Single Day Events

Trivandrum, India

20 June 2026

phuse.global

connect·share·advance

Enhancing Programming Efficiency through AI-Enabled Clinical Data Standards

Paul Kirubakaran & Sivapreetha
Icon plc

phuse.global

Disclaimer

The views and opinions expressed in this presentation are solely our own and do not represent the official position of ICON plc or its affiliates.

Abstract

The adoption of clinical data standards such as CDISC has long supported consistency and regulatory compliance in clinical trials. However, increasing data complexity and compressed timelines demand more efficient programming approaches. This paper explores how AI and ML can enhance standards driven programming workflows.

Use cases where AI assisted tools automate dataset mapping, metadata generation, and validation processes within SDTM and ADaM frameworks will be discussed. By leveraging standardized structures, these tools enable faster code generation, improved traceability, and reduced manual effort. The findings highlight that integrating AI with established data standards not only enhances productivity but also strengthens submission readiness.

phuse

connect·share·advance

AGENDA

Today's Overview

01 The Challenge
Data Complexity and Timeline Pressures

02 Standards as AI Foundation
CDISC Enabling Intelligent Automation

03 AI Use Cases
SDTM Mapping, ADaM Derivation and Validation

04 Tools and Technologies
Real World AI tools for Clinical Programming

05 Findings and Outcomes
Productivity Gains and Submission Readiness

AI Ecosystem

connect·share·advance

1950s

Artificial Intelligence (AI)

Human intelligence exhibited by machines

1980s

Machine Learning (ML)

AI systems that learn from historical data

2010s

Deep Learning (DL)

Neural networks that mimic human brain function

2020s

Generative AI (Gen AI)

Foundation models that create original content

NLP

Natural Language Processing

Bridges classical methods and LLMs

LLMs

Large Language Models

Foundation models trained on vast text using transformer architecture

The Challenge

Why traditional approaches are no longer enough

connect·share·advance

Data Complexity

Clinical studies today generate vastly more complex and voluminous data than in previous years

Timeline Pressure

Sponsors and CROs face increasingly compressed programming schedules driven by competitive and regulatory demands

Repetitive Workload

A significant share of programmer effort is consumed by repetitive tasks that offer little opportunity for expertise

Key Pain Points

- Manual SDTM mapping is time consuming, error prone and difficult to audit
- Metadata documentation consistently lags behind programming deadlines
- Validation cycles consume a disproportionate share of submission timelines
- Regulatory scrutiny demands tighter traceability with every new submission
- Inconsistent coding practices across teams increase review and QC burden

"The volume, variety, and velocity of clinical data has outpaced the traditional programmer's toolkit. AI is no longer optional"

CDISC Standards as AI Foundation

Structured standards create the perfect substrate for AI driven automation

connect·share·advance

Raw Clinical Data

EDC, eCRF, Lab Systems, Wearables, Devices

SDTM Domains

Interventions , Events, Findings and other Domains

ADaM Datasets

ADSL, BDS, OCCDS and other Analysis Datasets

TLFs and Regulatory Submission

Tables, Listings, Figures and Submission ready Package

Why Standards Enable AI

- Consistent variable naming
- Controlled Terminologies
- Defined data relationships
- Predictable Structure
- Built in Validation rules

Use Case 1: AI Powered SDTM Dataset Mapping

Automating Raw to SDTM variable assignment using ML and NLP

connect·share·advance

Before Manual Approach

1. Programmer reviews CRF page by page
2. Manually maps each variable to SDTM spec
3. Cross references CDASH to SDTM guide
4. Manually writes Mapping Specifications
5. QC Reviewer checks all mappings individually
6. Cycle repeats for every New study

After AI Assisted Approach

1. AI ingests aCRF and protocol documents
2. NLP suggests SDTM domain + variable matches
3. Programmer reviews and approves suggestions
4. Auto generates mapping specifications
5. AI validation flags discrepancies instantly
6. Reusable model improves with each study

