PharmaSUG China 2026 - Paper AA-195

ShellPainter & CDARS: An AI-Powered Digital Framework for Integrated Shell-TFL Automation

Jiapeng He, Mingliang Fan, Yanan Sui, Lin Jiang, Mingfeng Zhou, and Xuejie Zhou

ABSTRACT

Background: Tables, Figures, and Listings (TFLs) creation remains a labor-intensive process requiring significant coordination between statistical programmers, statisticians, and medical scientists. Traditional workflows involve manual shell design, repetitive parameter entry, substantial programming effort for each deliverable, and can't guarantee the consistency between shell and final TFLs.

Objectives: We present ShellPainter and CDARS (Clinical Data Analysis and Reporting System), an integrated web-based platform that transforms shell & TFL development through intelligent metadata management and AI-assisted automation.

Methods: ShellPainter provides an intuitive interface for designing study level master TFL shells, automatically capturing structural metadata (e.g., column hierarchies, table block definitions, and statistical model configuration), and tracking shell ongoing changes. CDARS extends this foundation by linking shell metadata with SDTM/ADaM annotations, enabling automated TFL generation & validation through SAS & R programs. The system leverages large language models (LLMs) to auto-select analysis contents in mock shell based on protocol/SAP, intelligently suggest variable/parameter mappings & analysis filters, auto debug & fix TFL metadata/program. Version control with full audit trails ensures regulatory compliance, while real-time visualization features streamline the review process.

Results: Implementation across 60+ clinical studies demonstrated 70% resource reduction in TFL delivery and significant decrease in programming errors. The AI-assisted metadata generation achieved 80-90% accuracy in suggesting appropriate analysis variables, parameters, and filters.

Conclusions: ShellPainter and CDARS represent a paradigm shift from document-centric to metadata-centric TFL development, enabling reproducibility, traceability, and substantial efficiency gains while maintaining the flexibility required for diverse analysis & reporting needs.

INTRODUCTION

Tables, Figures, and Listings are among the most resource-intensive deliverables in clinical-trial reporting. In a traditional workflow, statisticians define shell requirements in documents, programmers interpret those shells into metadata and code, reviewers evaluate outputs through separate systems, and teams repeatedly reconcile changes between shell, programming logic, and final deliverables. This creates duplicated effort, inconsistent interpretation, and limited traceability.

The prior CDARS paper [PharmaSUG China 2024 - Paper AA-10057] described an end-to-end digital platform for TFL automation, where mock shell content is parsed into title/footnote, column header, block, and value-level metadata; SDTM/ADaM annotations are attached using techniques such as fuzzy matching and semantic analysis; and dynamic SAS programs are generated from metadata. That paper established the value of metadata-driven TFL automation, but the broader challenge remained: the shell itself needed to become a structured, reusable, and automation-ready object rather than a static document.

ShellPainter addresses this upstream gap. ShellPainter is a centralized platform for company-, compound-, study- and task-level TFL shell generation and management, with centralized metadata, streamlined workflows, real-time visual editing, stronger standardization, and one-click flow to CDARS. Together, ShellPainter and CDARS extend automation from “metadata to TFL outputs” into a truly integrated “shell to TFL” lifecycle. And with AI implemented in both ShellPainter & CDARS, the automation level is further improved across this integrated “shell to TFL” lifecycle.

