Paper DS12

Using Standardization and Automation to Streamline Generation of SDTM/ADaM Codelists

Eric Du, Gilead Sciences, Foster City, USA

ABSTRACT

Generating SDTM and ADaM Controlled Terminology (CT) for Define-XML is currently a semi-manual process at Gilead. Due to the sheer amount of manual interventions, the process is prone to human errors, and may cannibalize manhours from more productive activities. We aim to establish a new process to increase efficiency by standardization and automation, relying on a centrally managed Standard CT. This Standard CT file will serve as a lookup table and be compiled from multiple standard sources, while also allowing for regular updates as NCI CT or Data Collection standards evolve. We will also present a new automation tool to generate study-specific SDTM/ADaM CT that can be manually customized, if necessary for a given study, and then ingested by Gilead's modified Define-XML v2.0 generation tool.

INTRODUCTION

Define-XML is a metadata file that describes datasets, variables, controlled terminology, computational methods, and other essential metadata that are used in an electronic submission and is required by U.S. Food and Drug Administration (FDA) [1] and Japanese Pharmaceutical and Medical Devices Agency (PMDA) [2]. In FDA CDER's binding guidance, it is stipulated that "a properly functioning *define.xml* file is an important part of the submission of standardized electronic datasets and should not be considered optional" [3].

Controlled Terminology (CT) is a required element in Define-XML that explains all allowable values across all variables that have finite sets of allowable values in the study [4]. CT is represented in the form of "codelists" that show the specific codes used in study and their decoded values. CT helps sponsors standardize naming conventions and values for CDISC-compliant datasets, and consequently improves data quality in the eyes of regulatory reviewers. The actual generation of CT for a study, however, is a time-consuming yet unimaginative task. The main challenge is that the CT is composed of information from different, unrelated sources, and controlling versions across these sources is a daunting effort. At Gilead, CT is often based on a mix of National Cancer Institute (NCI) SDTM terms, Gilead Architect Loader Specification (ALS), Gilead Standard Electronic Data Transfer Specification (eDTS) across vendors, and vendor-provided extensible codelists (***Figure 1***). Standardizing these CT sources is a crucial step before Sponsor can automate the CT generation process to improve upon operational efficiencies.

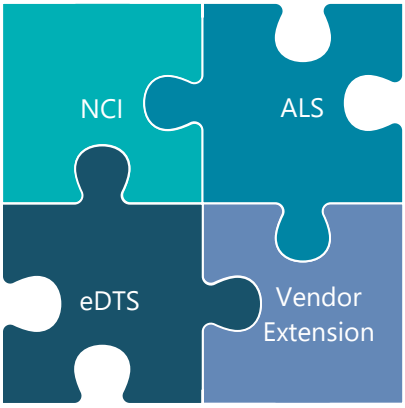

Figure 1 Most commonly seen CT sources at Gilead

CURRENT CHALLENGE AT GILEAD

At Gilead, Statistical Programmers (SP) use SDTM or ADaM Mapping Specifications (specs) to generate SDTM/ADaM datasets and SDTM/ADaM Define-XML. The specs is a standardized, Excel-based file that records standard and study-specific metadata about SDTM or ADaM datasets and how programmers intend to treat them.

The SDTM/ADaM specs are divided into different Excel worksheets containing relevant metadata. The "Codelist" sheet comes pre-populated with standard NCI/SDTM terms and standard Gilead extensions or terms. The primary programmer is responsible for manually checking and updating the spreadsheet with study-specific codelists and extensions to standard codelists. The information in this spreadsheet, along with the entire mapping specification, is then read by a Gilead automation tool to generate a compliant Define-XML file that can be used for regulatory submissions.

Although different Therapeutic Areas (TAs) may have had different ways of generating study CT, all the processes

are largely manual in nature and can be time-consuming and are error-prone. Our strategy is to standardize and streamline this process by implementing standard input source files and folder structures, as well as creating an automation utility (hereafter 'The Utility'). This will serve to not only reduce SP workload, but also to leverage our existing Define-XML generation tool to establish a seamless end-to-end process.