From CRF to ADaM Dataset

Three connected AI assisted steps

connect·share·advance

STEP 1 OF 3

Annotated CRF Generation

NLP driven CDASH to SDTM field classification on source CRF pages

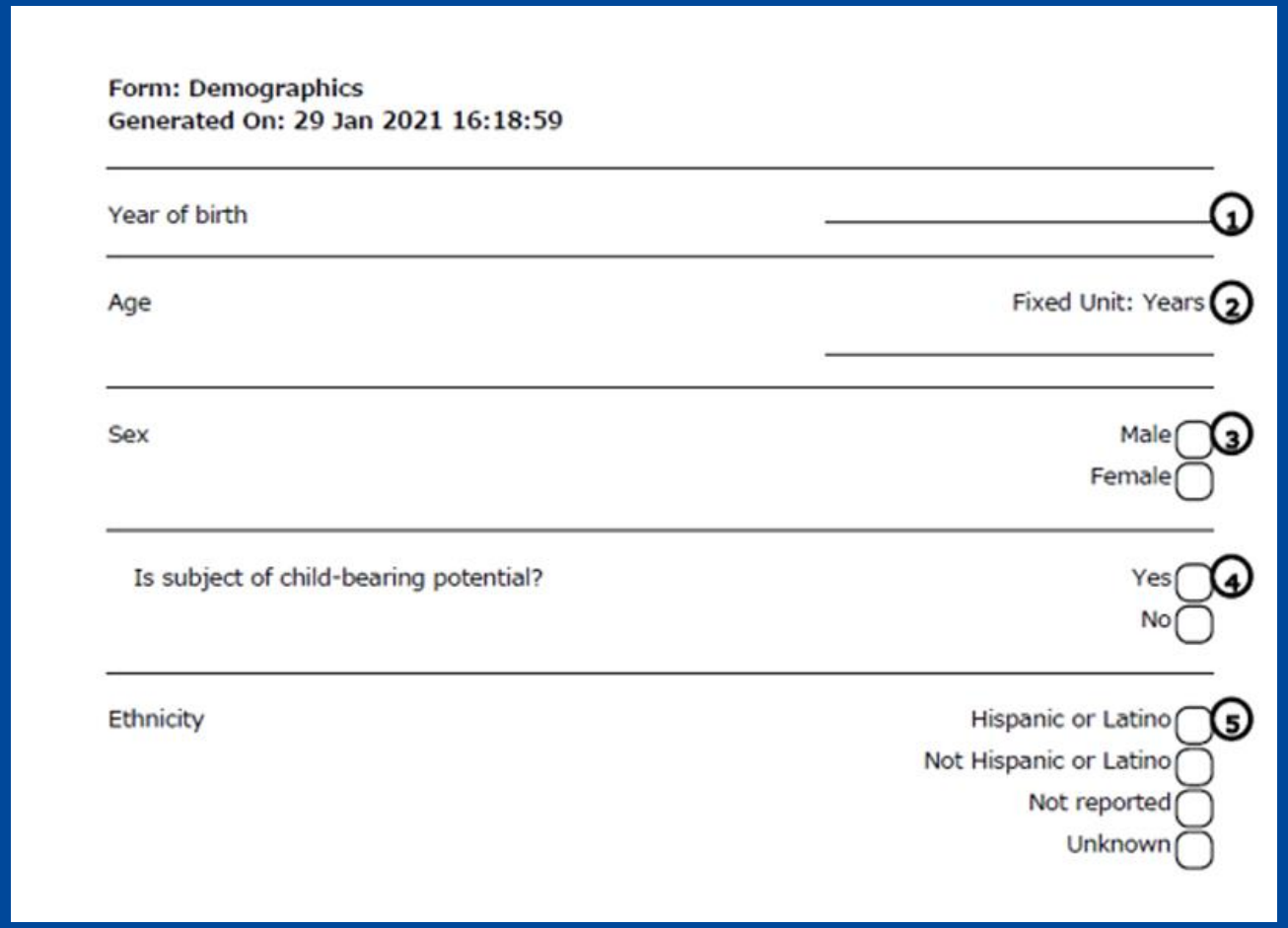
Annotated CRF Generation

NLP driven CDASH to SDTM field classification on source CRF documents

connect·share·advance

Files Required

- DM CRF pages
- CDASH Implementation Guide v2.3
- SDTM Implementation Guide v3.4
- CDISC CT & Protocol



