

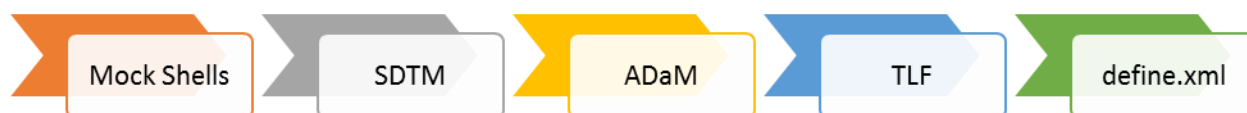
Harnessing the Web to Streamline Statistical Programming Processes

Daniel Boisvert, Biogen, Cambridge MA, USA
Pankaj Kumar, Biogen, Cambridge MA, USA
Gordon Schantz, Biogen, Cambridge MA, USA

ABSTRACT

Custom web applications allow Biogen to produce highly extensible, data-driven niche systems with low overhead and relatively short development cycles. Over time, Biogen has identified specific areas that can be automated or simplified and has created a set of interconnected applications that provide a controlled and efficient environment for statistical programming. The three core applications are: Workflow - a request tracker, Phoenix - an analysis dataset designer and define.xml authoring tool, and Venus - an electronic table shell maker. This paper will introduce our application suite, our development approach and our future vision, to examine the possibilities of custom web development for statistical programming.

CHALLENGES IN THE STATISTICAL PROGRAMMING PROCESS



A statistical programming process involves all the tasks listed above as well as the project management of each. Typically, statistical programmers do not have access to interconnected off the shelf systems to aid them in their work. Instead they may have home-grown SAS® macros mixed with Excel trackers. These have obvious limitations.

The process begins with the creation of mock table shells, which serve as the specifications for the analyses and as an addendum to the statistical analysis plan. Statisticians are responsible for this document and tend to draft their mock shells in MS-Word. This is a time-consuming task that does not make any of the information available for downstream usage. Often, they are frustrated by spending time formatting columns, aligning mock numbers and constantly having to revisit the document to keep it in sync with the changing outputs. Also, table standards are difficult to enforce with this manual approach. Statisticians may initially reference the standard, but over time will reuse existing mock shell documents, not taking advantage of new versions as they are deployed.

For SDTM and ADaM dataset creation, a basic approach involves writing specifications in Word. When the SAS program is drafted, metadata stored in the Word document will be manually reentered into the SAS program, and manually input to the define.xml (e.g., dataset label, dataset sort order, variable label, variable position). A more advanced solution would be to keep that metadata in Excel and use a SAS program to extract and apply it in the code. Excel, however, is limited in its functionality. For example, there is no concept of an audit trail in Excel. Secondly, individual Excel files do not force all information to be in a single central location. This reduces the ability to quickly locate specifications for any analysis. Lastly, variables that exist in one dataset have to have their metadata copied over to downstream datasets where the same variable exists. For example, ADSL.RACE would have to be redefined in ADLB. A time-consuming process ensues as the two variables would have to be kept in sync as their attributes change.

Dataset and table, listing and figure creation (we refer to each as a request) require a high level of project management to be successful. Stakeholders constantly ask to be updated on the completeness of deliverables and need to know the QC state of each. As the size of the analysis project grows and the number of programmers working on it increases, understanding the total amount of work completed and total amount left to do quickly becomes a very difficult question to answer. In a typical process, a primary programmer would draft the program

PhUSE US Connect 2018

then email the QC programmer to tell them to start QC. The back and forth, until the output is finalized, would be tracked through email. With this information locked in individual emails, it becomes near impossible for anyone to know the QC state of all requests. Also, without a centralized database to monitor each request and who is working on it, programmers' workload is hard to track. If one is overloaded and one is needing work, it is up to the lead programmer to make this distinction thereby distracting them from their programming tasks.

It is due to these limitations and the lack of commercially available software that we decided to build bespoke web applications to aid statistical programming.

APPLICATION DEVELOPMENT APPROACH

When developing an application, it is important to note that a system does not correct a poor business process. Time must be taken to understand the tasks at hand and identify clear use cases where an application can provide value. We have found that it is better for the process to exist prior to trying to automate it. While sounding counter-intuitive, we do not recommend creating a new business process for the sake of new technology. Instead, begin by creating a new business process leveraging standard technology (paper based, excel based, email based) then use the lessons from that to provide requirements for the system. When running a low-tech process users may realize the process does not often occur, or works well without technology, or is completely impossible without technology. It is the last case in which we invest.