FUTURE PROCESS AT GILEAD

In the future, Gilead will centrally manage a Standard SDTM Controlled Terminology that contains all standardized codelist sources that are commonly used by Gilead study teams (i.e. NCI, ALS, eDTS, and Vendor CT). The Standard SDTM CT will be versioned periodically and will be the starting point for generation of the Study SDTM CT. Subsequently, Study ADaM CT will be generated based on the Study SDTM CT (***Figure 2***).

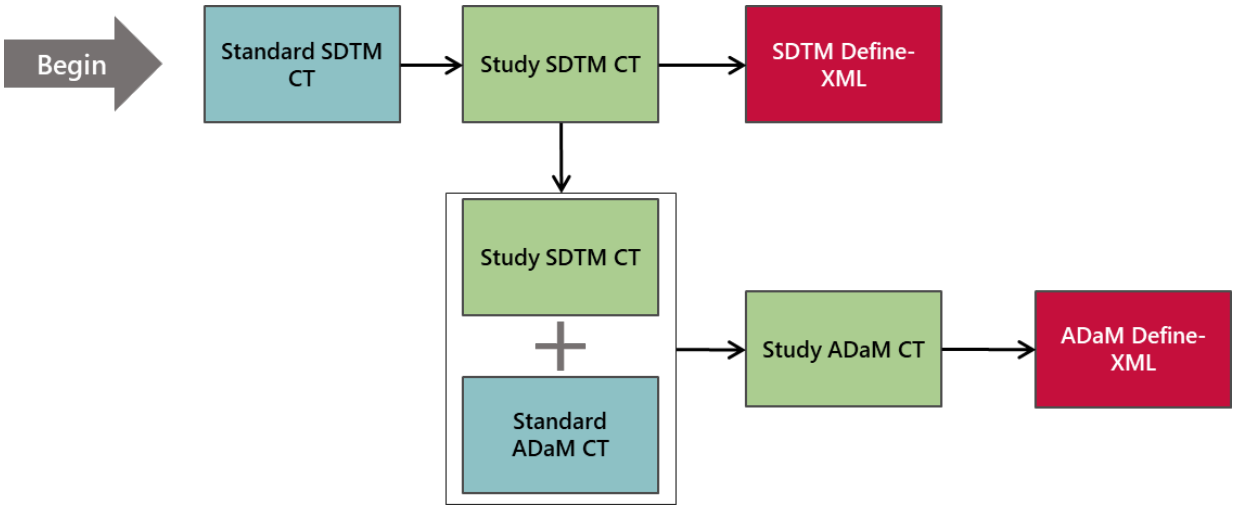

Figure 2 High-level view of future SDTM and ADaM CT Processes. ADaM will always begin after SDTM has been completed.

SDTM CT PROCESS

The goal of the SDTM CT process is to generate an exhaustive list of SDTM codelists from the Standard SDTM CT, while accounting for study-specific values by cross-referencing study ALS and study eDTS. To begin the process, programmers would have codelist source files stored in their designated folders as part of Gilead's standard biometric programming environment. Programmers would then execute the Utility, which would automatically run a series of cross-checks and worksheet entries. The end products of running through the SDTM CT process will be an Excel file of SDTM codelists that can then be used by a tool to generate SDTM Define-XML.

The Utility's automated process uses a series of logic and cross comparisons between data sources to discern the codelists that are used by the study from standard ones, a task that is currently performed manually and inconsistently.

LOGIC BEHIND THE UTILITY FOR SDTM

The Utility begins by comparing the "Codelist Name" column in Standard CT file against its analog, the "Data Dictionary Name" column in Gilead SDTM Study ALS (***Figure 3***). If the codelist names do not match, the Utility will identify if the discrepancy is a result of the codelist being used in study (as represented in ALS) but not part of standard, or vice versa.

