An AI-Powered Metadata-Driven CDISC ADaM Dataset Automation

PharmaSUG Conference

Surendra Gunti M.Tech, M.Pharm

Disclaimer

The information included in this presentation represents the opinion of the speaker, and it is no way related to Eli Lilly and Company or its affiliates. No one can use the presentation contents as a base for any claim, demand, or cause of action and, also no one is responsible for any loss incurred based upon

Standards Metadata Repository

Standards | Automation | Metadata | Process

A metadata-driven framework that standardizes, automates, and governs clinical data programming through defined processes—producing consistent, and inspection-ready outputs.

Foundational framework enabling the Standards automation

This separates standard automation from study-specific logic, enabling scalable delivery without compromising quality or compliance.

Why Metadata-Driven ADaM Automation?

Industry challenges driving the need

Scale

Growing study portfolios demand consistent ADaM delivery across therapeutic areas and teams without proportional headcount increases.

Complexity

Study-specific requirements, evolving standards, and diverse data sources make manual programming error-prone and time-consuming.

Compliance

Regulatory expectations for inspection readiness, and auditability require embedded governance — not afterthought review.

A metadata-driven automation engine that standardizes ADaM construction — generating draft programs directly from MDR specifications, ensuring consistency across every study.

ADaM Automation Overview

Three pillars of the automation framework

Metadata-Driven

ADaM specifications are maintained centrally in the MDR, ensuring consistency across all studies.

Automated Generation

Driver and dataset programs are generated automatically from metadata, eliminating manual coding errors and accelerating timelines.

Independent QC

The framework preserves independent replication pathways, ensuring regulatory compliance and data integrity.

End-to-End Workflow

Metadata flows from MDR into program generators, producing validated ADaM datasets with full audit trails and independent QC.

End-to-End Programming Workflow

ADaM Automation Workflow

End-to-end process from specifications to production datasets

01 **Complete ADaM Specs**

Define specifications in MDR

02 **Export from MDR**

Export ADaM Specs from MDR

03 **Save to Study Folder**

Save Specs in study Folder

04 **Run Automation**

Run adam_build_first_draft_programs

SPECIFICATION

AUTOMATION

05 **Review Programs**

Review/Update Programs (Study Logic)

06 **Execute Driver**

Execute Driver Program (.rds/.xpt)

07 **Review Messages**

Review ADAM_Review_MSG

REVIEW

SPECIFICATION (Steps 1-3)

Define ADaM specifications in MDR, export metadata from MDR, and save to the snapshot folder for version control.

AUTOMATION (Step 4)

Run adam_build_first_draft_programs to auto-generate R code from the exported specifications.

REVIEW (Steps 5-7)

Review and customize generated programs for study-specific logic, execute the driver, and validate output messages.

Where GenAI Fits

Programming Assistance Tool — GenAI-assisted code generation within the ecosystem

1 Custom Variable Identification

Metadata flags variables that cannot be auto-generated, targeting them for GenAI assistance