OVERALL FRAMEWORK

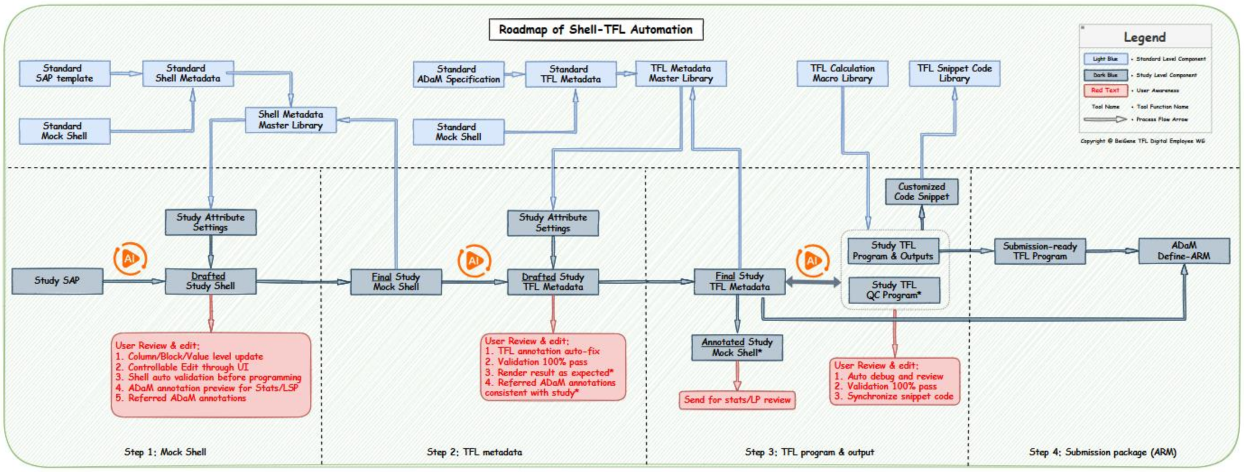

Figure 1. Integrated architecture of ShellPainter and CDARS

ShellPainter structures shell design into reusable metadata, while CDARS enriches that metadata with SDTM/ADaM annotations and automation logic to generate TFL programs and outputs. The shell-TFL overall roadmap describes the progression from CDARS platform proof-of-concept, metadata-driven TFL automation, ShellPainter development, one-button metadata flow to CDARS, and later intelligence shell–TFL automation with advanced AI. (Figure 1)

METHODS

1. SHELL DIGITALIZATION IN SHELLPAINTER

ShellPainter converts the shell from a static document into a structured, reviewable, and reusable analysis specification. ShellPainter also enables centralized study-level library management, direct shell creation from standard libraries, built-in metadata, real-time visualized editing, and one-click flow to CDARS.

A key process improvement is that shell-related information is maintained in ShellPainter and then synchronized to CDARS, rather than edited independently in multiple locations. If shell-related information needs updates, such as title, population, listing column label, or layout, users should update ShellPainter and then sync changes to CDARS. This design reduces ambiguity in ownership: ShellPainter becomes the source for shell structure, while CDARS becomes the downstream system for analysis annotation, program generation, and validation. (Figure 2)

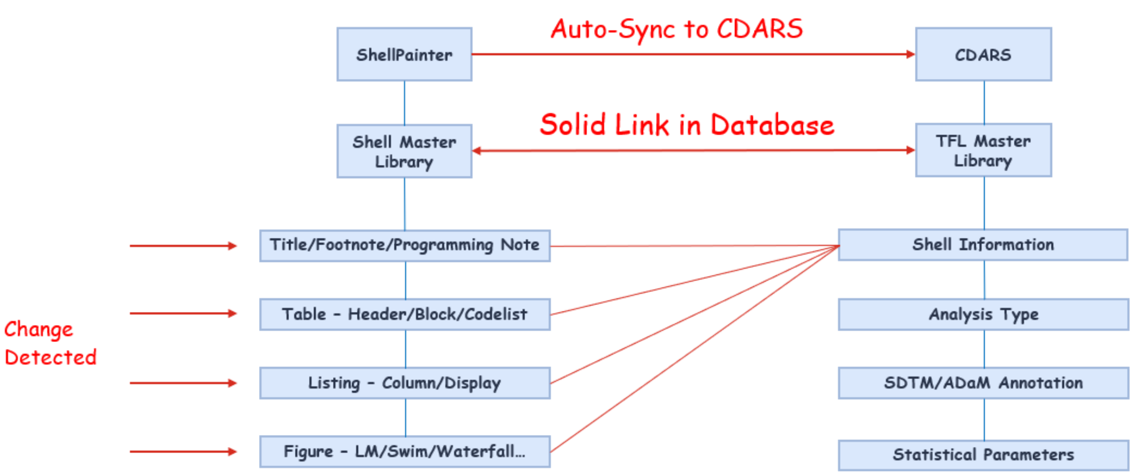

Figure 2. Controlled metadata flow from ShellPainter to CDARS

In CDARS, all shell-synchronized information is captured and presented to users through an intuitive interface. Key shell elements are clearly highlighted to distinguish synchronized content from manually edited components. This visual differentiation not only improves traceability but ensures consistency between shell design and downstream TFL generation.