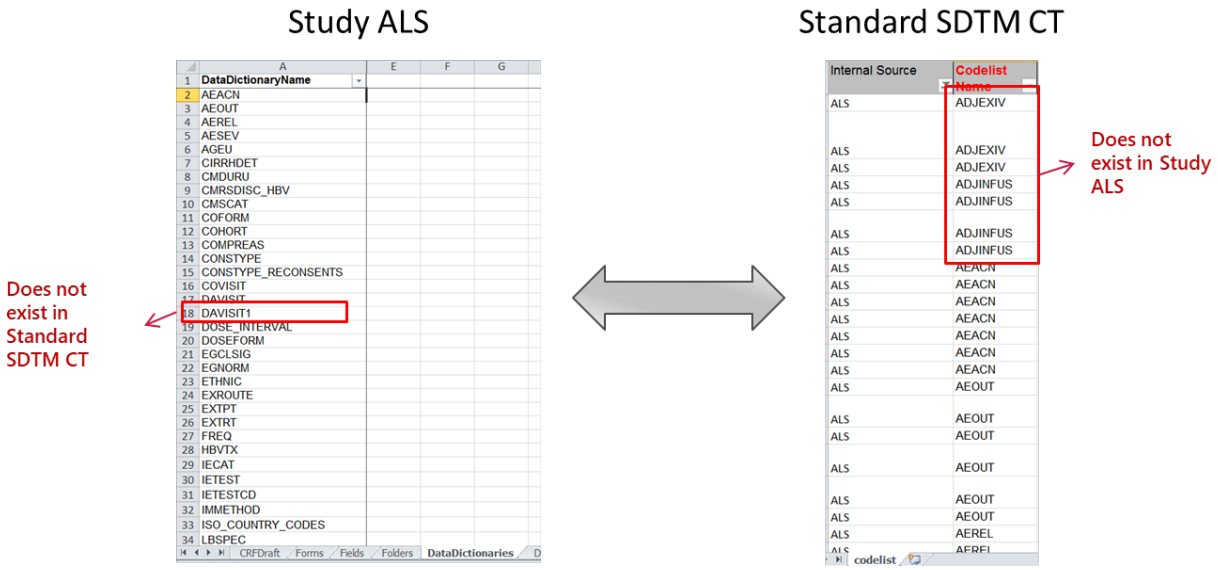

Figure 3 The Utility automatically compares codelist names in Study ALS and Standard SDTM CT

For codelists that are used in the study but not part of Standard CT, the Utility will document these study-specific codelists and they will appear on the Define-XML. As for the standard codelists that are not used in this study, they will be deactivated and therefore not included in Define-XML. In the case that codelist names do match, the Utility will then compare the decoded codelist values, and the same matching and documentation process described above applies (***Figure 4***).

