Paper RS08

Creating the Case Report Tabulation (CRT) for an NDA submission at the absolute last moment – NOT

Christine Connolly, Kevin King, Amanda Tweed and Steve Wong,
Millennium Pharmaceuticals, Cambridge Massachusetts

ABSTRACT

If your company is anything like ours, you've waited until the last possible moment to start assembling and building your CRT. Often times it is quite a flurry of activity filled with late nights spent meeting technical challenges, gathering information, and preparing the documents in order to meet the schedule specified by your regulatory or publishing department. However, by waiting until the last minute, you have ensured that you have caught all the changes that happened during ad-hoc analysis or while fixing bugs or algorithms, that didn't get recognized beforehand. The pain is worth it! But is it necessary?

At our company, the CRT was rarely on the statistical programmer's radar during the study. We were often busy writing programs supporting data cleaning, mapping data to company standards, reconciling vendor transfers, providing analyses to data safety monitoring boards (DSMBs), supporting other departments such as Clinical Pharmacology as well as providing standard tables, listings and figures (TLFs) for the clinical study report (CSR).

We began to question the submission process. **Who** can benefit if we start earlier? **What** are the components of the CRT that we need to create? **When** is the right time for us to start? **Where** can we find hints or tips to do this? **How** do we build the CRT? And finally, **Why** should we develop a new process for assembling CRTs? Answers to these and other questions will be discussed in the following paper.

INTRODUCTION

The Case Report Tabulation (CRT) is the collection of the annotated case report form (CRF), SAS® datasets, metadata, and source programs that comprise a portion of the NDA package submitted to the FDA. The FDA uses it when reviewing submissions. Review starts with the Define document which contains metadata describing the datasets, variables, and values. It is all tied together using internal and external hyperlinks, bookmarks, and destinations to make it easily navigable.

For our past three NDA submissions, we had one in-house CRT expert whom we relied on to generate and assemble all the relevant documents that comprised the CRT. This person was rarely the primary programmer on the studies being submitted; instead, he was a contractor who was often given the least desirable tasks in our department. His legacy of knowledge from working on the first CRT was carried forward to the other two submissions. On average, he spent 3 months on a submission, with each submission consisting of a main study, a few supporting studies and an integrated summary of safety (ISS). His work started soon after the main study analysis was complete. Not having worked intimately on the project until this point, he would have to start from scratch to learn each study--reconstructing specific algorithms and other calculated variable definitions from e-mails, spreadsheets, and other assorted documents. He also required frequent meetings with the primary programmer for each study to pick their brain on what to document, why a particular algorithm, whether variable definitions for the addendum were correct, and numerous other study details.

One of the reasons he was chosen to do this work was that he had a technical expertise that was not used on a day-to-day basis. We had confidence in his technical prowess and were uncertain how we would replace these skills if he were to move on. After he left the company, one of the primary tasks of interest for us was to automate the technical aspects of the CRT assembly and create a new process that any primary programmer could do.

A lull in our submission schedule coincided with the new requirements of the electronic Common Technical Document (eCTD) submission format to set the stage for us to re-examine how we assemble CRTs and to consider what we can do differently The authors were tasked with developing a process to streamline and optimize the assembly of the CRT documents for inclusion in an eCTD submission. Our resulting process automates the technical aspects of the CRT build and presents a series of steps that any primary programmer can follow and use. This paper documents the results of the project and outlines the process that we developed.

WHO

Who should start this process?

At our company, a statistical programmer owns the CRT process. In the past, this has been a programmer who had not worked directly on the study, since the primary study programmer was busy on other aspects of the submission. Using any documentation, formal or informal, that had already been prepared, the CRT programmer would piece the information together.

Under this model, the process of assembling the CRT would typically begin at the end of the study. There would be a lot of questions back and forth between the two programmers, the statistician, and others at a time when there were already numerous tasks impinging on people's time. Performing the task at this juncture of the study was stressful and had the potential to introduce errors into the CRT.

Who could benefit if we start earlier?