2. CDARS: METADATA-DRIVEN TFL AUTOMATION ENGINE

CDARS extends shell metadata into executable analysis metadata by connecting shell content to SDTM/ADaM datasets, variables, parameters, filters, formats, and statistical methods. One key foundation of CDARS is to parse mock shells into structured metadata levels and attach SDTM/ADaM annotations using techniques such as text fuzzy matching and semantic analysis, followed by automated SAS program generation. (Figure 3)

These capabilities position CDARS not just as a program generator, but as a controlled automation engine with built-in validation and metadata governance.

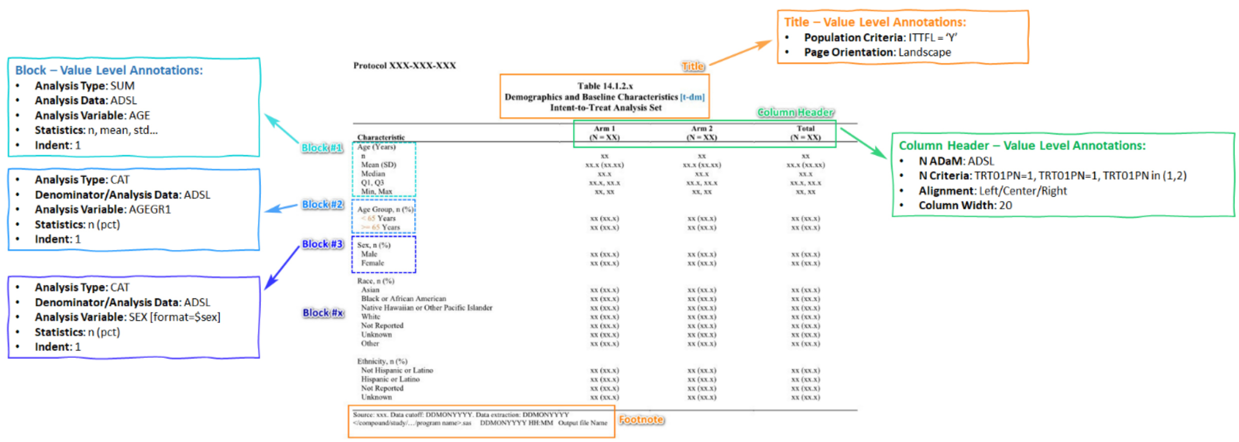

Figure 3. Digitalized TFL shell as metadata

3. AI-POWERED AUTOMATION LAYER

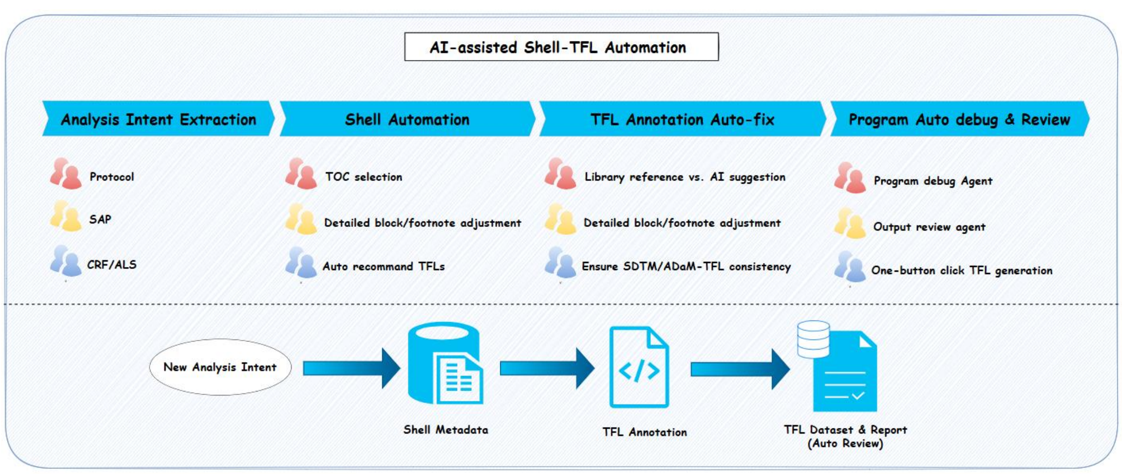

Figure 4. AI-assisted metadata intelligence loop

LLMs support analysis intent extraction, draft shell content generation in ShellPainter, variable/ parameter/ filter suggestion, and program auto debugging in CDARS. When new analysis intent is detected in SAP or in study ongoing discussion, the intelligence loop will be triggered to evaluate the analysis intent and automatically generate the shell/TFL metadata and analysis result following validated layout and structure. (Figure 4)