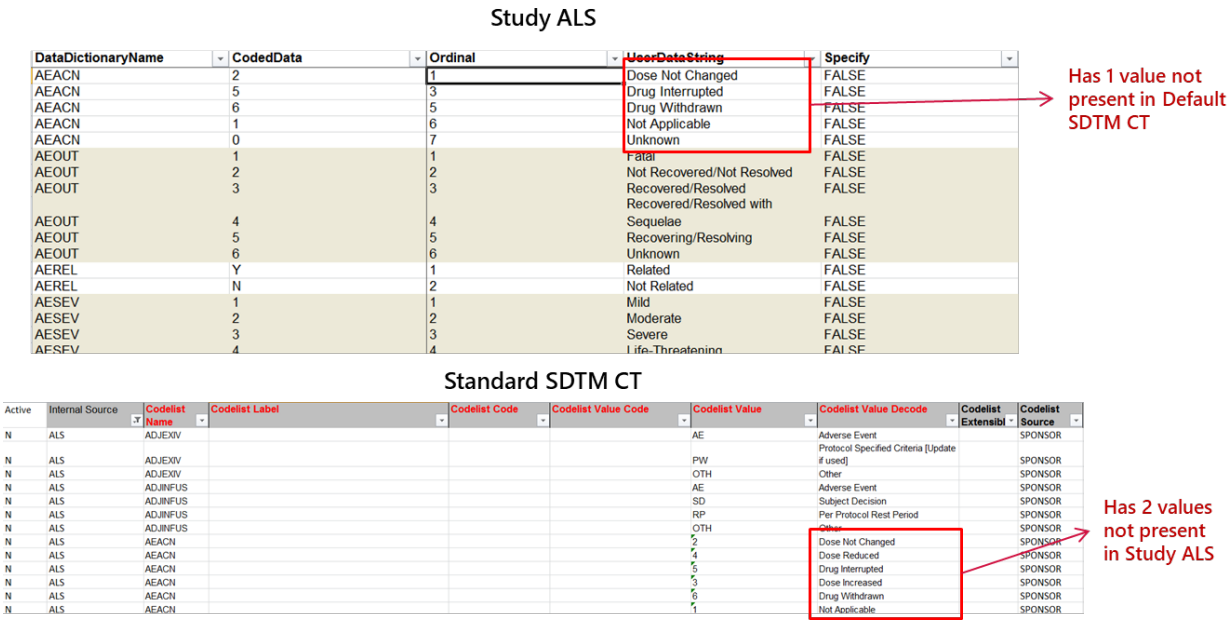

Figure 4 The Utility automatically compares codelist decode values if the codelist names match

In summary (***Figure 5***), these cross checks would essentially yield three types of codelists: 1) In Standard and used by study, 2) In Standard, but not used by study, and 3) Not in Standard, but used by study. At the end, the Utility would activate #1, deactivate #2, and add & activate #3.

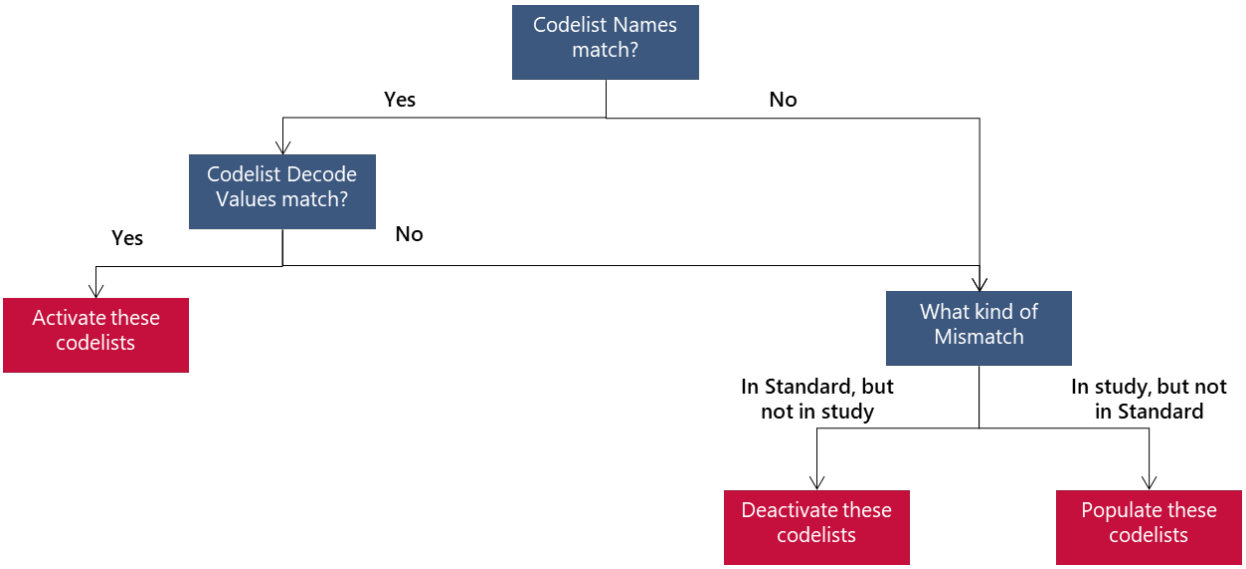

Figure 5 Decision diagram for the automated portion of SDTM CT process

ADAM CODELIST PROCESS

The ADaM CT process should begin only after Study SDTM CT have been completed, to ensure that codelists are consistently captured and duplicative efforts are not needed to realign SDTM and ADaM. The Utility will then complete the process by cross-checking other study specifications to account for any additions or changes for ADaM codelists. The end products of running this process will be ADaM codelists that can then be used by the Define-XML generation tool.

LOGIC BEHIND ADAM AUTOMATION TOOL

The Utility begins by checking if SDTM codelists have been previously generated by the Utility. This measure ensures that the ADaM codelists hereafter generated will be built upon complete, up-to-date, and standardized SDTM codelists. A warning would be given to users and the tool halts if the Study SDTM CT file is nonexistent in the designated folder This would reveal that SDTM programmers did not run the requisite Utility prior to the ADaM process.

