ARS*' PHUSE US Connect, March 2025.
8. Wachara, Julie. '*Let the Machine Do the Coding: A SAS-Based Framework for Automating TFL Generation with ARS Metadata*' PHUSE US Connect, March 2025.
9. U.S. Food and Drug Administration (FDA), *Safety Tables, Figures, and Listings (TFLs) Integrated Guide:* https://www.fda.gov/industry/study-data-standards-resources/safety-tables-figures-and-listings-integrated-guide
10. Japan Pharmaceutical Manufacturers Association (JPMA), *ADaM-Based Analysis and Reporting Considerations* . https://www.jpma.or.jp/english/reports/drug_evaluation_committee/adam_data.html
11. CDISC eTFL Portal, *Example TFL Packages and ARS-Aligned Templates.* https://www.cdisc.org/kb/etfl
12. AI Code Generator: https://clymbclinical.com/ai-code-generator/
13. Busa, Bhavin. '*Introducing TFL Viewer: Centralized Platform for TFL Review, Collaboration, and Approval*' PHUSE US Connect, March 2025.
14. TFL Viewer: https://clymbclinical.com/tfl-viewer/

ACKNOWLEDGMENTS

We would like to extend our gratitude to the entire Services and Solutions team at Clymb Clinical for their dedication and contributions. Their efforts have been instrumental in building an innovative solution that advances our vision of enhancing efficiency and quality in the analysis results generation, review, and approval process.

CONTACT INFORMATION

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

Malan Bosman
Lead Automation Engineer,
Clymb Clinical
mbosman@clymbclinical.com

Bhavin Busa
Principal & Co-founder,
Clymb Clinical
bhavin@clymbclinical.com