PharmaSUG China 2026 - Paper LS136-2026

Exploring the Architecture of Statistical Programming Languages in the AI Era

Jiang Chao, Wang Li, Wang Juan, Su Ya, Yan Na, SPH China;

ABSTRACT

The rapid advancement of artificial intelligence (AI) has fundamentally reshaped the landscape of statistical programming within the pharmaceutical and clinical research industries. The AI-enabled era has catalyzed a shift toward intelligent, automated, and highly diversified workflows. Traditional statistical programming—once dominated by manual coding, repetitive tasks, and rigid SAS-centric toolchains—now faces both unprecedented opportunities and new compliance challenges. In this context, the Shanghai Pharma statistical programming team has explored a modernized programming language framework. By integrating automation, multi-language workflows (SAS, Python, R, VBA), and AI-assisted tools, we have enhanced operational efficiency, output quality, and regulatory robustness. This paper also discusses the evolution of the "SAS Programmer" into a multi-disciplinary Clinical Data Innovator.

INTRODUCTION

I would like to share our team's journey in modernizing statistical programming. We will explore how we step out of our traditional comfort zone and leverage the architecture of modern statistical programming languages in the AI era.

My presentation will focus on three key areas:

- The platforms or tools we developed: Demonstrating how we automated daily tasks;
- The multi-language programming architecture: Detailing which languages (Python, R, VBA, SAS) are used for specific tasks;
- Redefining the Programmer in the Age of AI: How our work scope and capabilities are expanding beyond traditional boundaries.

THE PLATFORMS OR TOOLS DEVELOPED

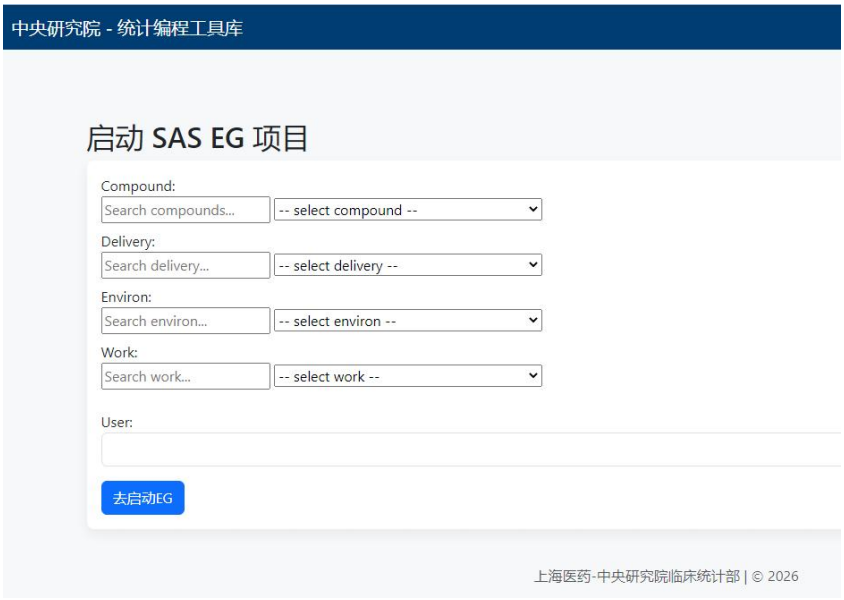
To break free from repetitive manual labor, our team developed several integrated platforms.

THE INTEGRATED PROGRAMMING PLATFORM

As programmers' tasks vary, utilizing unified tools is critical for quality assurance and efficiency.