Form: Demographics
Generated On: 29 Jan 2021 16:18:59

Year of birth _____ (1)

Age _____ Fixed Unit: Years (2)

Sex — Male ☐ (3) Female ☐

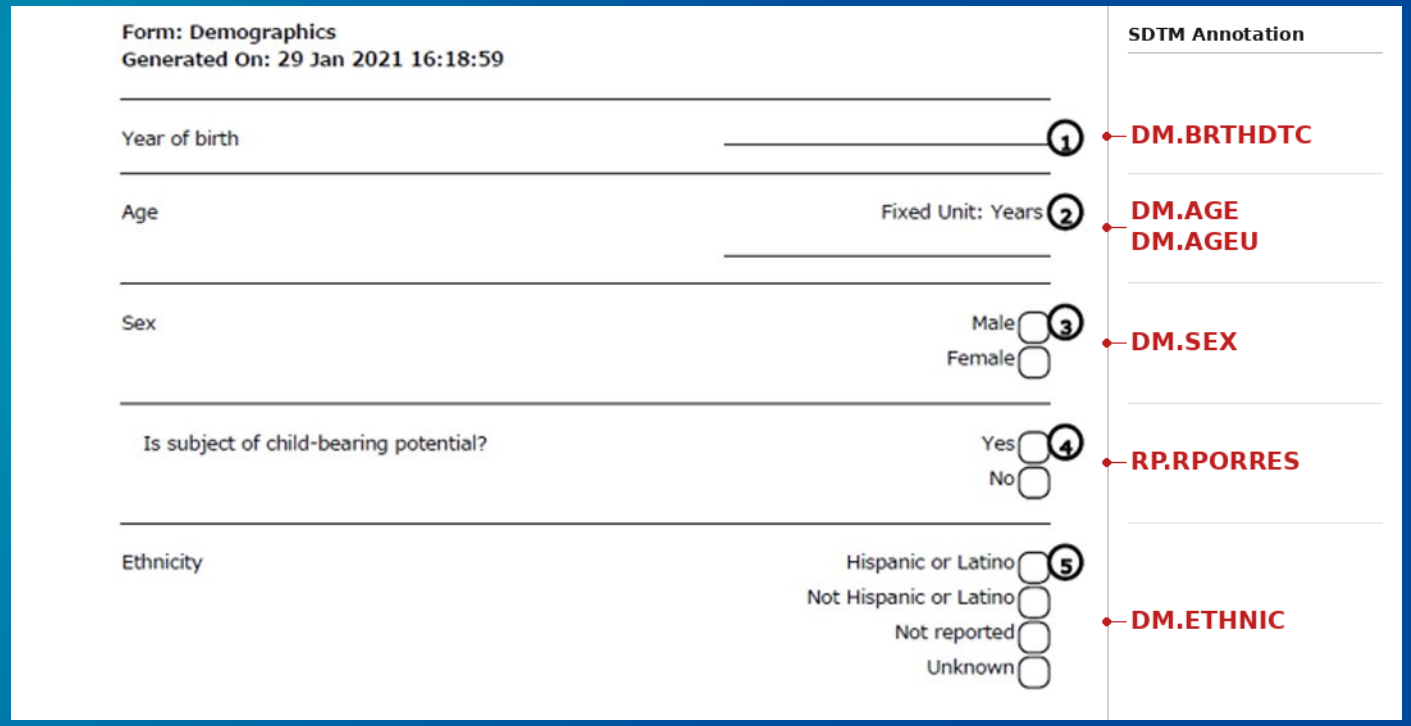
Is subject of child-bearing potential? — Yes ☐ (4) No ☐