4. VISUALIZED QUALITY CONTROL

Real-time check: Both ShellPainter and CDARS have a rule-based cross-checking system to ensure accuracy of title/footnote, column header and block metadata. For instance, the block analysis type is always checked if it is filled in properly because of its importance for analysis. Real-time checking is also applied whenever metadata finishes editing. If the checking system finds any issues, then CDARS will highlight background of relevant cells as red immediately to alert. You may get detailed specifications by double clicking red cells (Figure 5).

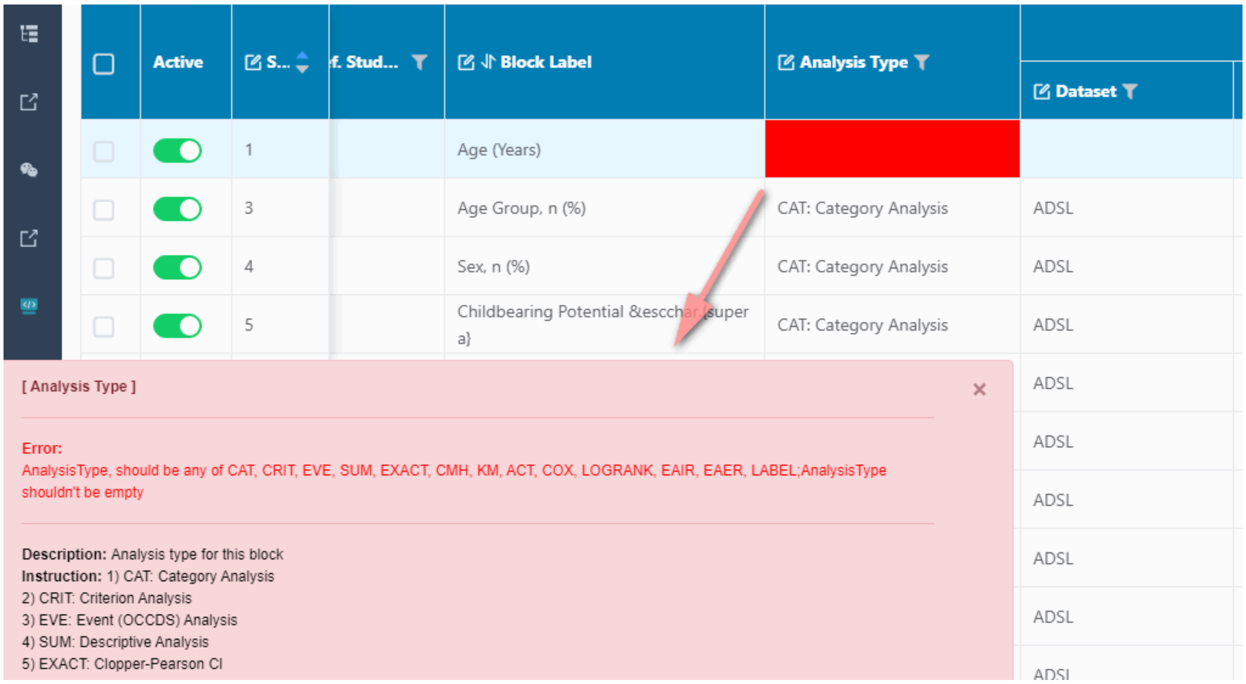

Figure 5. Real-time check

SDTM/ADaM-TFL crosscheck: Beyond TFL metadata check, SDTM/ADaM-TFL crosscheck is also performed in CDARS based on study level SDTM/ADaM variables information. Inappropriate metadata annotations which are either from reference or AI recommendation can be replaced with the specific and correct study SDTM/ADaM variables by utilizing study level specifications (Figure 6).