The Integrated Programming Platform, including:

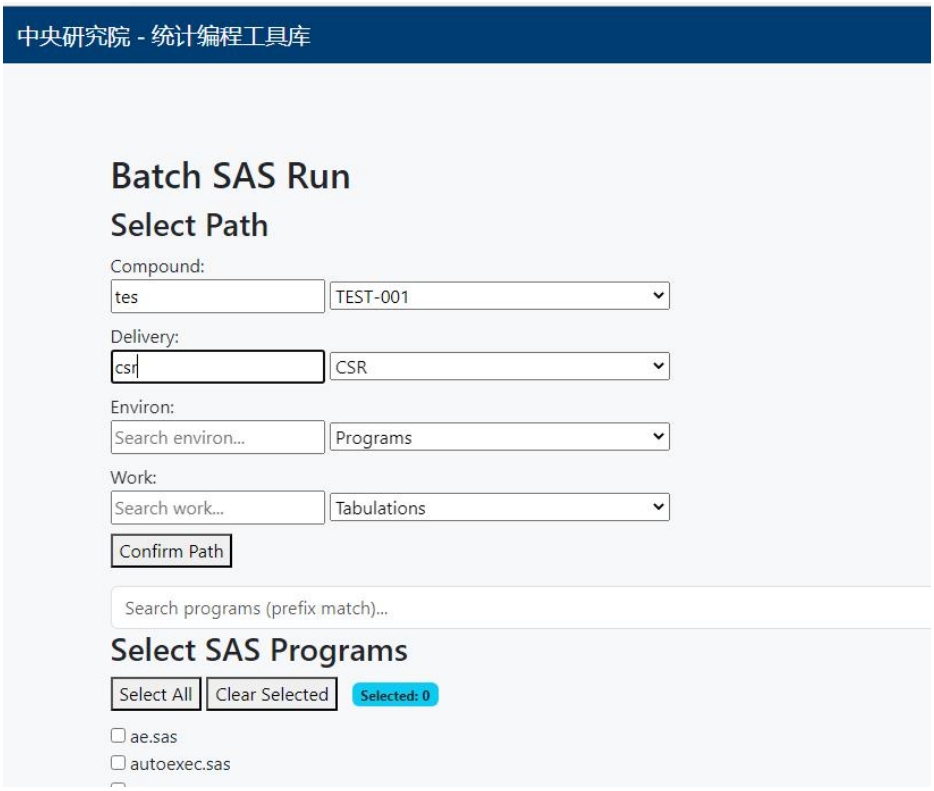
- **Open SAS Enterprise Guide (EG) Integration**: The platform opens SAS EG and automatically loads study paths, public macros, formats, or global values based on user selection (compound, delivery, environment). Users can open multiple EG windows for different studies, while administrators manage studies easily backstage.



- **THE ACRF/SDTM GENERATION TOOLS:** For aCRF, Using the DTS (EDC Dataset Metafile) and SDTM IG file, our tool automatically assigns variables in the blank CRF. For SDTM, It generates an adjustable aCRF Spec that can be tailored to each study. Inputting the updated Spec back into the tool yields the final aCRF PDF. The tool automates the generation of SDTM Specs and the corresponding SAS code files for each domain, drastically reducing boilerplate coding.

- **TLF Generation**: This function generates all Tables/Listings/Figures (TLF) SAS programs at once. It auto-populates standard program headers (name, purpose, author, date) and intelligently inserts required macros (e.g., %m_socpt for AE SOC/PT tables), along with general macros for reading titles, footnotes, and checking logs.

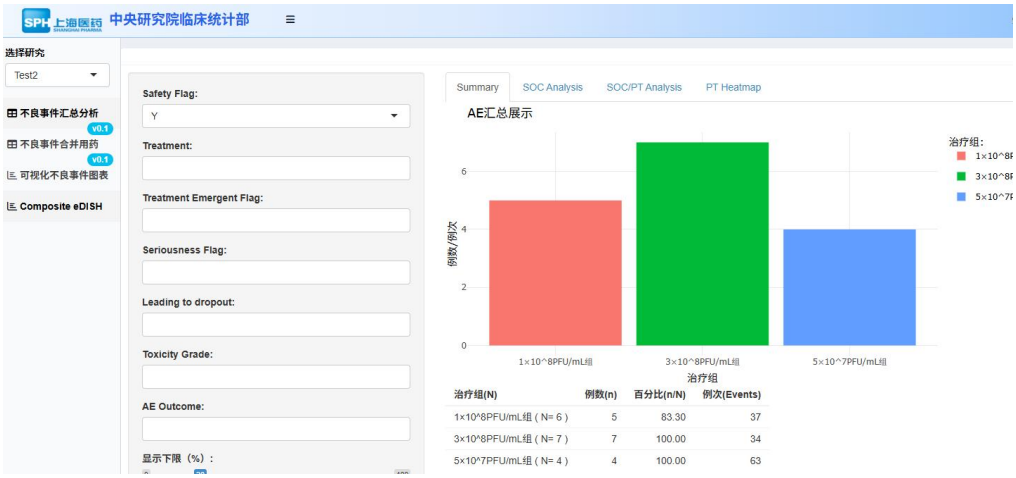
- **Batch Run**: Users can freely search, select, and batch-run programs. The process provides a visualized log window, streamlining the validation and execution process.