WHY WEB?

Accessibility: Users only need a web browser, which exists on virtually all modern computers. The full application suite is easily accessible via a URL without any separate software installation.

Open source, no licensing: Our applications are written predominately in PHP, JavaScript, Python, and MySQL, which are available for use with no licensing fees and are backed by large user communities. Packages to automate common tasks are available and can be applied with little configuration (e.g., PDF generation).

Rapid development: Web applications are fast to develop and troubleshoot because they are predominately based on modules and procedures that have already undergone a QA process and do not need to be reinvented. Since changes do not need to be compiled to debug, the developers can unit test their changes immediately.

Easy Deployment: Global updates can be rolled out easily because all the changes are applied on the server side and no reinstallation is required.

APPLICATION DESCRIPTIONS

WORKFLOW - STEAMLINING PROJECT MANAGEMENT

Clinical trial analysis involves a lot of moving pieces. An analysis consists of many datasets and TLFs, being created by a team of programmers statisticians. The project management of this is itself a full-time job. Wanting analysts to focus on analytics instead of project management, we decided to build an easy-to-use system to track all the work of statistical programmers. It was out of this necessity that our first application was built, Workflow. Since its inception in 2005 Workflow has grown to be a hub of tools to optimize statistical programming work.

Workflow, at its core, tracks requests through their work cycle. A request may start with a programmer, then go to a QC programmer, then back, then on to a statistician then finally to medical writing. Each of these transactions is tracked, and the data is recorded, by Workflow. Beyond this core functionality, Workflow has grown to automate other statistical programming processes. Among these are running a program or a batch of programs, scheduling a batch, creating a compiled PDF, tracking dependencies, and comparing subsequent runs to automate QC.

A key asset of Workflow is the amount of data that it contains. Every work product (dataset specs, dataset code, TLF codes) produced over the last 12 years have their history recorded in Workflow. This includes every transaction between programmers, statisticians and medical writers related to analyses. This data is used in project management dashboards and management metric reviews.

KEY FEATURES

Project management

Using Workflow as a project management tool allows programmers to monitor their study by seeing an overview of the study progression by status (development, QC or under Stat review). Workflow stores tracking history of every request in a centralized database, which, in turn, makes the data readily available for study/resources analysis and reporting

PhUSE US Connect 2018

Interface to run SAS

Being able to run SAS® program through Workflow allows programmers to manage their projects along with creating and updating outputs without having them to switch to a different application. Adding a Web user interface to a SAS macro can make training and use more intuitive. For example, we have an interface to pull data from EDC which is accomplished with a SAS macro to which we pass the schema and dataset names. Then the SAS code executes as usual and creates the output files and log. To the user, it is just a couple clicks and then the data is placed in the correct location. A record of the transaction is saved in the Workflow database.

Metrics Reporting / Resourcing

The Workflow data enables us to generate various type of reports. The study quality report, workload spreadsheet and comparative effectiveness of vendors are some of the examples.

SCREENSHOTS

The screenshot shows the 'Workflow - drug/study/csr' interface. The top navigation bar includes 'Workflow', 'Getdata', 'Setlist', 'Phoenix', 'Resources', and 'User Manuals'. A search bar on the right contains 'drug/study/csr'. Below the navigation bar, there are links for 'Need Help?', 'What's New', and 'Logout (pkumar)'. The main content area displays 'Requests by Project (drug/study/csr): 54 requests found'. A table lists various requests with columns for project name, user, date, status, and actions. The table includes links to 'Export Prolog' and 'Batch Processing'. The table data is as follows:

Project Name	User	Date	Status	Actions
Exploratory Analysis	Daniel Boisvert to Daniel Boisvert	01/30/18 09:30	Unplanned	exploratory.sas, qc-exploratory.sas, Table, CR#105800
Testing histogram code	PR: Vineeth Kurnala QC: S2:	04/18/17 11:30	Development	** development started ** f-histogram-r.sas, QC not tracked in WF, Graphs, CR#92246
Laboratory category <mean/median> values by visit - safety population	PR: Pankaj Kumar QC: S2:	10/12/17 11:55	Planned	** program overwritten from testproduct/templatestudy/csr ** f-lb-byvis.sas, qc-f-lb-byvis.sas, Graphs, CR#101453
Values over time for subjects with potential serious hepatotoxicity	PR: Pankaj Kumar QC: S2:	10/12/17 11:55	Development	** development started ** f-lb-heptox-byobj.sas, qc-f-lb-heptox-byobj.sas, Graphs, CR#101466
Evaluation of potential serious hepatotoxicity	PR: Pankaj Kumar QC: S2:	10/12/17 11:55	Planned	** program overwritten from testproduct/templatestudy/csr ** f-lb-heptox.sas, qc-f-lb-heptox.sas, Graphs, CR#101465
Listing of serious adverse events - safety population	PR: Pankaj Kumar QC: S2:	10/12/17 11:55	Planned	** program overwritten from testproduct/templatestudy/csr ** l-ae-ser.sas, QC not tracked in WF, Listing, CR#101280
Listing of adverse events that led to discontinuation of study drug - safety population	PR: Pankaj Kumar QC: S2:	10/12/17 11:55	Planned	** program overwritten from testproduct/templatestudy/csr ** l-ae-wdw-drug.sas, qc-l-ae-wdw-drug.sas, Listing, CR#101459

The screenshot shows the 'Workflow Metrics Reports' interface. The top navigation bar includes 'Workflow' and 'User Manuals'. A search bar on the right contains 'drug/study/csr'. Below the navigation bar, there are links for 'Need Help?', 'What's New', and 'Logout (pkumar)'. The main content area displays a grid of report categories:

Study Reports	Comprehensive Reports	Resource Management
Analysis Status Report - Select one or more study/projects to see SDTM, ADaM and TIGs requests grouped by different tags (first look, second wave etc.) Quantity and Quality Report - Report shows quantity and quality metrics from a study/project Report by Person - Pick a user with date range to see all his requests with revisions history Report by Product - Pick a product, study or project with date range to see all associated requests including revisions history Inspection Readiness Team Lists - Generate a report of current team list working on a study Dev Run Comments Tracker (individual) - Pick a product/study/project and a Setlist to view the report. Also, Export to Excel spreadsheet option available.	Overall QC Report - Single report shows number of requests returned to development for each study/project throughout workflow database Overall Status Report - Single report shows number of requests returned to development for each study/project throughout workflow database ODRP Report - Single report shows number of requests grouped by status for each ODRPs throughout workflow database	Study report by Programmers - Report displays number of workflow requests touched by Programmers. Also report shows number of total revisions generated by each programmer. Resource Report by Study - Report displays number of people worked on one, two or three studies in a custom date range Resource Report by Person - Report displays number of studies worked by people in a custom date range

PhUSE US Connect 2018

VENUS – STREAMLINING MOCK SHELL CREATION

Venus is an electronic table shell designer. Users, typically statisticians, use Venus to implement standard table shells and to create study specific ones. To ensure compliance, Venus controls access and does not allow users to edit standard shells. Venus has solely the functionality required to make table shells. Users can enter text, indent text, insert columns, insert rows. This straightforward approach, chosen over a full-fledged rich text editor, allows the statisticians to focus on their exact needs.

KEY FEATURES

Paper to Data

Table shells are historically kept in Word, making it difficult to mine or leverage the data later. Venus provides an interface for users to enter table shells and save that information as data. This centralized database enables Venus integration with other applications and opens the opportunity to automate table generation.