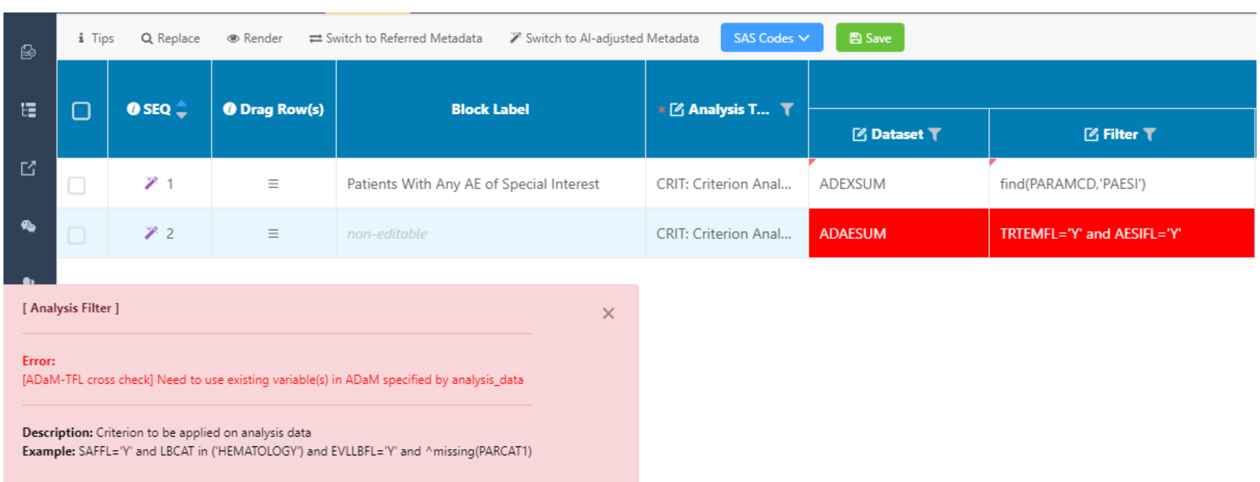

Figure 6. ADaM-TFL crosscheck

Real-time review with integrated information in TFL programming: Render function CDARS can visualize the output layout along with population, header, block level annotation behind. Besides, detailed format/function defined in specific block, SDTM/ADaM logic of analysis variable/parameters, statistical model code extracted from SAS log can also displayed in render page by popup floating window. This function can let user review all the related information behind TFL programming in one place and make the TFL review task straightforward, efficient and even joyful (Figure 7).