THE MEDICAL MONITORING PLATFORM

Using SDTM/ADaM datasets, we extended our services to medical monitoring, particularly for safety analysis..

Below is the screenshot of monitoring platform.



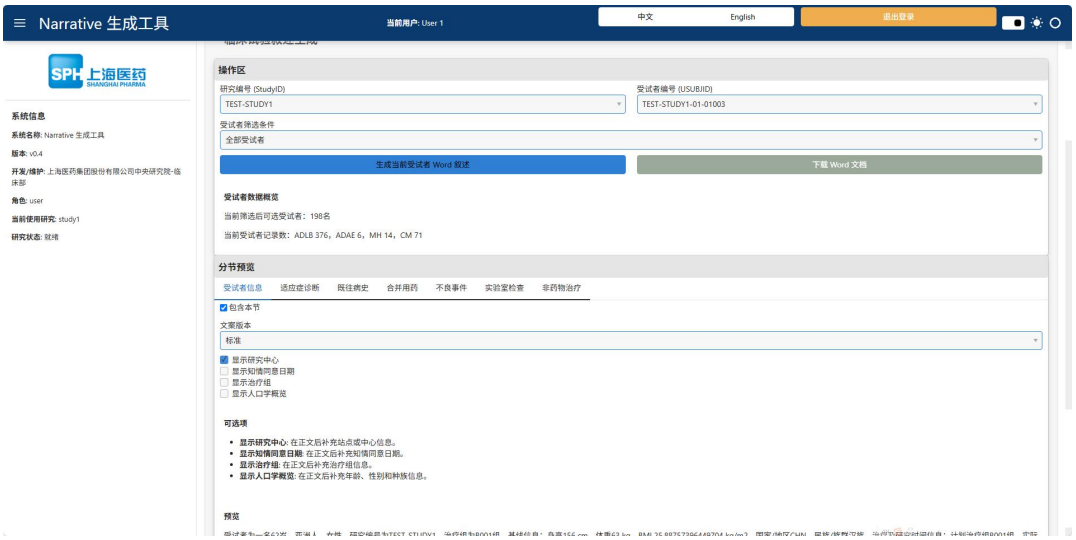
The platform includes:

- **AE Summary Analysis**: Displays summary comparisons between treatment groups via histograms and charts, with dynamic criteria controls (TEAE, TRAE, SAE, outcomes).
- **SOC/PT Analysis**: Provides interactive tables and histograms, allowing users to filter by TEAE, SAE, toxicity grade, and percentage bars.
- **AE and Concomitant Medication Relationship**: Visualizes the temporal relationship between AEs and medications for specific subjects, with customizable date ranges.

THE NARRATIVE GENERATION PLATFORM

Drafting patient narratives is time-consuming. Our platform uses SDTM/ADaM datasets to auto-generate draft narrative documents..

Below is the screenshot of this platform.



