Paper DH02

Analysis Concepts Definition: Initial Perspective from the CDISC Working Group

Stuart Malcolm, Veramed, United Kingdom

ABSTRACT

In January 2025, a CDISC working group was established to define and model Analysis Concepts, marking a significant step toward enabling end-to-end automation in clinical research as part of the CDISC 360i initiative.

While the Unified Study Definition Model (USDM) standardizes key elements of a protocol a crucial gap remains: the lack of structured metadata for describing the statistical analysis.

Analysis Concepts aim to:

- Provide a standardized framework for expressing clinical questions and data derivations
- Support a machine-readable Statistical Analysis Plan (eSAP).
- Informing both data collection needs and processing logic for analyses.

This paper will share current perspectives on:

- The rationale and vision for Analysis Concepts within the CDISC standards ecosystem.
- Key components and linkages to USDM, ADaM, and ARS.
- Early models and examples developed by the working group.
- Future directions for standardizing analysis planning and enabling traceable, automated pipelines.

PROJECT STATUS

Work on the AC/DC Framework is work-in-progress. This paper is a description on the current state of development and will change over time. The intention of this paper is to engage with the wider community.

INTRODUCTION

The current state of clinical research suffers from a profound "automation gap." While the USDM provides a structured framework for study design, the logic for statistical analysis remains trapped in unstructured, static formats. Statistical Analysis Plans (SAPs) are primarily authored as Word or PDF documents—human-readable but machine-opaque. This creates a manual "jigsaw puzzle" where programming logic must be manually transcribed from narrative text into platform-specific code (R, SAS, etc.), introducing significant risk of transcription error and hindering reproducibility.

Machine-readability is a prerequisite for end-to-end automation. The strategic imperative for Analysis Concepts lies in shifting from narrative-based descriptions to metadata-driven specifications. The goal is to move beyond the "static document" paradigm toward an eSAP—a digital representation of analysis intent that is version-controlled, traceable, and executable. Only through the formalization of these concepts can the industry achieve the "So What?" of the digital transformation: a reduction in manual cycle times and the creation of a portable, standards-based analytical pipeline.

THE AC/DC FRAMEWORK

The AC/DC framework utilizes a five-layer architecture to enforce a strict Separation of Concerns. This decoupling ensures that each component can be updated or replaced without destabilizing the entire system.

Table 1: The AC/DC Five-Layer Architecture

Layer	Purpose	Derivation Example	Analysis Example
SmartPhrase	User-facing narrative text bound to concepts	"change from baseline in {param}"	"analyzed using ANCOVA"

Method	Pure mathematical/statistical operations	M.Subtraction, M.Aggregation	M.MMRM, M.CoxPH
Transformation	Central orchestrator (Method ↔ Concept)	T.ChangeFromBaseline	T.CFB_ANCOVA, T.TTE_CoxPH
Concept	Semantic Pivot Point (independent entities)	C.AnalysisValue, C.BaselineValue	C.ANCOVA, C.MMRM
Implementation	Physical binding to standards and code	ADaM: AVAL, BASE	R: lm(), SAS: PROC GLM

CORE ARCHITECTURAL PRINCIPLES

The framework is governed by six central principles (Source Section 1.3) that ensure architectural integrity:

1. Method Isolation: Methods define "pure" operations with named roles—they know nothing about semantic concepts or data standards.
2. Concept Independence: Concepts exist as standalone semantic reference entities; they are not "owned" by Transformations.
3. Transformation Orchestration: The Transformation is the central orchestrator, binding independent Concepts to specific Method roles.
4. Implementation-to-Concept Mapping: Physical variables map directly to Concepts, not to Transformations, facilitating standard-independence.
5. Concept-Referenced SmartPhrases: SmartPhrases provide user-facing text that binds directly to Concepts.
6. Contextual Constraints: Transformations declare which SmartPhrases they accept, ensuring contextually appropriate usage.

THE RECIPE ANALOGY: MAKING SOPHISTICATED METADATA ACCESSIBLE

To communicate this architectural complexity to a broader audience, we employ the "Recipe Analogy." This distinguishes between the intent of an analysis and its execution.

Table 2: Mapping Culinary Concepts to AC/DC Components

Cooking Concept	AC/DC Component	Functional Role
Recipe	Transformation	The template defining the pattern and required components.
Ingredient Category	Concept	The abstract requirement (e.g., "Oil" = C.AnalysisValue).
Cooking Instructions	Method	The pure technique (e.g., "Emulsify" = M.Subtraction).
Specific Product	Implementation	The physical item (e.g., "Bertolli Olive Oil" = ADaM.AVAL).

THE "PANTRY" AND IMPLEMENTATION INDEPENDENCE

The structural superiority of this model lies in how the "Recipe" (Transformation) interacts with the "Pantry" (Implementation Binding). A recipe for a vinaigrette requires "Oil"—it does not care if the pantry contains Italian Olive Oil (ADaM) or Canola Oil (OMOP). By standardizing the Expression of the choice rather than the Judgement (the statistician's expert selection of the model), we preserve scientific expertise while enabling technical automation. The Transformation doesn't care about the physical data standard as long as it finds a Concept to act as a semantic pivot point.

FORMAL SCHEMA DEFINITIONS AND REFERENCE LIBRARIES

The framework is operationalized through machine-readable schemas, leveraging the W3C Data Cube Vocabulary for structural alignment.

Methods Library: Pure Operations

Methods are standard-independent operation specifications. They utilize Wilkinson-Rogers notation for expressions and define a MethodRole schema (name, direction, dataType).

* Example Method: M.ANCOVA
  * Expression: response ~ covariate + group
  * Roles: response (in), covariate (in), group (in).
  * Parameters: ss_type, contrast, adjust, cov_structure.