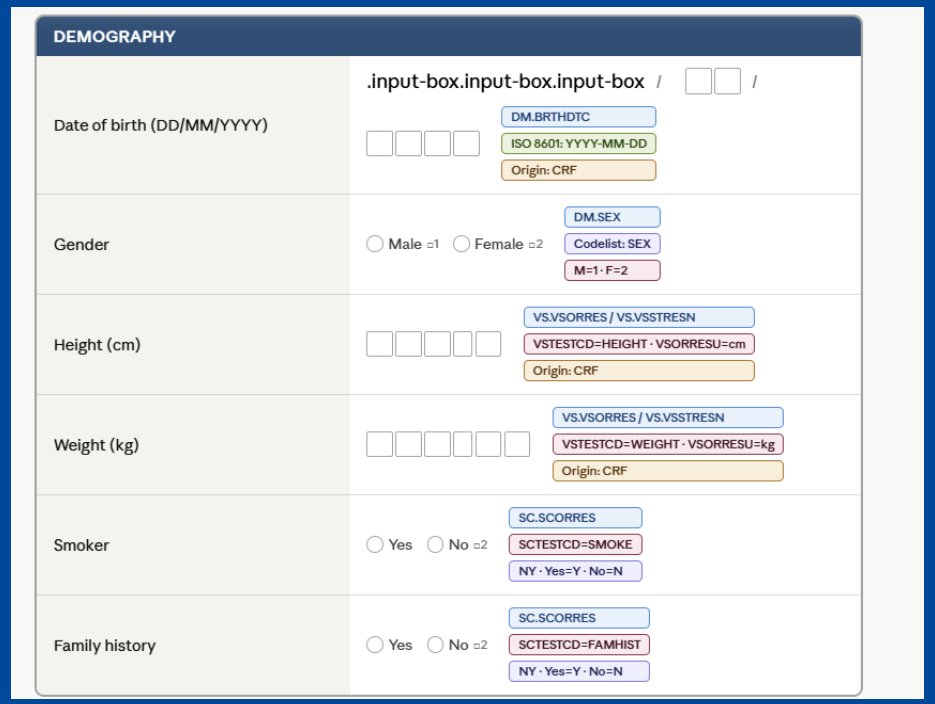
Ethnicity — Hispanic or Latino ☐ (5) Not Hispanic or Latino ☐ Not reported ☐ Unknown ☐