The Utility creates an all-inclusive baseline file by merging the existing, up-to-date Study SDTM CT file with the Standard NCI ADaM CT. The file will then be updated with study-specific information in the subsequent automation process. The Utility would compare this combined file against the study ADaM Mapping Specification to bring in user-modified metadata about study datasets.

Compared to the SDTM process, the ADaM process adds another layer of complexity because some variables are derived from SDTM predecessors that have their own codelists. As a result, the initial check the Utility would perform is to see if a given ADaM variable has any SDTM or ADaM predecessor (**Figure 6**). If there is no predecessor, then the Utility would assess whether the codelists used by ADaM Mapping Spec are already captured in the combined

CT, and will make decisions to update the CT accordingly. If the ADaM variable has a predecessor, then the Utility would locate the codelists used by the predecessor and update the CT accordingly.

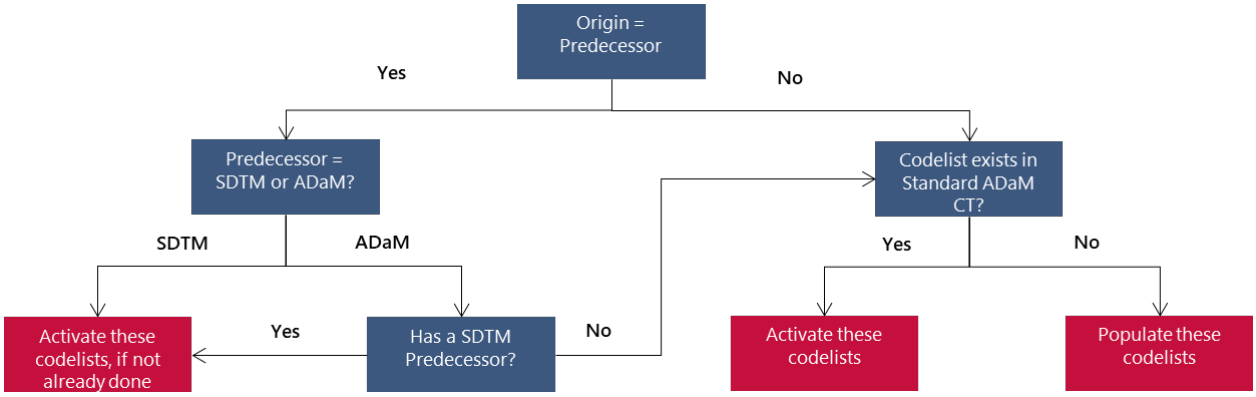

Figure 6 Decision diagram for the automated portion of ADaM CT process

In addition to the ADaM Mapping Spec, the Utility would also capture PARAMCD values in ADLB, ADVS, and ADEG datasets, and populate them in the Study ADaM CT as study-specific codelists (***Figure 7***).

Active	Internal Source	SDTM reference	Codelist Name	Codelist Label	Codelist Code	Codelist Value Code	Codelist Value	Codelist Value Decode	Codelist Extensible	Codelist Source
Y	ADLB		ADLB_PARA	Urine Glucose			TRACE			SPONSOR
Y	ADLB		ADLB_PARA	Urine Protein			1			SPONSOR
Y	ADLB		ADLB_PARA	Urine Protein			NEGATIVE			SPONSOR
Y	ADLB		ADLB_PARA	Urine Protein			TRACE			SPONSOR
Y	ADLB		ADLB_PARA	Urobilinogen			1			SPONSOR
Y	ADLB		ADLB_PARA	Urobilinogen			NORMAL			SPONSOR
Y	ADEG		ADEG_PARA	Interpretation (Clinical Significance)			Abnormal (NCS)			SPONSOR
Y	ADEG		ADEG_PARA	Interpretation (Clinical Significance)			Normal			SPONSOR
Y	ADEG		ADEG_PARA	Description of Abnormality			LEFT ANTERIOR FASCICULAR BLOCK			SPONSOR
Y	ADEG		ADEG_PARA	Description of Abnormality			LEFT AXIS DEVIATION			SPONSOR
Y	ADEG		ADEG_PARA	Description of Abnormality			LONG QT INTERVAL			SPONSOR