The platform includes:

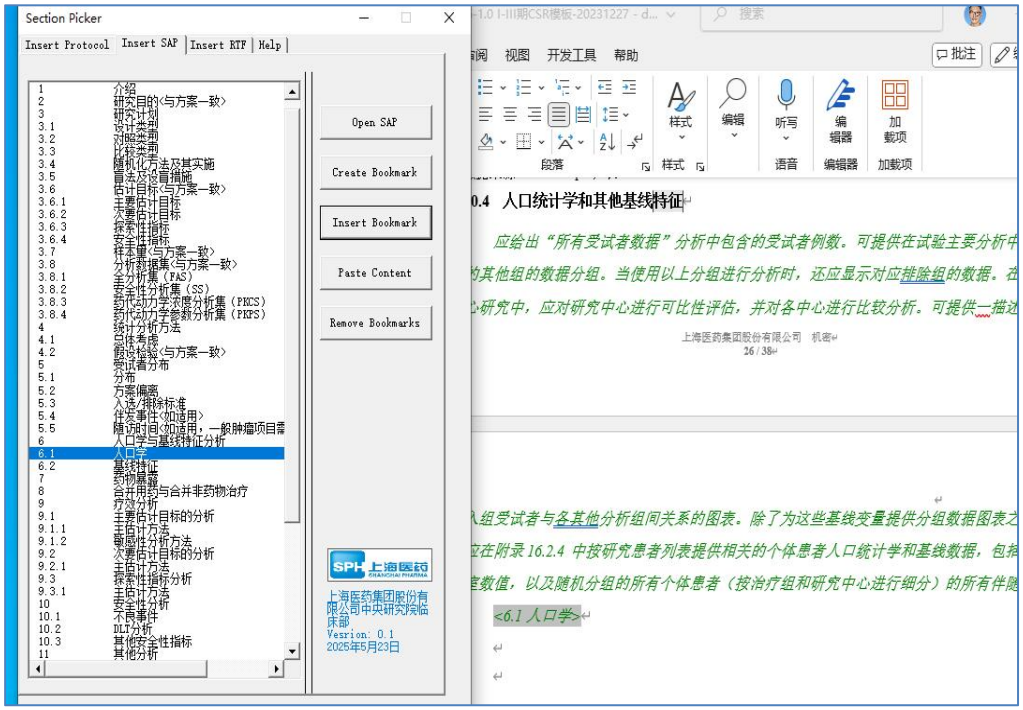
Users select a subject (USUBJID) and the tool compiles data across domains: Demographics, Disease History, Concomitant Medications, Adverse Events, Lab Tests, and Non-Medicine Therapy.

Users can select specific templates and options (e.g., displaying site, consent date, treatment group) to fit the project.

The auto-generated text is fully editable within the platform, and the final Word document can be downloaded directly. This turns a multi-hour manual writing task into a few minutes of review.

THE CSR DRAFTING TOOL

For the final clinical study report (CSR), we developed a Word-based drafting tool.



The tool includes:

- **Placeholder Insertion**: Writers can insert placeholders linked to the Protocol and SAP. Once these documents are finalized, a simple refresh populates the CSR with the latest content.
- **TLF Integration**: Placeholders for TLFs are inserted from dry-run outputs. After database lock, real outputs are refreshed instantly based on the finalized datasets.

THE MULTI-LANGUAGE PROGRAMMING ARCHITECTUR

Beyond traditional SAS language, with AI assistant, we learnt and used multiple programming languages based on users' demands.

Python: The Engine for Web UI and Data Processing

- **Web-based UI Framework (Flask, Panel)**: We use Python Flask to build the lightweight web servers and frameworks for our Integrated Programming and Medical Monitoring Platforms. Panel packages are used to create interactive web dashboards.
- **Data Process and Analysis (pyreadstat, pandas, json)**: Python reads SAS datasets (.sas7bdat), converts them to JSON structures, and performs advanced data manipulations. It acts as the bridge between SAS datasets and modern web applications..

Template and Text Processing (Jinja2, YAML, docx)

- We utilize Jinja2 and YAML files to create dynamic templates. The python-docx package allows

us to programmatically write, edit, and format Word documents, powering our Narrative Generation and CSR Drafting tools.

VBA: Document Transformation across Office Ecosystems

- VBA is used to implement content exchange across different Office applications. For example, in the CSR Drafting Tool, VBA inserts bookmarks in Word and automatically populates them with content extracted from Excel or other documents. It handles layout modifications and ensures style consistency, enabling seamless automation of text insertion.