Annotated CRF Generation

Generated Output

connect·share·advance





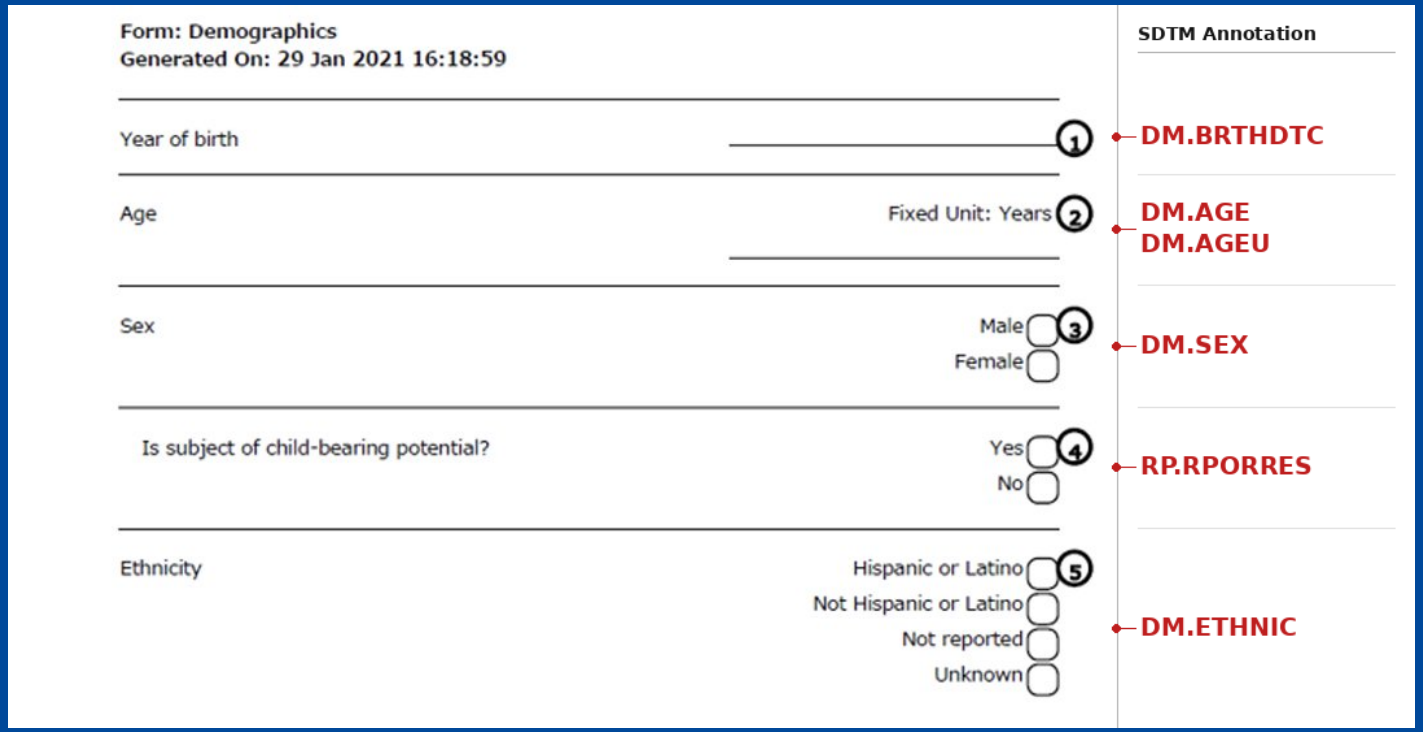
SDTM Specification Generation

LLM assisted domain mapping with controlled terminology resolution

connect·share·advance

Files Required

- Annotated CRF
- SDTM IG v3.4 DM domain section
- Protocol synopsis (treatment)
- CDISC CT package



Form: Demographics
Generated On: 29 Jan 2021 16:18:59

SDTM Annotation

Year of birth	①	DM.BRTHDTC
Age	Fixed Unit: Years ②	DM.AGE DM.AGEU
Sex	Male ③ Female	DM.SEX
Is subject of child-bearing potential?	Yes ④ No	RP.RPORRES
Ethnicity	Hispanic or Latino ⑤ Not Hispanic or Latino Not reported Unknown	DM.ETHNIC

SDTM Specification Generation

Generated Output

connect·share·advance

SDTM Specification — Demographics (DM) CRF Page

Dataset	Variable Name	Variable Label	Type	Controlled Terms / Format	Core	Origin	CDASH Variable (source)	CRF Field	Comment
DM	**BRTHDTC**	Date/Time of Birth	Char	ISO 8601 partial date (YYYY)	Perm	Collected	BRTHYR	Year of birth	Only birth year is collected on this CRF (privacy convention). SDTM BRTHDTC is date string, e.g. "1985".
DM	**AGE**	Age	Num	Integer; range per protocol eligibility	Exp	Collected	AGE	Age	Collected directly on the CRF as a nume
DM	**AGEU**	Age Units	Char	(AGEU) Controlled Terms: YEARS, MONTHS, WEEKS, DAYS, HOURS	Exp	Assigned	AGEU (fixed)	Age — "Fixed Unit: Years"	Unit is fixed at the form/header level as " collected per record; assign AGEU = "YE this form.
DM	**SEX**	Sex	Char	(SEX) Controlled Terms: M, F	Req	Collected	SEX	Sex	CRF offers two response options (Male/F to SEX codelist values M and F. No U/UN present on this form.
DM	**ETHNIC**	Ethnicity	Char	(ETHNIC) Controlled Terms: HISPANIC OR LATINO, NOT HISPANIC OR LATINO, NOT REPORTED, UNKNOWN	Perm	Collected	ETHNIC	Ethnicity	CRF offers four response options, mappi ETHNIC codelist.
RP / SUPPDM	**RPORRES (if RP) or QVAL (if SUPPDM)**	Childbearing Potential	Char	(NY) Controlled Terms: Y, N	Perm	Collected	RP.RPORRES, RPTESTCD=CBSPOT	Is subject of child-bearing potential?	Not a core SDTM DM variable. Per CDA modeled under Reproductive System Fin RPTESTCD="CBSPOT", RPTEST="Chi study does not implement RP, capture as to DM (SUPPDM: QNAM=CBSPOT, QL