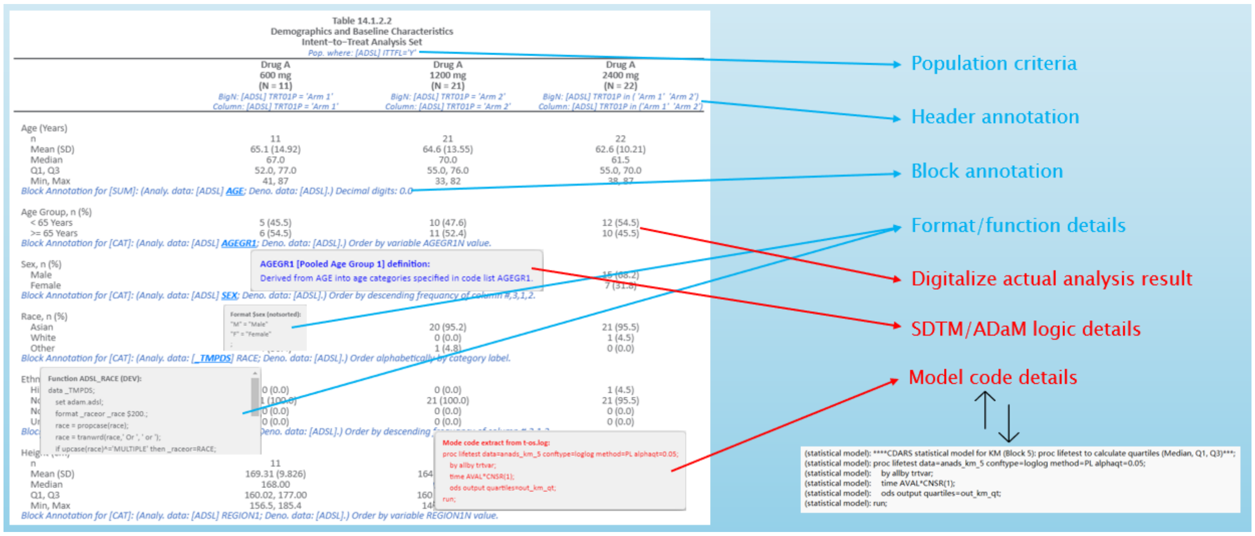

Figure 7. TFL review with integrated information

MEASURABLE OUTCOMES

1. LARGE OPERATIONAL SCALE

ShellPainter and CDARS have evolved beyond proof-of-concept into a production-oriented digital framework. It's an end-to-end automation platform from shell to TFL, a standard and extensive library containing 200+ unique TFLs, 60+ active studies, 100+ monthly active users across Stats and programming department, and comprehensive coverage various tasks like SMC/DMC, IA/CSR, SCS/SCE, IB/DSUR, publication, and ad-hoc use cases across the whole study lifecycle.

There is also a study specific analysis collection system in ShellPainter. When end users in study level found an analysis that is not covered in the TFL library, they can request TFL standard working group to implement this request in the platform, which makes the TFL library both standardized and extensive.

2. EFFICIENCY IMPACT

The integrated ShellPainter and CDARS workflow produced substantial efficiency gains by reducing manual shell handling, duplicate metadata entry, and repetitive programming. Large scale pilot states that ShellPainter and AI development improved metadata match rate from 75% to 95%, achieved metadata match correction rate from 70% to 90% in CDARS, and achieved 70% overall FTE saving rate for TFL delivery (compared to 55% FTE saving rate when CDARS is solo used without AI approaches).

3. QUALITY AND TRACEABILITY IMPACT

The integrated workflow improves quality by enforcing alignment between shell metadata, CDARS metadata, generated programs, and final outputs. In CDARS, besides metadata real-time check and render that mentioned in “Methods” section, more functions to enhance traceability are also developed, including timestamp comparison and customized code content comparison between metadata and TFL programs, notifications during TFL program generation, and E2E metadata flow.

ShellPainter also has the shell track-change function between different task and timestamp, central & batch editing functions, and RTF in-text shell download function. These functions also improved auditability, structured review, and end-to-end traceability.

4. AI-ASSISTED ACCURACY

Based on the statistics in 5 studies, the AI-assisted metadata generation achieved approximately 90% accuracy in suggesting appropriate analysis variables, parameters, and filters in table block analysis. Accuracy was assessed by comparing AI-suggested metadata against finalized programmer-reviewed metadata across selected outputs.

KEY DIFFERENCES COMPARED TO COMMON AUTOMATION APPROACH

Below table shows key differences between ShellPainter & CDARS compared to common TFL automation. We believe such differences build a solid foundation to the path of full TFL automation in clinical pharma industry.

Dimension	Common Automation	ShellPainter + CDARS
Starting point	Existing shell document or manual metadata setup	Structured shell design begins with ShellPainter and flows into CDARS.
Metadata ownership	Metadata is often recreated separately by programmers	Shell-related metadata is maintained in ShellPainter and synchronized to CDARS.
Automation scope	Program generation only	Shell design, metadata flow, annotation, program generation, validation, review, and submission ready.
Review model	Static document review	Structured review with ShellPainter track-change and metadata-level visibility.
Quality control	Output comparison and manual checks	Visualized and integrated review, end-to-end evidence chain validation.
Reuse	Study-specific macros or copied programs	Standard/extensive library with 200+ unique TFLs reported in internal highlights.
AI role	Standalone prompt assistance	Embedded AI roadmap for TOC selection, analysis-content adjustment, metadata auto-fix, program debug, and natural-language updates.

