Paper ST07

Use Python to make Clinical Trial Result Posting automatically

Derek Li, Bayer, NJ, US
Patty Hegarty, Bayer, NJ, US

ABSTRACT

According to US FDA and EMA regulations, clinical trial sponsors are required to disclose the aggregated results for their applicable clinical trials on https://clinicaltrials.gov/ (for US) or https://eudract.ema.europa.eu/ (for EMA) or both. Most pharmaceutical companies have design a group to do this work according to the study clinical summary report(CSR) with additional manual effort, as well as drawbacks of this step, like repeatedly revising the data back-and-forth to address comments from multiple reviewers of different groups, which is quite time consuming and difficult to ensure the accuracy and consistency with CSR. However since both websites provide a portal to use XML file to upload the posting required data, and the XML file for FDA results can be created based on the EMA XML file automatically, we explore to use this feature to create a draft XML file from EudraCT database as basis, then use a python developed web app to feed it with study TLFs metadata, after doing so, the clinical trial results posting data entry becomes smoother with higher accuracy, efficiency in an automatic way, and consistent with analysis results presented in the CSR. We will show you the details in this paper below about how to use this web app.

INTRODUCTION

For FDA basic results, the requirements of clinical trials' results disclosure on ClinicalTrials.gov can be found in data elements definitions defined by ClinicalTrials.gov at https://prsinfo.clinicaltrials.gov/results_definitions.html(CTGOV), and the required information includes:

- Results Point of Contact
- Certain Agreements
- Participant Flow
- Baseline Characteristics
- Outcome Measures
- Overall Limitations and Caveats
- Adverse Events

Similarly, for EMA, the detail requirements can be found at https://eudract.ema.europa.eu/multimedia_tutorials.html.(EudraCT), which includes:

- Trial Information
- Subject Disposition
- Baseline Characteristics
- End Points
- Adverse Events
- More Information

After explore above requirements in deeper, it is noted for both CTGOV and EudraCT disclosures, the clinical summary report(CSR) contains the data commonly required by both formats for:

- Trial Information
  - Count of subject of each country
  - Population Age Group
- Subject Disposition
  - total number of subjects completed / non-completed trial
  - reasons of discontinuations
- Baseline Characteristics
  - Age
  - Gender

  - Categorical Baseline Characteristics
  - Continuous Baseline Characteristics
- End Points/Outcome Measures
- Adverse Events

These aggregated results (except population age group) can be obtained from the TLFs metadata, which are derived based on the CDISC ADaM datasets per the TLF spec and used by the clinical trial study report. Both CTGOV and EudraCT registry systems have a provision to download and upload a XML file to post the clinical trial result with prespecified requirements/schemas, and the CTGOV required XML file can be converted from EudraCT's XML file, so we decide to use the EudraCT's XML file as the original input XML file, use python to develop an docker based web app to fill it with study TLFs metadata, to improve the accuracy, efficiency and consistency of results disclosure. And the design flow chart is:

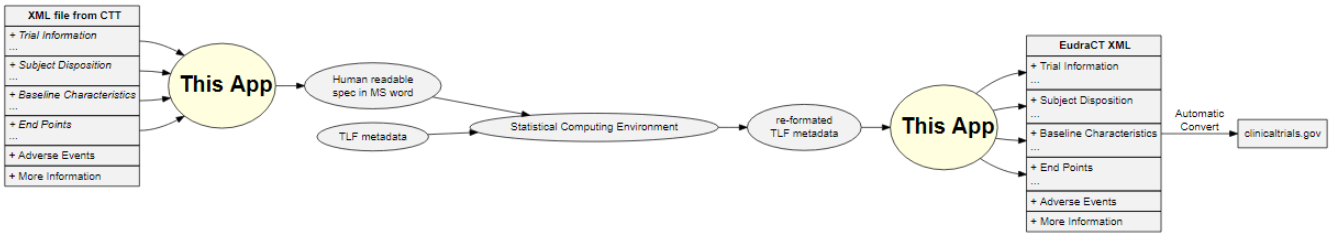
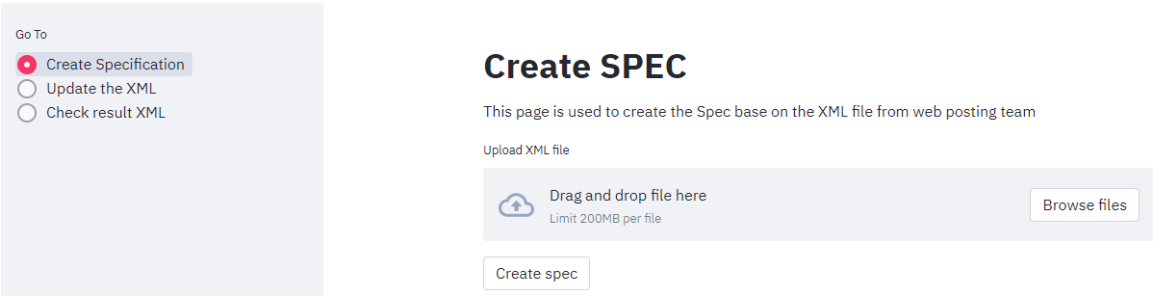