Code Generation via AI Code Agent

Conversational code generation inside an IDE embedded AI assistant

connect·share·advance

Requirements

- ADaM Spec
- Prompt
- Sponsor specific macros or internal Utils existing macro

In adam_advs from safety data package, can you please mark MCRIT1FL eq "Y", if AVAL is greater than 90 for PARAMCD = "DIABP"

```r
library(safetyData)

advs <- safetyData::adam_advs

if (!"MCRIT1FL" %in% names(advs)) {
  advs$MCRIT1FL <- NA_character_
}

advs$MCRIT1FL[advs$PARAMCD == "DIABP" & advs$AVAL > 90] <- "Y"
```

Code Generation via AI Code Agent

Conversational code generation inside an IDE embedded AI assistant

connect·share·advance

Next Edit Suggestions

```
24  ASTDT <- case_when(
25          is.na(year) ~ as.Date(NA),
26          !has_month  ~ make_date(year, 1, 1),
27          has_month & !has_day & year < year(trtsdt) ~ make_date(year, month, 1),
28          has_month & !has_day & year >= year(trtsdt) ~ make_date(year, month, 1)
29    )
30
```

- Fix
- Explain
- Modify
- Review

```
ASTDT <- case_when(
          is.na(year) ~ as.Date(NA),
          !has_month  ~ make_date(year, 1, 1),
          has_month & !has_day & year < year(trtsdt) ~ make_date(year, month, 1),
          has_month & !has_day & year > year(trtsdt) ~ make_date(year, 1, 1),
          has_month & !has_day & year == year(trtsdt) ~ make_date(year, month, 1),
```

Rewrite

Modify

Review

Findings & Outcomes

What AI integration delivers across the clinical programming lifecycle

connect·share·advance

SDTM Mapping Efficiency

Dramatically reduced manual effort in mapping raw data to CDISC SDTM standards

ADaM Dataset Delivery

Faster and more consistent delivery of ADaM datasets

First Pass Validation

Higher rate of validation runs passing on the first attempt

Submission Cycle Speed

Weeks recovered per submission cycle

Key Findings

- AI tools reduce manual effort without displacing programmer expertise or judgment
- Standardized prompt templates allow AI models to generalize reliably across therapeutic areas
- Traceability improves as AI auto documents all data transformations in real time
- Submission packages become more consistent and regulatory review ready
- QC effort shifts from finding errors to validating AI generated outputs
- Programmer capacity is freed up for complex programming and standards governance

Submission Readiness Impact

Traceability **Significantly Improved**	Consistency **Markedly Higher**
Error Rate **Substantially Reduced**	Review Time **Notably Shorter**

Challenges and Limitations

connect·share·advance

01 Non-Standard Sponsor Fields

Custom CRF questions outside the CDASH IG vocabulary are not something the model can map confidently. These still require human judgment.

02 Unit and conversion fields

A measurement value and its units are often two separate fields. Agent annotates only value field and misses the unit field.

03 SAP Driven Derivation Logic (Fine tuning)

For partial dates or ongoing flags getting the LLM to the exact SAP intended logic usually takes a few iterative correction rounds not one prompt.

04 Cross domain relationships

The model maps individual domains reasonably well but does not reliably infer relationship between domains (RELREC)

05 Plausible but Incorrect Code

Generated SAS/R code often runs cleanly and looks right, but can contain subtle logic errors

06 Calibrating Trust

The real skill shift is knowing when to trust AI output. Over trusting and under using are both failure modes which we must learn.

Key Takeaways

- CDISC standards are the essential foundation for AI in clinical programming.
- AI can automate a major proportion of repetitive mapping and metadata tasks.
- Human oversight and regulatory validation remain non-negotiable.
- The clinical programmer's role is evolving and not disappearing.

connect·share·advance

THANK YOU!

phuse

The Global Healthcare Data Science Community

Contact Channels

office@phuse.global

phuse.global

Social Media

/company/phuse

/phusetube

/phuse_global

/phusebook