Table 1. Differentiation summary table for ShellPainter + CDARS

FUTURE PATH TO INTELLIGENT CLINICAL ANALYSIS

Looking ahead, the future of ShellPainter & CDARS is to evolve from an integrated automation platform into an intelligent clinical analysis ecosystem. By combining the functions below, the platform can further reduce manual interpretation, strengthen traceability across the study lifecycle, and transform clinical reporting from task-based execution into intent-driven, knowledge-reusable automation. These future capabilities represent the next step toward a more standardized, scalable, and intelligent model for clinical analysis and regulatory reporting.

Digital customer supporter: A natural extension of current LLM capabilities is the introduction of an AI-powered chatbot within ShellPainter and CDARS. This chatbot will enable users to query and modify TFL metadata using natural language, generate or refine shells without manual navigation, debug metadata or program issues interactively, and ask context-aware questions (e.g., “Why is this population filter applied?”). This reduces the learning curve for new users and shifts interaction from UI-driven workflows to intent-driven workflows, significantly improving usability and efficiency.

End-to-end metadata flow: Future development aims to extend the current shell-to-TFL integration into a full lifecycle metadata pipeline, connecting protocol, SAP, shell, TFL, and CSR in a continuous and traceable workflow. By enabling seamless propagation of analysis intent across systems, this approach establishes a single source of truth for clinical reporting, minimizes repeated interpretation across roles, and aligns with emerging CDISC standards such as ARS/ARD, ultimately improving consistency, traceability, and regulatory readiness.

One-button TFL generation: A key vision of the platform is to achieve near-zero manual intervention in TFL production through fully automated metadata completion and validation. With AI-assisted metadata initialization, built-in quality checks, and automated execution pipelines, users will be able to generate complete sets of TFL outputs in CDARS with a single action. This represents a shift from automation-assisted programming toward programming-free TFL generation, significantly accelerating study delivery timelines.

Clinical data visualization: To complement traditional tabular outputs, CDARS will evolve to support advanced and interactive clinical data visualization capabilities. These include dynamic dashboards linked to TFL metadata, auto-generated figures based on analysis intent, and integrated visualization tools for both patient-level and population-level exploration. Such capabilities enable more intuitive data review, facilitate decision-making, and bridge the gap between statistical programming and clinical interpretation.

Cross-study TFL recommendation: With the continuous expansion of standardized metadata libraries, the platform will support intelligent recommendation capabilities across studies. By leveraging historical metadata and study similarities, the system can suggest appropriate TFL structures, analysis content, and reusable components, as well as identify missing or inconsistent outputs. This enhances study startup efficiency, promotes standardization, and enables effective knowledge reuse across therapeutic areas and development programs.

CONCLUSION

ShellPainter and CDARS demonstrate how clinical reporting can evolve from a document-centric process into a metadata-centric, AI-assisted digital workflow. By structuring shell design at the source, synchronizing shell metadata into CDARS, enriching that metadata with SDTM/ADaM annotations, and generating validated SAS/R programs, the framework reduces repetitive manual effort while improving traceability and consistency.

The key contribution is not only automation of TFL outputs, but integration of the full shell-TFL lifecycle: design, metadata flow, annotation, programming, validation, review, and reuse. With AI-assisted metadata generation, protocol/SAP interpretation, and future natural-language metadata interaction, ShellPainter and CDARS provide a practical blueprint for the next generation of clinical trial reporting automation.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to all developers' contribution and cooperation, as well as all users' valuable feedback and comments. Especially to our CTO Leo Li, the platform and its developers grew and improved rapidly with his oversee, guidance and solid support. Also allow me to introduce our creative developers and perfect teamwork more:

- **Tingting Zeng** originated the platform design and development.
- **Jinling Li** designed the framework of shell/TFL metadata.
- **Mingliang Fan**, the owner of plenty of SAS calculation macros and TFL master library.
- **Yanan Sui**, front-end engineer who built highly intuitive UI to make the platform available for 100+ end users across Stats and programming department.
- **Lin Jiang,** back-end developer who developed complex metadata check and validation program automation to ensure overall quality and make findings traceable.
- **Mingfeng Zhou** implemented shell review and other operational functions into ShellPainter.
- **Xuejie Zhou** initiated the AI exploration on TFL metadata automation.
- I took the opportunity to be a product manager and back-end developer.

CONTACT INFORMATION

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

Jiapeng He
jiapeng47.he@gmail.com