R: Advanced Data Visualization

- Using R together with R Shiny, we provide interactive data visualization environments. While SAS is great for tables, R excels in generating complex plots (scatter plots, box plots, outlier detection) and heatmaps. In our Medical Monitoring Platform, R Shiny highlights patterns of variable intensity, supporting cluster analysis and correlation exploration dynamically.

SAS: The Unshakable Foundation

- SAS remains the gold standard for regulatory compliance and CDISC standards. We use SAS for robust data derivations (SDTM/ADaM) and validated TLF generation. The difference is that SAS is no longer the only tool; it is now a powerful module within a larger, orchestrated pipeline.

REDEFINING PROGRAMMER IN THE AGE OF AI

How Our Work Will Change

- **Language Diversity**: We are no longer confined to SAS. Python, R, Julia, and modern frameworks like Shiny or Dash coexist in the same workflow, each chosen for its specific strengths.
- **AI Integration**: Code generation, anomaly detection, and automated reporting are part of our daily toolkit. AI is not just assisting—it is accelerating the pace of development.
- **Workflow Evolution**: Instead of monolithic SAS scripts, we now orchestrate modular pipelines that combine multiple languages, APIs, and AI services..

Considerations for Tool Development

- **Interoperability**: Tools must bridge SAS with other ecosystems, ensuring smooth data exchange and compliance with CDISC and regulatory standards.
- **Transparency & Auditability**: AI-assisted outputs must be traceable, with clear metadata and reproducibility.
- **User Experience**: Interfaces should empower statisticians and medical writers to interact with data intuitively, not just through code.
- **Scalability**: Solutions must handle both traditional clinical datasets and modern, high-volume real-world data.

Transformation of the “SAS Programmer”

- **From Coders to Architects**: The role evolves from writing individual SAS macros to designing multi-language, AI-enabled workflows.

- **From Executors to Innovators**: We must evaluate which language or tool best solves a specific problem, rather than defaulting to SAS out of habit.
- **From Specialists to Integrators**: Future programmers will integrate SAS, Python, R, and AI seamlessly. Our work scope has expanded from purely generating datasets and TLFs to providing end-to-end solutions—including medical monitoring, narrative automation, and CSR drafting.

Planning for the Future

- **Upskilling**: Invest time in learning Python, R, and machine learning fundamentals.
- **Mindset Shift**: Embrace AI as a collaborator, not a competitor.
- **Strategic Vision**: Position ourselves as leaders in automation, compliance, and innovation—driving the next generation of clinical programming.

CONCLUSION

In conclusion, the age of AI does not diminish the role of the statistical programmer—it elevates it. As we have demonstrated through our integrated platforms, the modern programming environment is a symphony of languages: Python for web architecture and data wrangling, R for dynamic visualization, VBA for Office automation, and SAS for rock-solid regulatory compliance.

This multi-language architecture frees us from the drudgery of repetitive manual coding. By automating aCRF, SDTM, TLF generation, medical monitoring, narratives, and CSR drafting, we reclaim valuable time. More importantly, our work scope has fundamentally expanded. We are no longer just "SAS Programmers" generating tables behind the scenes; we are system architects, data integrators, and clinical data innovators providing comprehensive, cross-functional solutions.

The challenges of the AI era—compliance, interoperability, and transparency—are real, but they are surmountable with the right architecture. By expanding our toolkit, rethinking our workflows, and redefining our professional identity, we position ourselves at the forefront of clinical research innovation. The future of statistical programming is not just about writing code; it is about orchestrating intelligence to deliver deeper insights and better patient outcomes.

REFERENCES

Dony Unardi, Nina Qi. 2026. “Empowering Clinical Data Review with Interactive Patient-Level Visualisations in {teal}.” Phuse-US, PAP_DV09

Steve Ross, Ilan Carmeli. 2024. “Optimizing Clinical Research: Using AI for Automated Validation of Output Tables against ADaM”. Phuse-US, PAP_ET06

https://pharmaverse.org/

https://github.com/insightsengineering/teal

https://posit.co/products/enterprise/connect/

CONTACT INFORMATION

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

Jiang Chao
SPH China
jiangchao@sphchina.com