Leverage Data in Other Applications

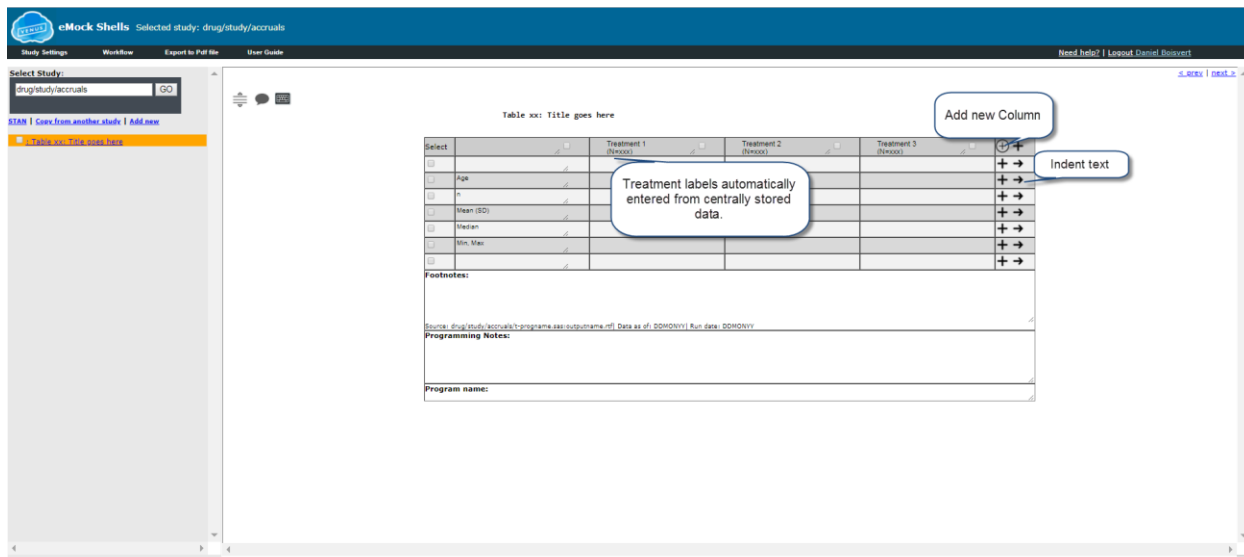
Venus automatically creates requests in Workflow and inserts titles and footnotes which are then received by the SAS program and added to the output. There is no chance of titles and footnotes getting out of sync from the shell to the output.

Centralized Data Definition

Data can be written once and used multiple times. In Venus, treatment labels can be authored centrally then serve as the default when a new shell is created. This allows statisticians to update treatment headers in one place and have it propagate as requested to all the tables.

SCREENSHOTS

MAIN SCREEN



PhUSE US Connect 2018

STANDARD TABLE IMPORT

Save to drugs study accruals
Close and refresh main window

Select Phase:
no selection

Demographics
Demography - <population>

Adverse Events

- Overall summary of adverse events - safety population
- Adverse events by system organ class and preferred term sorted by decreasing frequency - safety population
- Adverse events by system organ class and preferred term sorted by alphabetical order - safety population
- Adverse events by system organ class - safety population
- Adverse events at least x% higher in incidence by system organ class and preferred term for <active> - safety population
- Adverse events by preferred term - safety population
- Adverse events by preferred term with an incidence of x% or more safety population
- Severe adverse events by system organ class and preferred term - safety population
- Severe adverse events by preferred term - safety population
- Adverse events by maximum severity by system organ class and preferred term - safety population
- Adverse events by maximum severity by preferred term - safety population
- Related adverse events by system organ class and preferred term - safety population
- Serious adverse events by system organ class and preferred term - safety population
- Serious adverse events by preferred term - safety population
- Listing of serious adverse events - safety population
- Related serious adverse events by system organ class and preferred term - safety population
- Adverse events that led to discontinuation of study drug by system organ class and preferred term - safety population
- Listing of adverse events that led to discontinuation of study drug safety population
- Adverse events that led to withdrawal from study by system organ class and preferred term - safety population
- Listing of adverse events that led to withdrawal from study - safety population
- Adverse events that led to <drug interrupted, dose reduced, dose increased value to match CRF> - safety population
- Incidence rates of adverse events by system organ class and preferred term - safety population

Laboratory Analyses

- Hematology actual values by visit - safety population
- Hematology <mean/median> values by visit - safety population
- Hematology change from baseline by visit - safety population

Copy all standard tables in one click.

[illegible]

PHOENIX - STREAMLINING DATASET SPECIFICATIONS

Phoenix was initially conceived to facilitate the creation of define.xml files from XPT dataset files, as a replacement to a notoriously buggy off-the-shelf product that the company had been using for this purpose. It stands as a good example of the user community steering design and development, because it has evolved from this simple task to be a dataset design tool. Historically, time-consuming and error prone manual processes were used to write analysis dataset specifications in Word then update the SAS program referencing the metadata in the document. If the document was updated, the SAS program would have to be updated manually, and there were no automated checks in place to ensure the two were kept in sync. At submission time, the define.xml would be created from the datasets and the Word specifications. Since we already had a tool that stored the study's metadata and could create the XML output, it was a short step to optimize the process, removing the need for Word, Excel, and external documents.