Figure 1: Work flow of the automation process

DETAIL STEPS

Create specs from XML file

As XML file format is not human read friendly, to overcome this weakness, the first step is to create a human readable specification with Microsoft word format(DOCX), which is just like a TLF specification for our TLF programming during our general work as a statistical analyst.



Fgure 2: Create specification from input XML

From this specification, we know which kind of data are needed and should be in what kind of format, like below snapshot:

Periods and discontinuation reasons

Shells

phase	period	periodid	categoryid	details	other	group0	group1	group2	group3
0	1	PostAssignmentPeriod-46431	CompletedMilestone-113150	complete treatment	xx	xx	xx	xx	xx
0	1	PostAssignmentPeriod-46431	StartedMilestone-113151	start treatment	xx	xx	xx	xx	xx
999	1		ReasonNotCompleted-126704	physicianDecision		9	9	9	9
999	1		ReasonNotCompleted-126705	other	Private reason	9	9	9	9

period: flag variable of each period, mandatory

periodid: ID of each period, extract from input xml, only for reference

phase: flag variable of not complete reason of each period, only need to populate 999 for all withdraw reasons, mandatory

categoryid: ID of each category,extract from input xml, mandatory

details/other: detail wording of each category, extract from xml and for reference

groupxx: Treatment group variables

NOTE: Only exist reasons in input dataset will be kept

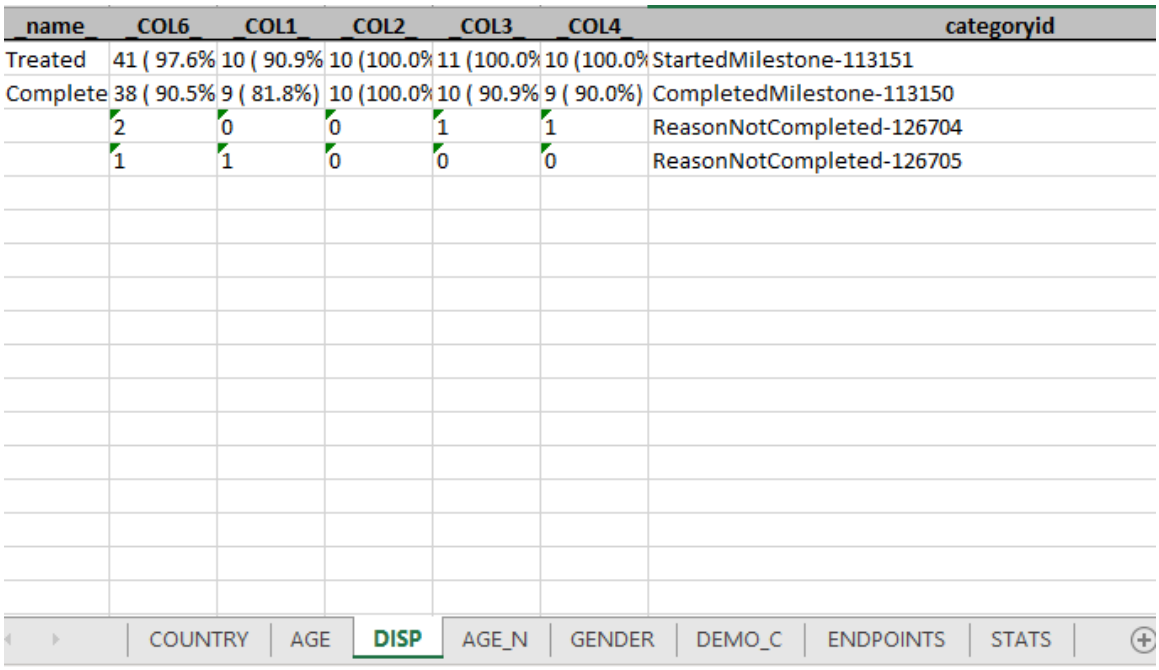
Fgure 3: Specification example

Since the study TLFs metadata are stored in SAS7BDAT format, and what we need are usually not exactly matching tables from the CSR, like one part from table 14.1/3, and another part from table 14.1/4, or the data is shown in horizontal direction, but we need vertical presentation etc., which means we need to “re-format” the TLFs metadata, like split, re-combination, transpose and so on, also another tricky part is, for those categories required data, like gender, age category, each category has a specified categoryid created by the EudraCT system, and we need to combine them to the needed TLFs metadata, to make it easier, a SAS programming template is also created at this step and below SAS codes are part of it:

```
*? FOR DISPOSIITON SECTION;
*************************************************************************;
*** usually these information can be found in key table;
***? NOTE: copy the IDs from spec and update accordingly;
*************************************************************************;
data DISP;
   set TLFMETA.T_14_1_2_ADSL_DISPOVERALL_1;
   period=1;
   if _order_=int(_order_) then phase=0;
   else phase=999;
      if _order_  = 4 then categoryid="CompletedMilestone-119604";
      if _order_  = 3 then categoryid="StartedMilestone-119605";
      if _order_  = 7.001 then categoryid="ReasonNotCompleted-132869";
      if _order_  = 6.003 then categoryid="ReasonNotCompleted-132867";
      if _order_  = 6.001 then categoryid="ReasonNotCompleted-132865";
      if _order_  = 6.002 then categoryid="ReasonNotCompleted-132866";
      group0=_col1_;
      group1=_col2_;
      group2=_col3_;
      group3=_col4_;
      if not missing(categoryid);
        keep phase period  categoryid details  group0 group1 group2 group3;
run;
```

Update the XML file with data from TLFs metadata

After having the specification and SAS programming template, we can create the SAS programs to prepare the needed aggregated results from both ADSL domain and the study’s TLFs metadata, like split, set, re-format etc., and then to create an Excel file at the last step:



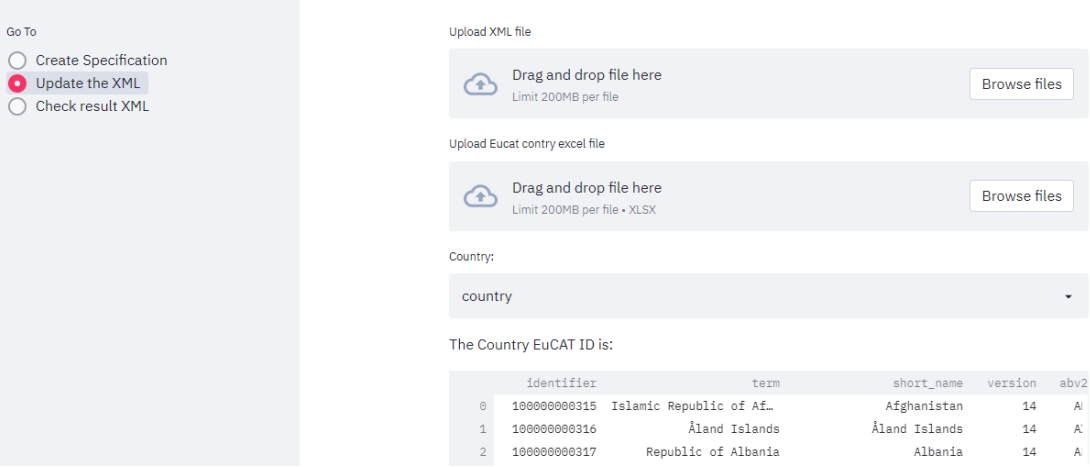
name	_COL6_	_COL1_	_COL2_	_COL3_	_COL4_	categoryid
Treated	41 (97.6%	10 (90.9%	10 (100.0%	11 (100.0%	10 (100.0%	StartedMilestone-113151
Complete	38 (90.5%	9 (81.8%)	10 (100.0%	10 (90.9%	9 (90.0%)	CompletedMilestone-113150
	2	0	0	1	1	ReasonNotCompleted-126704
	1	1	0	0	0	ReasonNotCompleted-126705