Our new model pushes the CRT assembly to an earlier point in the submission cycle. The task is given to the primary programmer and performed during study conduct. This results in documentation happening in real-time, rather than having it occur months or years after the fact when relying on a program header or e-mail message to document a complex algorithm is the only option.

Using the new process, the CRT should be ready-to-go when the time comes for a submission. This will lead to a higher quality product for the key recipients, both internally (Regulatory) and externally (FDA). In addition, the quality of life for the statistical programmers during a submission should greatly improve as a key deliverable will be nearing completion at a much earlier point in the cycle.

Some important fringe benefits are also realized. First, programmers responsible for QC during regular study conduct have one place to look for thorough documentation. If parallel programming is part of QC, this is an invaluable resource. Second, new programmers or statisticians coming to work on the study will have a one-stop shop to familiarize themselves with the data. Their learning curve will be shortened substantially. Finally, using this process throughout an organization results in a consistent method of documentation for all clinical studies, whether or not they will ever be submitted.

WHAT

What is a CRT?

The CRT is essentially a collection of data and documentation for a study. It contains features such as bookmarks and links to allow reviewers to easily navigate the submission. For consistency, there are guidelines from FDA[1] and CDISC defining the components though the guidelines are limited in scope.

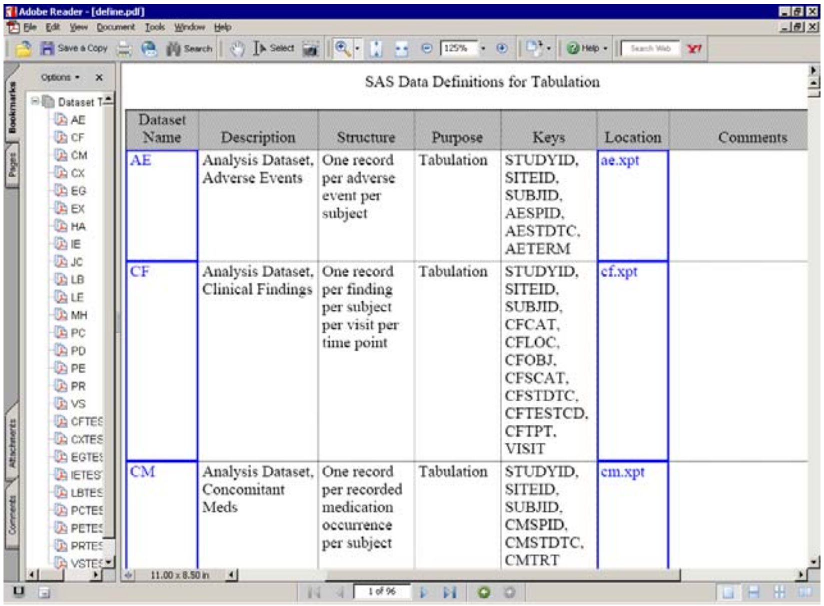
What are the components of the CRT that we need to create?

- Define Document

The Define document is the central, top-level document that allows easy navigation to the various components. The document contains metadata on datasets and variables. It contains bookmarks and links to each dataset. Currently, the expected format is a PDF document, but eventually the requirement will be XML.

Figure 1 contains an example of dataset–level metadata. Metadata describing the dataset contents, structure, and uniqueness keys are included. Clicking on the dataset name brings the viewer to variable-level metadata, while clicking on the location opens the actual dataset.