KEY FEATURES

Audit Trail

By implementing version history when records are saved, it is possible to determine who changed what and when. This is useful not only in the case of a system audit, but if a comment or origin suddenly contains an unexpected value and needs to be tracked back to the source of the edit. Phoenix includes a set of version reports and something resembling "diff" in the Unix world that can show all changes in a dataset that occurred between any two dates. Beyond versioning, we have also deployed an annotation system that saves its data outside of the study metadata, where programmers and statisticians can leave commentary and have conversations about a given variable.

Enforce Standards

When creating a dataset, SDTM and/or ADaM standards are met through a simple process of automatically including all required variables and only allowing the addition of variables valid to the given domain. Phoenix then fills in any appropriate information

PhUSE US Connect 2018

such as Role, Core or Label, avoiding the likelihood of typographical errors. Likewise, the correct codes for controlled terminology for multiple CDISC versions are stored and available for use, with all necessary codes, submission values, and preferred terms available.

Links

Synchronization of data between domains is further ensured by variable links. If the base variable is edited, the edit propagates throughout the study in all ADaM datasets that are deriving this variable. So for example, if the length of DM.AGEU is modified, the change automatically carries over to ADSL.AGEU, ADEXP.AGEU, etc. with no extra effort on the part of the user.

Define.xml generator

The original purpose of Phoenix remains one of the most compelling, because literally, clicking a single button on the screen results in the creation of a complete define.xml file which is then ready for validation against the data that will be submitted to the regulators.

SCREENSHOTS

DATASET MAIN PAGE

Current Study: study-training:training - eSpec (drug/study/training)

Study Insert/Edit Data Other Resources Logout dboisver

Edit Dataset/Variable Level Metadata

Dataset: ADSL Dataset Options

Label: Subject Level Analysis Dataset Purpose: Analysis Class: SUBJECT LEVEL ANALYSIS
DATASET
Structure: One record per subject
Keys:
Notes:

Variable Level Metadata [Add new variable](#)

Variable	Label	Comments (for define.xml)	Mapping Spec (Comment for Programmer)	Computational Method	Code List	Length	Type	Sig Digits	Origin	CRF Page Number	SAS Format Name	History
STUDYID	Study Identifier	DM STUDYID	DO NOT REDERIVE, TAKE FROM DM			8	text	0	Derived			
USUBJID	Unique Subject Identifier	DM USUBJID	DO NOT REDERIVE, TAKE FROM DM			20	text	0	Derived			
SUBJID	Subject Identifier for the Study	DM SUBJID	DO NOT REDERIVE, TAKE FROM DM			6	text	0	Derived			
SAFFL	Safety Population Flag			13	Create / Edit N	0	text					
RFSTDTCT	Subject Reference Start Date/Time	DM RFSTDTCT	DO NOT REDERIVE, TAKE FROM DM			19	text	0	Derived			
RFENDTCT	Subject Reference End Date/Time	DM RFENDTCT	DO NOT REDERIVE, TAKE FROM DM			19	text	0	Derived			
AETERM	Reported Term for the Adverse Event	AE AETERM	DO NOT REDERIVE, TAKE FROM AE			154	text	0	Derived			
AESEV	Severity/Intensity	AE AESEV	DO NOT REDERIVE, TAKE FROM AE		AESEV	8	text	0	Derived			
AESER	Serious Event	AE AESER	DO NOT REDERIVE, TAKE FROM AE		NY	1	text	0	Derived			
RACE	Race	DM RACE	DO NOT REDERIVE, TAKE FROM DM		RACE	25	text	0	Derived			
SEX	Sex	DM SEX	DO NOT REDERIVE, TAKE FROM DM		SEX	1	text	0	Derived			
AGE	Age	DM AGE	DO NOT REDERIVE, TAKE FROM DM			8	integer	0	Derived			
AGEGR	Age Group	If DM AGE is <10 and not missing, then AGEGR=<10; If DM AGE >=10 then AGEGR = >=10.		13	Create / Edit AGEGR	20	text	0	Derived			

CREATE A VARIABLE

Add Variables

☐ **Link**

- Manage links to variables in other datasets.