Fgure 4: Input Excel file

As shown in this snapshot, each required section from the XML file has a sheet:

- COUNTRY: Count of subject of each country, mandatory
- AGE: Population Age Group, mandatory
- DISP: Population disposition, the category may not consist with the data shown in CSR, mandatory
- AGE_N: Age summary as in CSR, optional
- AGE_C: Age category analysis as in CSR, optional
- GENDER: Gender Baseline summary as in CSR, optional
- DEMO_C: Categorical Baseline Characteristics, optional
- DEMO_N: Continuous Baseline Characteristics, optional
- ENDPOINTS: End Points/Outcome Measures, mandatory
- STATS: The statistical analysis for each end pints, optional

The meaning of each column of each sheet can be found in the specification, to save some space, we do not show the details here. After have this Excel file crated, the user can use this web app to update the input XML file by click buttons:



Fgure 5: Update XML file

As you can see from above figure, instead of using the country name, EMA have define an ID for each country with version control, which is same for AE SOC information in adverse event section, for details please see https://spor.ema.europa.eu/rmswi/#/lists/100000000002/terms, on the other hand, per the CDISC standard, the country name is shown in 3 letters, to have the country ID and version information, this web app have provide a default version, but the user can use a new version by uploading an Excel file.

Accuracy Check

Validation is a mandatory step of all our delivery as statistical analyst, during above step when we update the XML file with an input Excel file, we can consider it as a transition from Excel file to XML file, to check the updated XML file, in this step an reverse transition is used, we create an Excel file at this step by using the updated XML file as the input.

A	B	C	D	E	F	G	H	I	J
phase	period	periodid	categoryid	details	other	group0	group1	group2	group3
0	1	PostAssig	Complete	complete treatment		9	10	10	9
0	1	PostAssig	StartedMi	start treatment		10	10	11	10
999	1		ReasonNc	physicianDecision				1	1
999	1		ReasonNc	other	Private re	1			

COUNTRY | OVERALLAGE | DISPOSITION | BASELINE | ENDPOINT | STATS

Fgure 6: Check Result

After have this Excel file populated, the user can compare it with the input Excel file, which is created from the TLFs metadata at last step, to make sure all needed data are filled correctly.

CONCLUSION

As the input XML file is from EudraCT website directly, we do not need to worry about the system requirements/schemas, and also those system created categoryIDs, which means the update XML file can be used by the EudraCT website smoothly. After testing on several studies, the used time has been decreased from more than 2 days to 2 hours with high accuracy and consistency with the data from CSR.

This web app is distributed through docker to make sure the availability, so anyone from our team can use it. For those outsource studies, this app can also be used by providing vendor with the Microsoft word specification and reqest them to provide that Excel file back to us, no matter they have TLFs metadata or not.

As the requirement for adverse events section is quite different compare to the other sections, and our company already has a very well established process on this part, so this section is not covered by this web app.

After testing this web app on several studies, we note the big pain point is how to find the needed data from tens of hundreds of TLFs metadata datasets, especially when the needed data are from more than one datasets, which need to split, merge or re-format them to put all needed data together. Therefore, to alleviate this pain point, it is better to flag out which data will be used for posting during preparation of the TLF specification and analysis plan, and the best case would be each posting required data point can have a corresponding CSR table.

REFERENCES

ClinicalTrials.gov. Accessed March 8, 2018. http://www.clinicaltrials.gov
EudraCT Results related documentation. Accessed March 8, 2018. https://eudract.ema.europa.eu/result.html
Automation of Clinical Trial Result Posting to ClinicalTrials.gov and EudraCT. 2018. Sherry Meeh

ACKNOWLEDGMENTS

I would like to thank my below colleagues, who help me to do the validation and testing during this web app development.

Johnson Liu, Peter Verhallen, Reiner Faber

CONTACT INFORMATION

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

Derek Li, Patty Hegarty
Industry Regulatory Interaction-Statistical Programming Analyst,
Bayer HealthCare
Email: derek.li1@bayer.com, patricia.hegarty@bayer.com