Figure 1. Dataset-Level Metadata Example

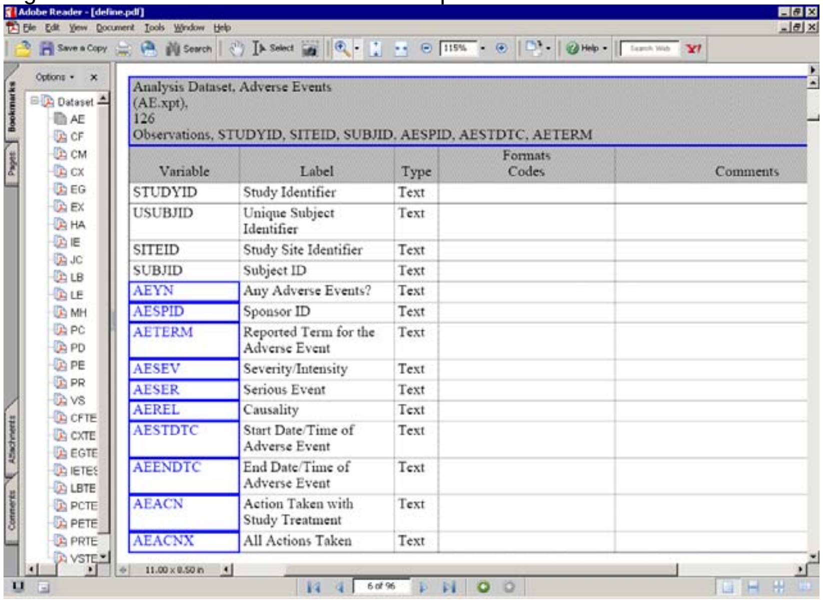


SAS Data Definitions for Tabulation

Dataset Name	Description	Structure	Purpose	Keys	Location	Comments
AE	Analysis Dataset, Adverse Events	One record per adverse event per subject	Tabulation	STUDYID, SITEID, SUBJID, AESPID, AESTDTC, AETERM	ae.xpt	
CF	Analysis Dataset, Clinical Findings	One record per finding per subject per visit per time point	Tabulation	STUDYID, SITEID, SUBJID, CFCAT, CFLOC, CFOBJ, CFSCAT, CFSTDTC, CFTESTCD, CFTPT, VISIT	cf.xpt	
CM	Analysis Dataset, Concomitant Meds	One record per recorded medication occurrence per subject	Tabulation	STUDYID, SITEID, SUBJID, CMSPID, CMSTDTC, CMTRT	cm.xpt	

Figure 2 contains an example of variable-level metadata for the Adverse Events (AE) tabulations dataset. The dataset label and number of observations appear in the header. Each variable in the dataset is listed, along with associated metadata such as label, type, and controlled terminology. Variables that were collected on a CRF contain links to the appropriate page on an annotated CRF casebook that is included in the CRT (see blankcrf below). There is also a column for additional comments.

Figure 2. Variable-Level Metadata Example



Analysis Dataset, Adverse Events
(AE.xpt),
126
Observations, STUDYID, SITEID, SUBJID, AESPID, AESTDTC, AETERM

Variable	Label	Type	Formats Codes	Comments
STUDYID	Study Identifier	Text		
USUBJID	Unique Subject Identifier	Text		
SITEID	Study Site Identifier	Text		
SUBJID	Subject ID	Text		
AEYN	Any Adverse Events?	Text		
AESPID	Sponsor ID	Text		
AETERM	Reported Term for the Adverse Event	Text		
AESEV	Severity/Intensity	Text		
AESER	Serious Event	Text		
AEREL	Causality	Text		
AESTDTC	Start Date/Time of Adverse Event	Text		
AEENDTC	End Date/Time of Adverse Event	Text		
AEACN	Action Taken with Study Treatment	Text		
AEACNX	All Actions Taken	Text		