☐ **Copy Product Metadata**

- Copy variable information from this product's other studies.

☐ **Standard ADaM Metadata**

- User selects from a list of standard ADaM variables available to this dataset

☐ **Custom Variable**

- Hand-crafted variables uniquely suited to the programmer's tastes and desires

[Continue](#) [Cancel](#)

PhUSE US Connect 2018

EDIT VARIABLE SCREEN

The screenshot shows a web-based form titled "Edit Variable: PARAMCD". The form contains several input fields and dropdown menus. The "Name" field is labeled "Name **" and contains the text "PARAMCD". The "Label" field is labeled "Label **" and contains the text "Parameter Code". The "Type" field is labeled "Type **" and is a dropdown menu with "text" selected. The "Length" field is labeled "Length **" and contains the number "0". The "Sig Digits" field is labeled "Sig Digits" and contains the number "0". The "Origin" field is labeled "Origin **" and is a dropdown menu with "Assigned" selected. There are also fields for "CRF Page No." and "SAS Format Name" with a dropdown menu showing "- SAS Format Name -". At the bottom, there are two large text areas: "Comments (for define.xml)" and "Mapping Spec (Comment for Programmer)". A legend at the bottom left indicates that "** Required Field". At the bottom right, there are two buttons: "Cancel" and "Save".

FUTURE

As we start to think about the next steps for our application suite, we are exploring ways to take all the data that we have amassed and put it to work for us. There are two main projects under review, new analytics and more automation.

ANALYTICS

PREDICT REQUEST ASSIGNMENT

Workflow contains information about all tables listings and figures over the last 12 years and everything that is due in the near future. When creating a new request, the user must think to whom to assign the tasks. The lead programmer might do a quick subjective assessment about the current workload or who is good at this type of request. Based on the data that Workflow can access, we can do a much more rigorous assessment and are able to predict who would be the best fit for the request. This will allow better use of resources and uncover programmers who are strong in a particular area (e.g., graphing specialists, analysis dataset specialists). For this project we are considering total turnaround time, number of iterations to get the output correct, number of iterations based on programming error, current number of requests assigned, and historical proficiency.

PREDICT TOTAL WORK

While we have robust prediction for core planned study work, it is harder to anticipate the exact number of datasets and outputs that will be requested over the life of the study. Typical resourcing algorithms will predict towards a deliverable (eg. CSR), but do not tell the full story of demand. Ask any programmer and they will tell you of how many requests come in around conferences, how much work is done on a study after it is submitted, or how work is still being done on that one study from 10 years ago. Now that we have so many years' worth of actual data we feel this prediction is more possible. The end goal would be to be able to input a few key parameters (e.g., phase, indication, # subjects) and get the month by month total work demand for a yet to be started study.

AUTOMATION

At this point we have individual solutions to optimize specific pieces of functionality. Now we are starting to connect our systems. The first connection that we made was to connect Workflow and Phoenix. Workflow now contains a request for a dataset specification along with a request for the actual dataset. Depending on the status of the specification request, Phoenix will lock the specification from editing thus ensuring the specification and the dataset are kept in sync.

Venus (our electronic table shell creator) and Workflow (our request tracker) are now also connected. The table shell outputs of Venus are the requests the programmers track in Workflow. Instead of maintaining the data in two places, Venus will become the interface to create and manage TLF requests. This will ensure the table specifications are always reflecting the actual output.

Our long term goal for Venus is to extend it into a table creator. We plan to go beyond the shell and automate the table creation process using only the web application for printing and Python for statistical calculations. This will allow for fast, flexible and easy table creation. As soon as a shell has been specified it will have also been "programmed" and only awaits the incoming data to be complete. To date we have completed a successful proof of concept and will continue to develop this method.

PhUSE US Connect 2018

CONTACT INFORMATION

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

Daniel Boisvert

Biogen

Daniel.boisvert @ biogen.com

Pankaj Kumar

Biogen

Pankaj.kumar @ biogen.com

Gordon Schantz

Biogen

Gordon.Schantz @ Biogen.com

Brand and product names are trademarks of their respective companies.