PharmaSUG China 2023 - Paper AA-120

Implementation of an end-to-end Web-based SAS Programming Platform

Weidong Liang, Phastar Inc.

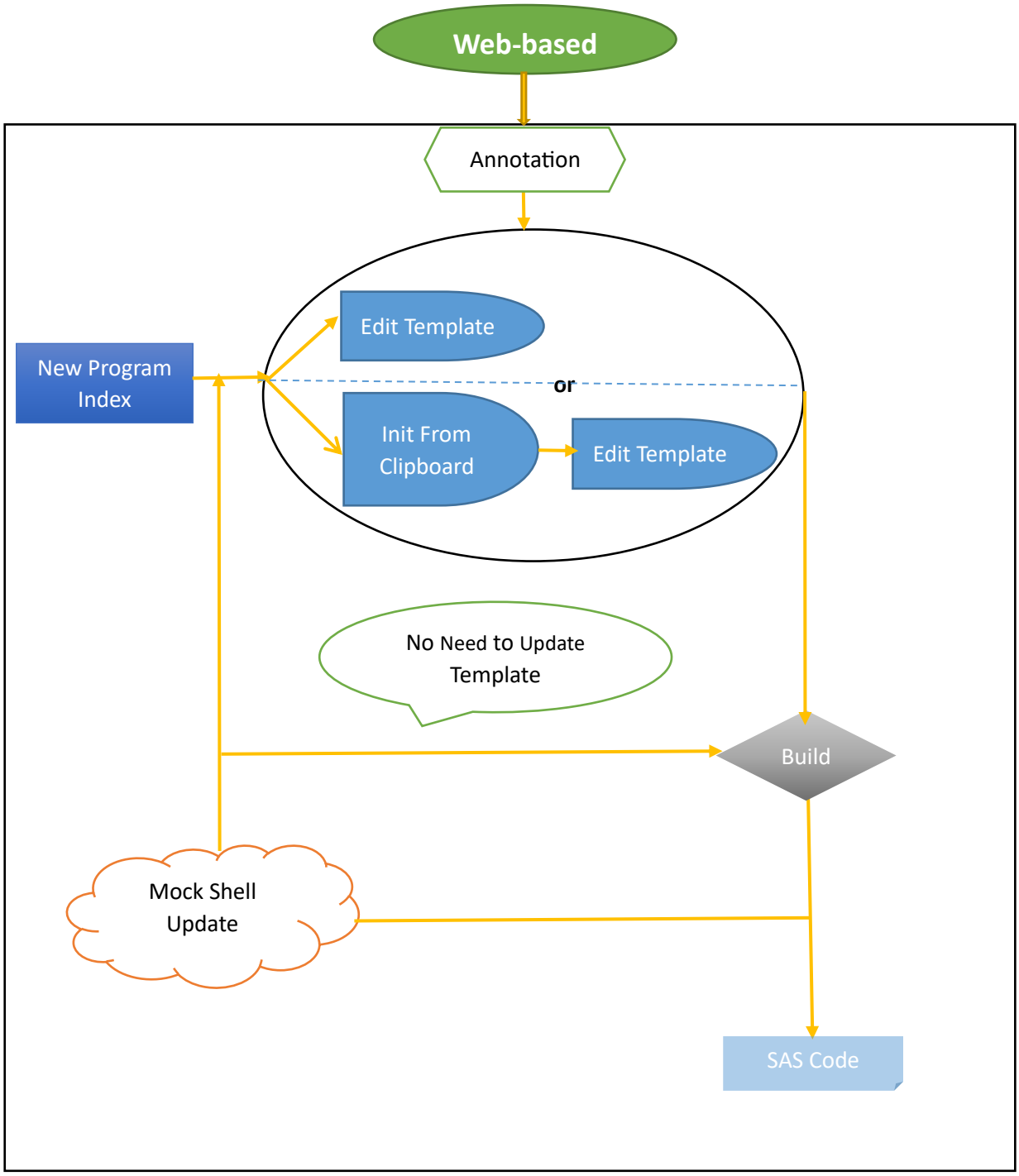
ABSTRACT

Quality and efficiency of generating clinical reports are important. Different people have different technical skills of completing tasks. Developing a set of macros or copying code from similar studies may be on the top of your list. Those are good solutions and are adopted universally in the industry. This paper describes a web-based collaboration system for TLF automation. Statistician logins to the system to create mock shells which can be downloaded for eTMF filing etc. SAS programmers annotate these shells with predefined syntax. These annotations are then used as metadata for generating SAS programs that produce the actual TLFs. Your TLF task would be performed easily just like validating submission package in PINNACLE 21.

INTRODUCTION

The system is web-based. What you see is what you will get; as such you can use it with less effort. HTML is the standard markup language for Web pages and is used to prepare mock shell. You can treat HTML as XML which contributes to its machine readable. Define.XML has a beautiful appearance and layout because of the outside style file. Similarly, HTML rendering depends on CSS (Cascading Style Sheets) and JavaScript. Contents and style definition are stored in files separately which makes it possible to parse it robustly using python. Every change you've made in mock shell will affect downstream automatically without manual efforts.

1. Schema


Web-based
Annotation
Edit Template
New Program Index
or
Init From Clipboard
Edit Template
No Need to Update Template
Build
Mock Shell Update
SAS Code

2. HTML Bundle

2.1 Supported Mock Shell

2.1.1 Common

The section demonstrates the most situation we may encounter in our daily work, calculating the number of subjects (or events) when conditions are met. The blue highlighted content (WORK.ADSL | ITTFL = "Y") has two portions separated by vertical bar (|). One is the content before vertical bar which indicates the denominator dataset, the other is the content after vertical bar which indicates the SAS where clause.