- SAS® Transport files

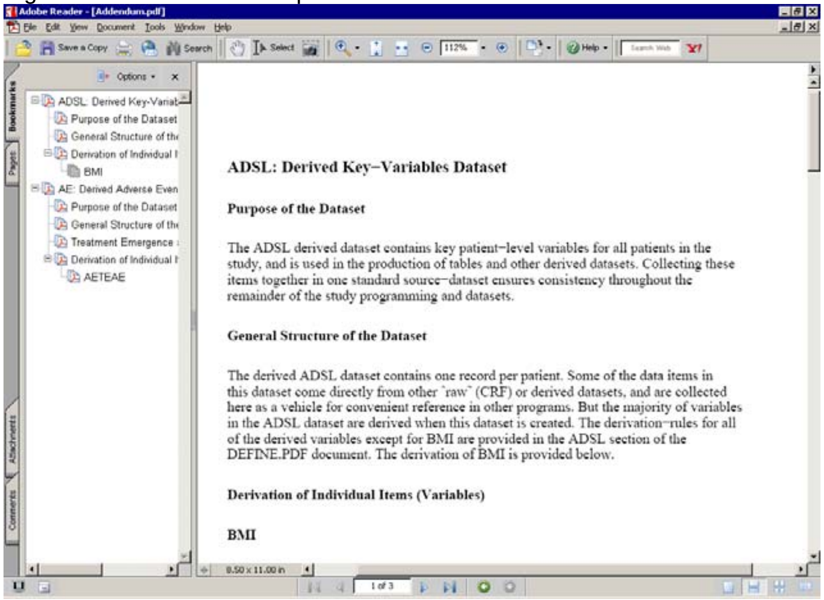
The CRT contains data for the study stored in SAS® transport files. Currently, the FDA requires data to be in SAS® XPORT Transport Format. This is an open format, meaning it can be translated to and from the XPORT transport format to other commonly used formats without the use of programs from any specific vendor[2].

Each transport file contains one dataset. There are two types of datasets: tabulations --- essentially the raw data -- and analysis --- data structured so that it is analysis-ready. In the CDISC model, tabulations datasets are expected to use the Study Tabulation Data Model (SDTM). For analysis datasets, an analogous structure, the Analysis Data Model (ADaM) has been defined. Currently, ADaM is less established than SDTM and we have chosen not to implement it as of yet.

- Addendum

As noted above, the variable-level metadata contains a space reserved for additional comments. In some cases, there may be a long explanation that is preferable to store outside of this table. The addendum document is the place to do this. It contains descriptions of complex algorithms and detailed explanations that would be too long for the metadata table. There is no concrete definition for “too long”, but anything longer than a half-page is our suggested cutoff. Figure 3 provides an example of a description of a key, subject-level analysis dataset called ADSL, and of how a specific variable, Body Mass Index (BMI), was derived.

Figure 3. Addendum Example



ADSL: Derived Key−Variables Dataset

Purpose of the Dataset

The ADSL derived dataset contains key patient−level variables for all patients in the study, and is used in the production of tables and other derived datasets. Collecting these items together in one standard source−dataset ensures consistency throughout the remainder of the study programming and datasets.

General Structure of the Dataset

The derived ADSL dataset contains one record per patient. Some of the data items in this dataset come directly from other `raw` (CRF) or derived datasets, and are collected here as a vehicle for convenient reference in other programs. But the majority of variables in the ADSL dataset are derived when this dataset is created. The derivation−rules for all of the derived variables except for BMI are provided in the ADSL section of the DEFINE.PDF document. The derivation of BMI is provided below.

Derivation of Individual Items (Variables)

BMI

- Blankcrf

The blankcrf is a PDF file containing the full set of CRFs, annotated with the variable names for each CRF item that is included in the tabulation datasets[1]. As mentioned above, CRF variables in the variable-level metadata of the Define document will link to the appropriate page in this file.

- Programs

The programs are the actual code that was written to generate the analysis datasets.

WHEN

When is the right time for us to start?

The right time to start is contingent upon your company’s processes and the individual study. Generally, we recommend that work on the CRT begin as early as possible. However, study desgn, including number of subjects and duration, will be key to developing a realistic timeframe. Ideally, the process should be started early enough to make it more time efficient, but not so early it becomes cumbersome. To help facilitate this, we need to consider the stability of the information we are using to build the CRT to determine our starting time point.

One factor to consider is when the Statistical Analysis Plan (SAP) will be finalized. The SAP will provide the bulk of the definitions for analysis variables to be incorporated into the Addendum file. A second factor to consider is the stability of the datasets to be included in the CRT, as their structure will be documented in the Define file. Consider: How much do you expect the SAP and the structure of the datasets to change as the study progresses? Analysis variable definitions and dataset structures in flux may not be worth documenting too early if it leads to excessive updating of documentation to keep up with the changes.