Concepts Library: Semantic Pivot Points

Concepts provide the canonical information layer, mapped to W3C Data Cube classes:

* Measure (qb:MeasureProperty): The observed or computed value (e.g., C.AnalysisValue, C.ChangeValue).
* Dimension (qb:DimensionProperty): Variables that identify observations (e.g., C.TreatmentPlanned, C.Visit).
* Attribute (qb:AttributeProperty): Metadata qualifying observations (e.g., unit or status).

Transformations: The Binding Orchestrator

Transformations (T-level) bind specific Methods (M-level) to Concepts (C-level). For example, T.ChangeFromBaseline binds M.Subtraction to C.AnalysisValue (minuend) and C.BaselineValue (subtrahend).

STUDY INSTANTIATION AND EXECUTION PATHS

Study Instantiation is the process of generating study-specific configurations from reference libraries. This involves selecting a Transformation, assigning values to Method parameters (e.g., alpha=0.05, ddfm=KR), and defining analysis subsets via the Slice mechanism.

The Slice Mechanism (qb:Slice)

Subsetting is handled via the W3C qb:SliceKey and qb:Slice. A SliceKey defines which dimensions are fixed (e.g., C.PopulationFlag, C.Visit), while a Slice Instance fixes specific values (e.g., Population='ITT', Visit='Week 24') to create the precise context for analysis.

Execution Paths: Direct vs. Generated

1. Direct Execution (Runtime Engine): A metadata-driven engine interprets the Transformation Instance and executes the analysis dynamically, resolving Concepts to physical variables at runtime.
2. Resolved Code Generation: The system combines the metadata withCAMIS reference implementation "snippets" to produce platform-specific code (R/SAS). This is preferred for regulatory audit trails as it provides "human-readable" code that reflects traditional programming practices.

Implementation Binding Example

Concept	ADaM (Dataset.Variable)	SDTM (Dataset.Variable)
C.AnalysisValue	ADEFF.AVAL	VS.--STRESN
C.TreatmentPlanned	ADSL.TRT01P	DM.ARM

THE E-SAP AND SMARTPHRASE RESOLUTION

SmartPhrases enable dual-output capability: machine-executable specifications and human-readable narrative text. By referencing Concepts, SmartPhrases ensure absolute synchronization between the digital SAP and the final clinical study report.

SmartPhrase Resolution Example:

* Template: {SP_CFB_ENDPOINT} {SP_TIMEPOINT} {SP_METHOD_ANCOVA}
* Resolution: "Change from baseline in HbA1c at Week 24 will be analyzed using ANCOVA." This approach eliminates transcription errors; if the metadata for C.Visit changes, the narrative and the execution logic update simultaneously.

ECOSYSTEM INTEGRATION: USDM, ADAM, AND ARS

Analysis Concepts bridge the "metadata gap" across the CDISC standards lifecycle:

1. Protocol (USDM): Defines objectives and estimands.
2. Data Collection/Tabulation (CDASH/SDTM): Standardizes raw inputs.
3. Analysis (ADaM + AC/DC): AC/DC defines how data is derived and analyzed.
4. Results (ARS): The Analysis Results Standard captures the final outputs.

The AC/DC framework acts as the connective tissue, ensuring that the transition from Analysis Data (ADaM) to Analysis Results (ARS) is traceable and automated.

STAKEHOLDER IMPACT: USER STORIES FOR THE DIGITAL ERA

Stakeholder	Value Driver	Evaluated Impact
Study Statistician	Accelerated Start-Up	Rapidly drafts digital SAPs by selecting from libraries; auto-identifies required variables.
Standards Expert	Metadata Automation	Build structured repositories that serve as authoritative inputs for downstream TFL generators.
Statistical Programmer	Automated Analysis	Implements "topological execution ordering" to run standard analyses without manual logic recreation.

CONCLUSION AND FUTURE DIRECTIONS

The AC/DC framework transforms clinical research from a fragmented "jigsaw puzzle" into a modular, traceable "LEGO-like" ecosystem. The modularity of the framework allows for a Dependency Resolution Algorithm (Source Section 10.5): when an Analysis Transformation (AC) is selected, the system automatically identifies and orders the necessary Derivation Transformations (DCs) by matching AC inputs to DC outputs.

The next phase of CDISC 360i will finalize the YAML-based reference libraries, including:

* acdc_schema.yaml: The LinkML-style core schema.
* acdc_methods_library.yaml: The repository of pure operations.
* acdc_concepts_library.yaml: The semantic reference library.

By standardizing the expression of analysis through these structured digital artifacts, the industry achieves the transparency, reproducibility, and efficiency required for the future of clinical data science.

REFERENCES

- Analysis Concepts Definition: Initial Perspectives from the CDISC Working Group. Presented by Bhavin Busa (Clymb Clinical) and Brian Harris (AstraZeneca) at the 2025 CDISC US Interchange in Nashville, 13–17 October 2025.
- 360i Roundtable Discussion: Understanding, Developing, and Leveraging Analysis Concepts. Facilitated by Peter Van Reusel and recorded by Edwin Van Stein.
- AC/DC Framework (Analysis Concepts / Derivation Concepts) Model Specification, Draft v0.5. A formal technical document defining the machine-readable schemas for the five-layer architecture.
- The AC/DC Framework: A Recipe Analogy. An educational guide explaining the separation of concerns between methods, concepts, and implementations using a culinary metaphor.
- Analysis Concepts: Initial Perspectives from the CDISC Working Group (Paper DS05c). Presented by Karl Wallendszus (University of Oxford) and Edwin van Stein (CDISC) at PHUSE EU Connect, 18 November 2025.

CONTACT INFORMATION

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

Author Name: Stuart Malcolm
Company: Veramed
Email: stuart.malcolm@veramed.com
Website: www.veramed.com