For annotations in treatment columns, we have three branches in the example below. If one record with TRT01PN of 1, then calculator of Drug A will be increased by 1.

For annotations in the body, (WORK.ADSL | Race = "RACE01") also has two portions.

Table 14.1.5.1 Demographic characteristics (Intent-to-treat)

work.adsl | ITTFL = "Y"

Demographic characteristic		Drug A (N=&BIG_1_N1) TRT01PN = 1	Drug B (N=&BIG_1_N2) TRT01PN = 2	Total (N=&BIG_1_N3) TRT01PN in (1, 2)
work.adsl | Race = "RACE01" Race n (%)	RACE01	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adsl | Race = "RACE02"	RACE02	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adsl | Race = "RACE03"	RACE03	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adsl | Race = "RACE04"	RACE04	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adsl | Race = "RACE05"	RACE05	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adsl | Race = ""	Missing	xx (xx.x)	xx (xx.x)	xx (xx.x)

Figure 1. Basic Summary of Demographic Characteristics

2.1.2 Layer

Sometimes we may report clinical results depending on actual data. In which case, it is not possible that we iterate all situations. The blue highlighted contents (WORK.ADAE. AEBODSYS#AEDECOD) indicates that input dataset is ADAE, and we have two layers separated by pound. Layer 1 is AEBODSYS and layer 1 is AEDECOD.

In most cases we usually need to sort our dataset. We can accordance with the syntax in annotation (WORK.ADAE. AEBODSYS #AEDECOD order by AEBODSYS descending NUM_CHAR_1). You may be confused with the internal variable NUM_CHAR_1. In the scenario below, NUM_CHAR_1 represents the count in Drug A and NUM_CHAR_2 represents the count in Drug B.

Table 14.3.2.2 Adverse events by system organ class and preferred term (Safety analysis set)

work.adae| SAFFL= "Y"

	Durg A (N=&BIG_1_N1) TRT01AN = 1	Durg B (N=&BIG_1_N2) TRT01AN = 2
work.adae.aebodsys#AEDECOD --SOC#PT--	xx (xx.x)	xx (xx.x)

Figure 2. Basic Summary of Adverse Events by Soc, PT

The system also supports counting events. XX is capitalized, which is the place holder of the number of events.