Once the SAP is finalized, the Addendum file can be started by incorporating the analysis variable definitions. At our company, raw data is mapped to an SDTM-like structure that approximates what we plan to submit to the FDA. Given the general flow of studies, we feel our datasets are generally stable when most of the data is mapped to the SDTM-like structure and approximately 50% of the data for the study is collected. For us, this may be the ideal time to start building the Define file.

When is the right time to update documentation?

Once the framework for the Addendum and Define files is established, we recommend that updates occur on an ongoing basis throughout the course of the study to maintain consistency between study decisions and the associated documentation.

WHERE

Where can we find hints or tips to do this?

In developing our process, we found that solid documentation describing the CRT and the accompanying Define document was limited. Our primary guides were found on the CDISC and FDA websites. Specifically, we found dated material describing the first version of Define.xml on the CDISC website[3] The Case Report Tabulation Data Definition Specification document, written by the CDISC Define.xml team, describes in detail "the standard for providing [CRT] data definitions in an XML format "[3] to the FDA or similar regulatory authorities. The website also provides examples of the Define data documentation in xml, with and without reference style sheets. Although we have not yet embraced xml, we found the table structure provided in their style sheet reference example to be helpful in building our pdf and believe that this could serve as an additional aide to companies who wish to develop a similar process. It is also our expectation that this information will be useful as we prepare to transition to xml in the future.

In addition to the aforementioned materials on Define.xml, we used the Metadata Submission Guidelines Appendix to the Study Data Tabulation Model Implementation Guide[4] , prepared by the CDISC SDS Metadata team, as a reference while developing our process. The FDA website repeats information provided by CDISC[1] and also provides useful background materials on the larger electronic Common Technical Document[5] of which the CRT is an integral part.

Where else might we look for tips?

Most companies will also have employees or consultants who have worked on submissions in the past and can contribute knowledge gained through these experiences. In developing our process, we frequently looked back in order to look forward and used our past experience to supplement the guidance provided by CDISC. Reviewing SAS® programs used to generate the CRT for prior submissions provided us with a clearer picture of how we might best achieve our end product. Internal expertise was also solicited directly from knowledgeable colleagues. Prior to departing, our internal expert on the CRT process met with a small team to describe in detail the work he had done. He provided notes on his process and explained the steps he had followed for each submission. While our goal was to simplify this process and to remove the requirement of technical prowess, his documentation was invaluable in helping with our interpretation of the industry guidelines and gave us a strong backbone of code for moving forward. We recommend that companies developing submission processes consult experienced colleagues for additional tips.

HOW

How do we build the CRT?

Early in our process development, we determined that the best approach to streamlining CRT production was to divide the work into three modules: data, addendum, and define.

- Data

The data module includes documentation on the study metadata as well as actual clinical data in the form of SAS® transport files. For metadata, we used Microsoft Excel to create template shells representing the datasets, the variables and the value-level metadata. The shells contain fields required for creation of the Define documentation.

The dataset metadata spreadsheet contains the following columns:

Variable	Description	Example
Dataset	Dataset name	AE
Location	SAS® transport file name	ae.xpt
Purpose	Dataset type	Tabulation
XPT size	Size of the transport file	463 KB
Observations	Number of observations in the dataset	126
Description*	Dataset label	Adverse Events
Structure*	Dataset structure	One record per adverse event per subject
Class*	SDTM dataset class	Events
Keys *	Unique identifiers in dataset	STUDYID, SITEID, SUBJID, AESPID, AESTDTC, AETERM
Program*	Name of the program that generates the SAS® dataset	ae.sas
Usr_Comment	Any additional comments	This dataset contains all adverse events collected on the CRF.