2 GenAI-Assisted Code Generation

AI generates R code for flagged variables with human review at every step

KEY SAFEGUARDS

Human-in-the-Loop

Full control at every step

Immutable Audit Trail

Complete traceability

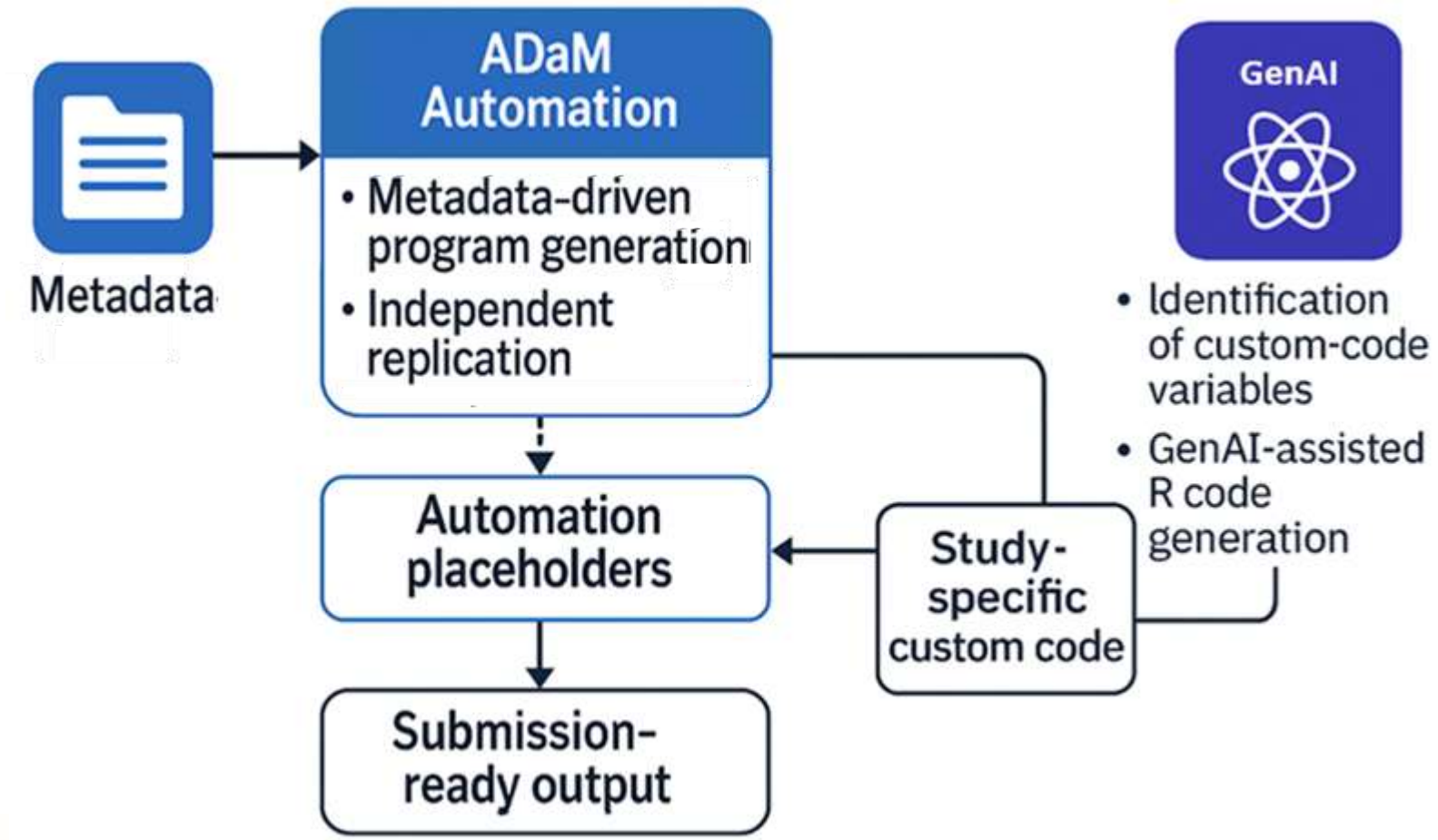
Seamless Integration

Fits existing workflows

Tool focuses only on the custom code gap — variables that our automation cannot handle. Everything else is fully automated.

MDR + GenAI Integrated Architecture

How standardized automation and GenAI-assisted coding work together



GenAI Assistance— Complete Workflow

PHASE 1: Setup & Input

1 — Upload Driver

Browse and select your .R driver program

2 — Select Variables

Choose custom-code variables to generate

3 — Generate Prompt

GenAI creates prompt from ADaM specs

4 — Edit/Save Prompt

Review and refine the AI-generated prompt

PHASE 2: Generation & Output

5 — Generate Code

Select variables and trigger code generation

6 — Review Output

Code files saved to your Study location

7 — Test & Iterate

Validate results; re-iterate or edit directly

Business Value

Cycle Time & Quality Impact

Faster delivery with built-in inspection readiness

CYCLE TIME REDUCTION

Auto-Generated Programs
From MDR specs — no boilerplate recreation

Instant Regeneration
Programs rebuilt automatically when specs change

Review Over Coding
Programmers focus on verification, not blank-page work

QUALITY & INSPECTION READINESS

Standardized Logic
Consistent transformation patterns across all teams

Immutable Audit Trail
Every specification change logged

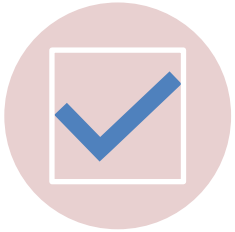
Early Issue Detection
Spec-level checks catch problems before coding begins

Impact: Weeks of setup → days of focused review. Every output standardized, traceable, and inspection-ready.

Cross-Study Consistency
Uniform output across teams and TAs

Inspection Confidence
Embedded governance with full audit transparency

PharmaSUG Takeaways



Practical GenAI use in regulated environments

Scalable enterprise ADaM delivery model

Thank you

Any questions

- Name: Surendra Gunti
- Organization: Eli Lilly and Company
- Address: 1st Floor, Building Primrose (7B), Embassy Tech Village, Outer Ring Road, Devarabisanahalli.
- City, State ZIP: Bengaluru, 560103
- Work Phone: 9160565758
- E-mail: Surendra.gunti@lilly.com