Table 14.3.2.3 Adverse events by system organ class and preferred term (Safety analysis set)

work.adae| SAFFL= "Y"

	Durg A (N=&BIG_1_N1) TRT01AN = 1	Durg B (N=&BIG_1_N2) TRT01AN = 2
work.adae.aebodsys#AEDECOD --SOC#PT--	xx (xx.x) XX	xx (xx.x) XX

Figure 3. With Calculating Number of Events

2.1.3 Stat

Simple statistical includes n, min, Q1, median, mean, standard deviation, Q3, max.

work.adsl | SAFFL= "Y"

		Durg A (N=&BIG_1_N1) TRT01PN = 1	Durg B (N=&BIG_1_N2) TRT01PN = 2	Total (N=&BIG_1_N3) TRT01PN in (1, 2)
	Age			
work.adsl.trtdur.n	n	xx	xx	xx
work.adsl.trtdur.min	Min	xx	xx	xx
work.adsl.trtdur.q1	Q1	xx.x	xx.x	xx.x
work.adsl.trtdur.median	Medain	xx.x	xx.x	xx.x
work.adsl.trtdur.mean	Mean	xx.x	xx.x	xx.x
work.adsl.trtdur.std	SD	xx.xx	xx.xx	xx.xx
work.adsl.trtdur.q3	Q3	xx.x	xx.x	xx.x
work.adsl.trtdur.max	Max	xx	xx	xx

Figure 4. Computing Specific Descriptive Statistics

work.adsl | SAFFL= "Y"

		Durg A (N=&BIG_1_N1) TRT01PN = 1	Durg B (N=&BIG_1_N2) TRT01PN = 2	Total (N=&BIG_1_N3) TRT01PN in (1, 2)
	Age			
work.adsl.trtdur.n	n	xx	xx	xx
work.adsl.trtdur.min#max	Min, Max	xx, xx	xx, xx	xx, xx
work.adsl.trtdur.q1	Q1	xx.x	xx.x	xx.x
work.adsl.trtdur.median	Medain	xx.x	xx.x	xx.x
work.adsl.trtdur.mean	Mean	xx.x	xx.x	xx.x
work.adsl.trtdur.std	SD	xx.xx	xx.xx	xx.xx
work.adsl.trtdur.q3	Q3	xx.x	xx.x	xx.x

Figure 5. Re-arrangement of Descriptive Statistics

2.1.4 Any Combination of Above

work.adfaon.aval.n | paramcd = 'DIAGDY'	Time from diagnosis to start of study treatment (days)	n	xx
work.adfaon.aval.mean | paramcd = 'DIAGDY'		Mean	xx.x
work.adfaon.aval.std | paramcd = 'DIAGDY'		SD	xx.xx
work.adfaon.aval.median | paramcd = 'DIAGDY'		Median	xx.x
work.adfaon.aval.min | paramcd = 'DIAGDY'		Min	xx
work.adfaon.aval.max | paramcd = 'DIAGDY'		Max	xx
	Overall disease classification [a]		
work.adfaon | paramcd = 'FADXOCLS' and avalc="Metastatic"	Metastatic	n (%)	xx (xx.x)
work.adfaon | paramcd = 'FADXOCLS' and avalc="Locally advanced"	Locally advanced	n (%)	xx (xx.x)
work.adfaon | paramcd = 'FADXOCLS' and avalc=""	No evidence of disease	n (%)	xx (xx.x)

Figure 6. Combination of Above

2.1.5 Denominator Overwriting

The whole section shares the same denominator. By adding a property node named as single denominator, the percentage will be calculated based on the macro variable nobs_1, rather than BIG N.

```
<tr single_denominator = "&nobs_1">
    <td>work.adlb | param = "Param01" and btoxgrn = 0</td>   <td></td>   <td></td>
</tr>
```

work.adlb|saffl = "Y"