The variable metadata spreadsheet contains the following columns:

Variable	Description	Example
Dataset	Dataset name	AE
Variable	Variable name	AETERM
Type	Variable type	Text
Order*	Variable order in the dataset	007
Description*	Variable label	Reported Term for the Adverse Event
Format*	Variable format or controlled terminology	$200.
Origin*	Variable origin	CRF
Role *	SDTM variable role	Topic
CRFPage*	CRF page number for variable collection	Page 49
Addendum*	Identifies whether variable has additional information provided in the addendum	N
Usr_Comment	Any additional comments	This is the verbatim term reported on the CRF by the site.

The value-level metadata spreadsheet contains the following columns:

Variable	Description	Example
Srce_var	Short name for the source variable being described	IETESTCD
Value	Possible value for the source variable	I03
Rel_var	Related variable for source variables that may be tied to other source variables	
Description*	Full name for the source variable being described.	BMI between 18-32 kg/m² and body weight between 50-100 kg
Type*	Variable type	Text
Format*	Possible responses based on the specified value	Y, N
Origin*	Variable origin	CRF
Role *	SDTM variable role	Topic
CRFPage*	CRF page number for variable collection	Page 7
Usr_Comment	Any additional comments	This is inclusion question #3 for the eligibility criteria.

For all tables, variables identified with an asterisk (*) have two variables associated with them. The first variable is the default obtained automatically through submission of a SAS® macro (%makexls_meta.sas). The default values selected by the macro come from both the study data and from a global metadata library.

The second variable is a user-defined version that the CRT programmer manually enters into the spreadsheet, when applicable. This layer allows the programmer to override the default entry if the pre-populated value is incorrect or not applicable in its current form.

In addition to the metadata, a key component of the data module is the SAS® transport files. Recognizing that some data formatting issues may escape unnoticed during normal study conduct, we built a macro (%xptchanges.sas) that contains code for modifying the source datasets during the creation of the target XPT files. The xptchanges macro is called by the program that generates these files. The modifications made by the macro do not alter nor destroy any results but can be useful when data provides conflicting information on a variable that should be unique. For example, in developing our process, we found a case where we had mapped the same lab test code to two slightly different variations of the same lab test description, depending on whether the result came from a local or a central lab. Since the database had already been locked when this was noted and there was no effect on the analyses, we decided to make a modification to the lab test description at the time the transport files were created.

- Addendum

The addendum module is largely completed through manual entry and programmer knowledge of the study. For our CRT process, we created a template spreadsheet that identifies the structure needed for Addendum. The spreadsheet consists of one worksheet per dataset and the following columns on each worksheet: text, bookmark, and destination. Text provides the actual text that will be displayed in the addendum file. Bookmark identifies whether the associated text will become a bookmark in the subsequent pdf file; per our process, dataset names, variable names, and additional header details are set as bookmarks. Destination is an indicator variable that specifies whether or not the text will be linked to the Define document. Once the spreadsheet has been created, a SAS® program is run to generate the Addendum postscript file. This program labels destinations in a consistent fashion, by dataset and variable name, to simplify linking to the Define documentation.

- Define

The define module requires that the data and addendum modules have been initiated, as it relies on their contents to execute. The Define document is built by a SAS® program (createdefine.sas) that executes a series of macros that read in the data spreadsheets and build links and bookmarks between tables and external PDF documents (such as the addendum and blankcrf) and transport files before producing a final postscript file. The postscript file is then distilled within Acrobat to produce the final Define.pdf. Assuming that the data spreadsheets have been created and the associated files saved to their appropriate destinations, the createdefine.sas program is completely standardized and can be used by any study without modification.

The workings of the three modules come together through a fourteen step process that begins with folder setup and ends with checking the validity of the imbedded hyperlinks in the final deliverable. A list of our steps is provided in Table 1 to demonstrate the simplicity of our new process.