Figure 7 The Utility also checks value-level datasets (e.g., ADLB, ADVS, ADEG) to identify codelists

ADDITIONAL FUNCTIONALITIES FOR THE TOOLS

MANUAL INTERVENTION ENABLED

Both SDTM and ADaM processes results in codelists in Excel format and serve as a manual checkpoint before the codelists move on to the next step of automated generation of Define-XML. Programmers are offered the opportunity to add/delete/change information in the Excel file, as needed, for their study. After changes are made and saved, the Excel files will then serve as the input source to another Gilead automation tool that is made for generating Define-XML.

TOOL RE-RUN ENABLED

We expect to see use cases where programmers may need to re-run the Utility, perhaps due to updates in input source files (e.g. Study ALS) that may have downstream effect on the codelists. When the re-run command is given, the current version would be archived, and the entire process would be executed from the beginning. For the changes that are not made in the automation process, but rather at the manual checkpoint, programmers are given the function to flag these codelists (***Figure 8***) so that these changes are carried over to the new version after re-run.

Active	Internal Source	SDTM Reference	Codelist Name	Codelist Label	Codelist Code	Codelist Value Code	Codelist Value	Codelist Value Decode	Codelist Extensible	Codelist Source
Y	ALS		RACESCAT				CNM	Chinese (Mainland)		SPONSOR
Y	ALS		RACESCAT				CNO	Chinese (Other)		SPONSOR
Y	ALS		StagingMethod				M	Metavir		SPONSOR
Y	CUSTOM	VS.VSTESTCD	VSTESTCD_X	Vital Signs Test Code			VSALL	Vital Signs	Y	SPONSOR
Y	CUSTOM	VS.VSTEST	VSTEST_X	Vital Signs Test Name			Vital Signs	Vital Signs	Y	SPONSOR

Figure 8 In the manual portion of the SDTM/ADaM CT process, programmers can flag manually-inserted codelists as "CUSTOM", so that they will be carried over even after re-run

VERSION CONTROL ENABLED

Managing versions of Controlled Terminology is a major undertaking for both study team programmers and the data standards governance team. Traditionally, Gilead keeps track of versions manually and the practice is largely based on institutional knowledge. In the future standard process that leverages the automation Utility, the programmer would specify the NCI SDTM or ADaM version in their SDTM/ADaM Mapping spec, and the Utility would match it with the Standard CT of the same version. This feature, along with the entire automation process and re-run capability, would save programmers significant time that would have been wasted in aligning different versions of CT and updating changes manually.

CONCLUSION

Generating Define-XML is a required process for FDA and PMDA submissions. However, completing the codelist section consumes significant time that can be otherwise invested into other creative work. Currently, this process is largely manual and is subject to programmers' or study teams' own methodologies, which often do not remain institutional once they leave the department. The standard process that Gilead Biometrics is developing will leverage automation to improve efficiency and reduce errors, while also maintaining traceability. Our next steps will be to begin the development of the tool and pilot test the process.

REFERENCES

1. U.S. Food and Drug Administration. (2019). Study Technical Conformance Guide v4.3. March 2019.
2. Japan Pharmaceuticals and Medical Devices Agency. (2015). Technical Conformance Guide on Electronic Study Data Submissions. Notification No. 0427001.
3. Center for Drug Evaluation and Research. (2011). CDER Common Data Standards Issues Document Version 1.1. December 2011
4. CDISC. (2013). CDISC Define-XML Specification Version 2.0.

ACKNOWLEDGMENTS

I would like to acknowledge and thank the great data standards team at Gilead for their continued dedication and hard work in getting the standards published and governed, as well as the many volunteers who contribute to standards development concurrently with study work. In addition, thanks to Dhananjay Chhatre and Kevin Miller for helping to review this paper, and Greg Campbell and Pei Sern for their oversight of the SP technology and data science teams.

CONTACT INFORMATION

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

Eric Du
Gilead Sciences
333 Lakeside Drive
Foster City, CA 94404
Work Phone: 650-372-7048
Email: eric.du@gilead.com
Web: www.gilead.com