	Parameter (unit)	Group	Nobs	Baseline CTCAE grade	Worst on-treatment CTCAE grade: Grade 0 n (%) atoxgrn = 0	Grade 1 n (%) atoxgrn = 1	Grade 2 n (%) atoxgrn = 2	Grade 3 n (%) atoxgrn = 3	Grade 4 n (%) atoxgrn = 4
	Param01	Drug A N=&BIG_N							
work.adlb | param = "Param01" and btoxgrn = 0			&nobs_1	Grade 0	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adlb | param = "Param01" and btoxgrn = 1				Grade 1	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adlb | param = "Param01" and btoxgrn = 2				Grade 2	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adlb | param = "Param01" and btoxgrn = 3				Grade 3	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)
work.adlb | param = "Param01" and btoxgrn = 4				Grade 4	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)

Figure 7. Cross-table of ADLB with single denominator

2.2 Tags

Tokens	Mandatory	Singleton	Description
<pre></pre>	N	Y	Put your custom initiate code between the paired tags
	N	N	Put title between the paired tags
<tfoot></tfoot>	N	N	Put footnote between the paired tags
</br>	N	N	Break line
	N	N	Indentation
<table></table>	Y	Y	The <table> tag defines an HTML table.
<tr></tr>	Y	N	The <tr> tag defines a row in an HTML table.
<th></th>	Y	N	The <th> tag defines a header cell in an HTML table.
<td></td>	Y	N	The <td> tag defines a standard data cell in an HTML table.

2.3 Init from Clipboard

It is very likely that statisticians aren't very cooperated because editing HTML is not user friendly for most people. So, an alternative is provided in the system. If mock shells are delivered in excel file, then you can copy cells and click a button as indicated via following screenshot. After doing that, a draft HTML file will be created automatically. You can modify as what you want.