Table 1: CRT Process

Step #	**Step**
1	Setup CRT folders
2	Setup StdInclude
3	Copy BLANKCRF.PDF into the TABULATONS folder.
4	Create Excel shells to hold metadata
5	Create XPTCHANGES.SAS macro shell to hold special code
6	Create and run CALL_MAKEXPT.SAS in the Programs directory.
7	Create and run CALL_MAKEXLS_META.SAS in the Programs directory.
8	Edit data metadata files if necessary

9	Edit addendum metadata file if necessary
10	Create and run CALL_ADDENDUM.SAS in the Programs directory.
11	Create and run CALL_CREATEDEFINE.SAS in the Programs directory.
12	Distill the PS files.
13	Copy analysis SAS® programs into ANALYSIS\PROGRAMS folder
14	Check links using Adobe Acrobat ISI toolbox.

A number of the steps simply relate to the copying of files including pdf documents (e.g. blankCRF), SAS® programs (e.g. call_createdefine.sas), and Excel templates. The most time consuming aspect is updating the metadata and Addendum spreadsheets to reflect study specifics. We expect this process will be a vast improvement over our previous methodology.

WHY

Why should we develop a new process for assembling CRTs?

As stated earlier, a lull in our submission schedule and new submission requirements presented us with the perfect opportunity to re-examine and streamline our past process. We decided to take advantage of this moment for a number of reasons.

First, as noted, our past process was stressful and inefficient. In rethinking how we assemble CRTs, it made sense to develop a manageable strategy for primary programmers involved in the day-to-day workings of studies. We believe we have done this by redistributing CRT related workload to earlier study stages, by automating technical aspects of the CRT build, and by defining clear steps for CRT assembly. We expect that the redistribution of the CRT related workload will help alleviate stress related to timelines. Streamlining technical issues will, in part, alleviate the need to involve programmers unfamiliar with studies at the last minute. By clearly defining our process, we have made the CRT build accessible, repeatable, and easy to learn for any programmer in our group.

Second and probably most importantly, we are a growing company. We anticipate our work load will increase greatly within the next few years. As such, we have been working to standardize a number of our processes to make our daily tasks more efficient. It follows that that this would be one of those processes.

CONCLUSIONS

We think this approach makes sense for our needs. The actual time spent creating and assembling the CRT has not necessarily been reduced, but we have redistributed much of the activity to start earlier and have created a well-defined process which is simple, efficient, consistent and repeatable. Since we have defined a process that utilizes the primary study programmer, there is no need to transfer specific study knowledge. The model shifts the technical aspect to a more programmatic approach, and allows the primary programmer's familiarity with the data to shine through.

We have implemented this approach on a few pilot studies. These studies are in different stages of completion at the time of writing this paper. Some have completed, some are ongoing, and some are at the beginning of enrollment; similarly, each CRT is at a different stage of completion.

There is synergy gained from learning the study while simultaneously creating the CRT. As the primary programmer understands the algorithms of the study he or she can immediately apply that knowledge to document the CRT.

REFERENCES:

[1] FDA Study Data Specifications (www.fda.gov/cder/regulatory/ersr/Studydata.pdf)

[2] SAS® FAQ on XPORT format (http://www.sas.com/govedu/fda/faq.html)

[3] CDISC website on define.xml (www.cdisc.org/models/def/v1.0/index.htm)

[4] Metadata Submission Guidelines Appendix to the Study Data Tabulation Model Implementation Guide (www.cdisc.org/models/sdtm/v1.1/index.html)

[5] .FDA Electronic Common Technical Document (eCTD) (www.fda.gov/cder/regulatory/ersr/ectd.htm)

CONTACT INFORMATION

Christine Connolly
Associate Statistical Programmer
35 Landsdowne Street
Cambridge, MA 02139
35-6 (6174G)
(617) 551-3658
Christine.Connolly@MPI.com

Kevin King
Senior Manager, Statistical Programming
35 Landsdowne Street
Cambridge, MA 02139
35-6 (6140)
(617) 444-3104
Kevin.King@MPI.com

Amanda Tweed
Statistical Programmer
35 Landsdowne Street
Cambridge, MA 02139
35-6 (6124)
(617) 551-8760
Amanda.Tweed@MPI.com

Steve Wong
Associate Director, Statistical Programming
35 Landsdowne Street
Cambridge, MA 02139
35-6 (6132)
(617) 444-1604
Steve.Wong@MPI.com