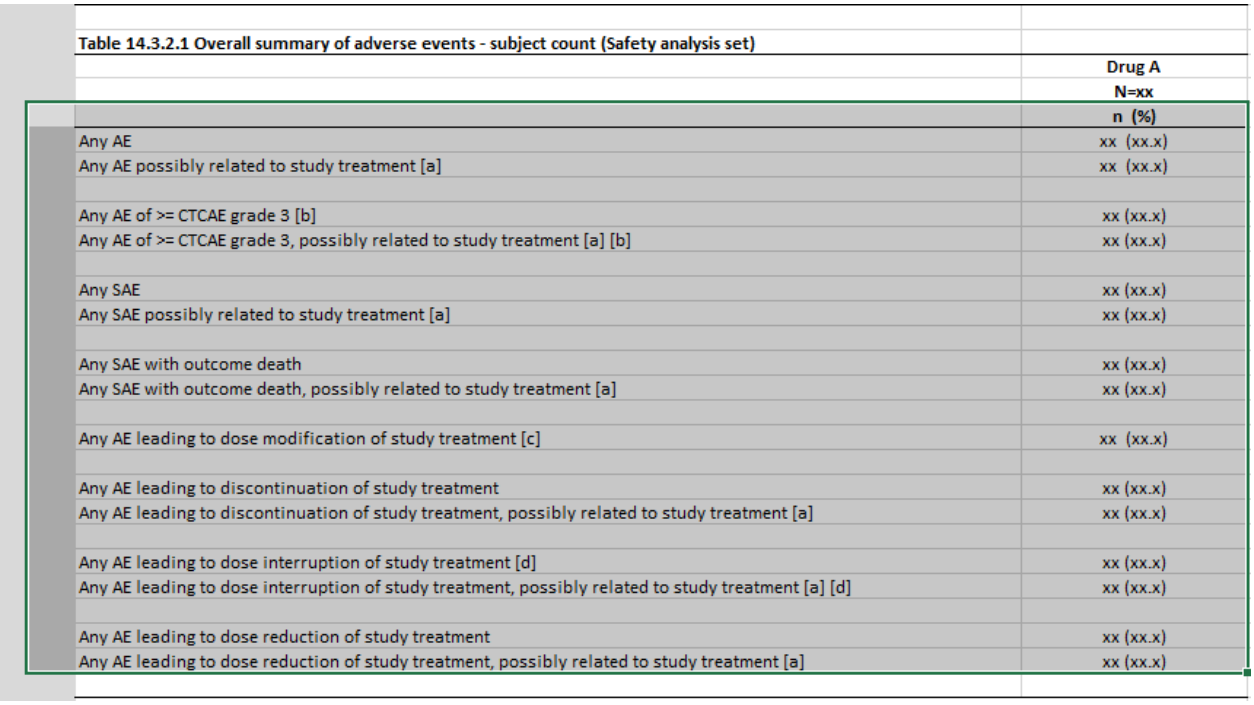


Table 14.3.2.1 Overall summary of adverse events - subject count (Safety analysis set)	
	Drug A
	N=xx
	n (%)
Any AE	xx (xx.x)
Any AE possibly related to study treatment [a]	xx (xx.x)
Any AE of >= CTCAE grade 3 [b]	xx (xx.x)
Any AE of >= CTCAE grade 3, possibly related to study treatment [a] [b]	xx (xx.x)
Any SAE	xx (xx.x)
Any SAE possibly related to study treatment [a]	xx (xx.x)
Any SAE with outcome death	xx (xx.x)
Any SAE with outcome death, possibly related to study treatment [a]	xx (xx.x)
Any AE leading to dose modification of study treatment [c]	xx (xx.x)
Any AE leading to discontinuation of study treatment	xx (xx.x)
Any AE leading to discontinuation of study treatment, possibly related to study treatment [a]	xx (xx.x)
Any AE leading to dose interruption of study treatment [d]	xx (xx.x)
Any AE leading to dose interruption of study treatment, possibly related to study treatment [a] [d]	xx (xx.x)
Any AE leading to dose reduction of study treatment	xx (xx.x)
Any AE leading to dose reduction of study treatment, possibly related to study treatment [a]	xx (xx.x)

Figure 8. Mock Shell of Adverse Events

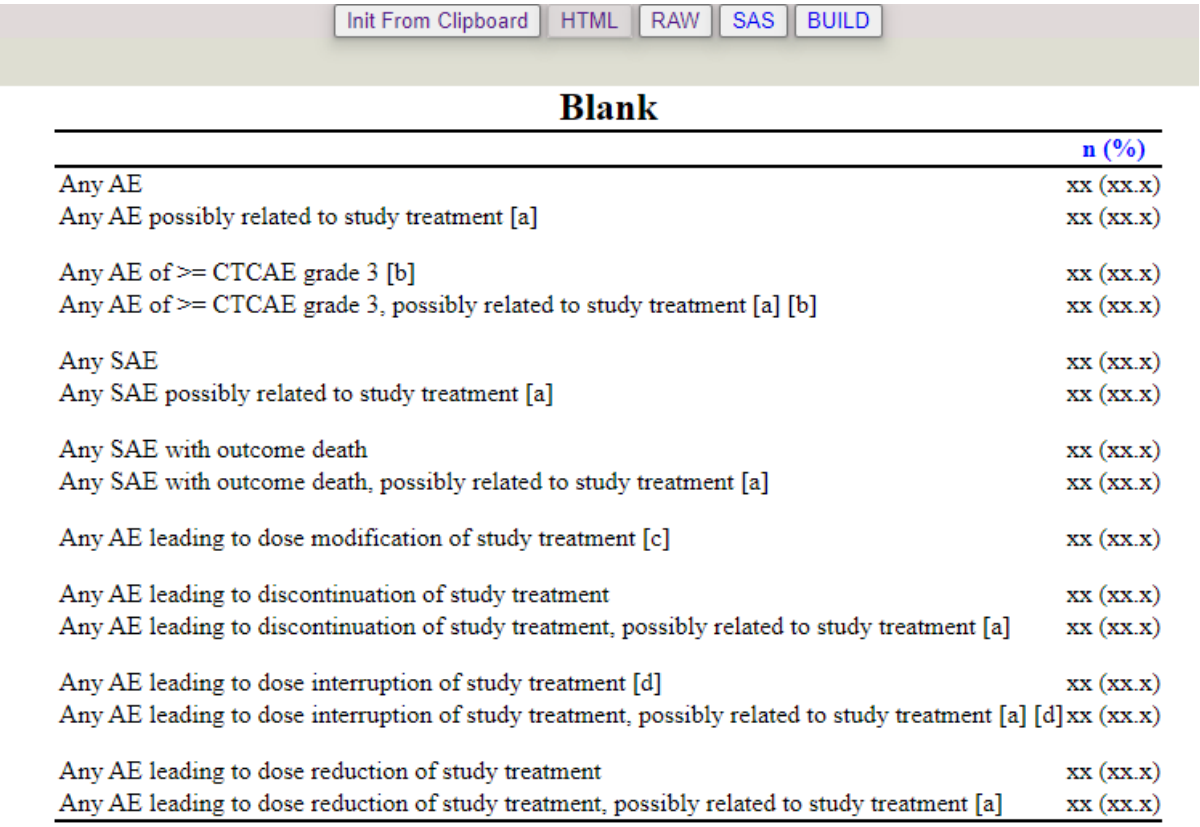


Figure 9. Rendered HTML Based on the Content on Clipboard

2.4 Sample Raw HTML

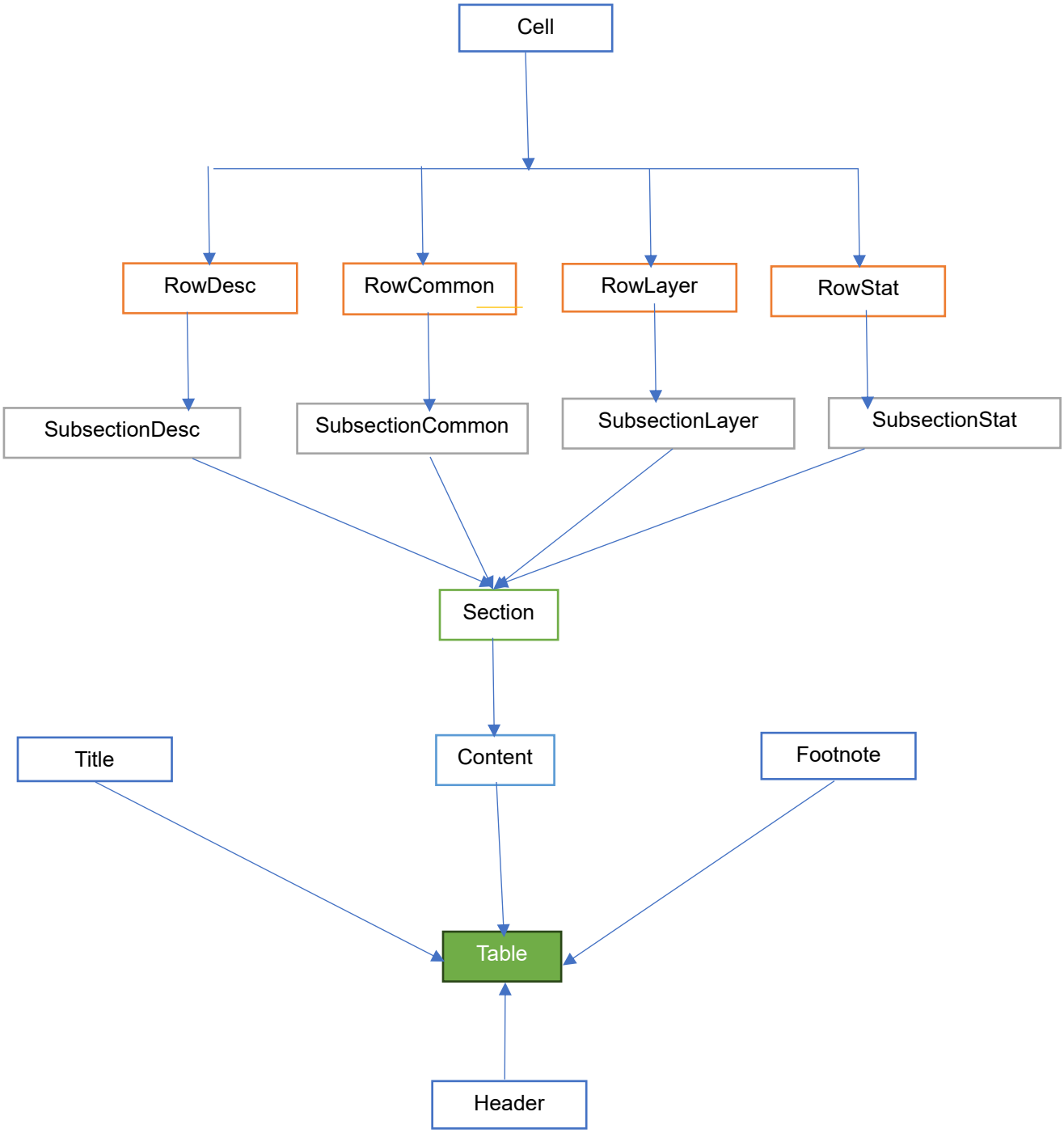
```
<pre>
/*Your Custom code here...*/

</pre>
<h2>Blank</h2>
<h2></h2>
<table>
<tr><th></th><th></th><th>n (%)</th></tr>
<tr><td></td><td>Any AE</td><td>xx (xx.x)</td></tr>
<tr><td></td><td>Any AE possibly related to study treatment [a]</td><td>xx  (xx.x)</td></tr>
</tr>
<tr class='section'><td></td><td>Any SAE</td><td>xx (xx.x)</td></tr>
<tr><td></td><td>Any SAE possibly related to study treatment [a]</td><td>xx (xx.x)</td></tr>
</tr>
</tr>
</table>
```

3. Dependencies

Category	Third Party Tool
Interface	Bootstrap
Interface	Tabulator
Interface	jQuery
Interface	jQuery-linedtextarea Plugin
Interface	Particles
Functionality	Python 3.0+
Functionality	Flask
Functionality	Beautiful Soup

4. Implementation Details

4.1 Class hierarchy



5. Challenges

In some cases, we need to display statistical horizontally, which still not be implemented.

Table 14.2.3.16.2 Event-free survival based on BICR assessments, evaluation-time bias sensitivity analysis (Modified intent-to-treat)

				Comparison between groups		
Group	N	Number (%) of subjects with events [a]	Mean (months) [b]	Hazard ratio [c]	95% CI [c]	2-sided p-value [d]
Arm A	xx	xx (xx.x)	xx.x	x.xx	x.xx, x.xx	x.xxx
Arm B	xx	xx (xx.x)	xx.x			

Figure 10. Efficacy Analysis Based on BICR PFS Endpoint

Efficacy tables and listings are also not covered well.

Requirements of figures cannot be fully described in the system.

CONCLUSION

By maintaining every material in the system, we can have good consistency with less effort. It saves programmers from some of the more tedious jobs and lets them focus on more skilled work. The system may not be a good implementation, but it proves that the schema is feasible and practical.

CONTACT INFORMATION

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

Weidong Liang
Phastar Inc.
weidong.liang@phastar.com