Alternative Approaches to Creating Disposition Flow Diagrams

Brian Fairfield-Carter,
ICON Clinical Research,
Redwood City, CA
What does ‘disposition’ mean?

- “Predominant tendency”; “propensity”; “final settlement of a matter”
- Screening
- Enrollment
- Randomization
- Completion
- Early discontinuation
  - AE
  - Lack of efficacy
  - Etc.
Why summarize disposition?

- Check for unbalanced groups
- Effectiveness of recruitment procedures (number of screen failures, etc.)
- Logistical, safety, perception issues.
## Disposition Summary: Tabular...

### Table 14.1.1.2

*Subject Disposition*

<table>
<thead>
<tr>
<th>Randomized Set</th>
<th>Placebo ((N=XXX))</th>
<th>2 mg ((N=XXX))</th>
<th>10 mg ((N=XXX))</th>
<th>50 mg ((N=XXX))</th>
<th>Total ((N=XXX))</th>
<th>P-value ((p))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enrolled Set</td>
<td>(n) ((%))</td>
<td>(n) ((%))</td>
<td>(n) ((%))</td>
<td>(n) ((%))</td>
<td>(n) ((%))</td>
<td></td>
</tr>
<tr>
<td>Enrolled but not treated</td>
<td>(XX) ((XX.X))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Randomized Set</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td></td>
</tr>
<tr>
<td>Full Analysis Set</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td></td>
</tr>
<tr>
<td>Per Protocol Set</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td></td>
</tr>
<tr>
<td>Subjects Completing Treatment ((a))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td></td>
</tr>
<tr>
<td>Subjects Discontinuing Treatment</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td></td>
</tr>
<tr>
<td>Reason for Discontinuing Treatment</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Adverse Event</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Lost to Follow-up</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Death</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Protocol Violation</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Subject Decision</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Physician Decision</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Sponsor Decision</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Subject’s Physician Decision</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
<tr>
<td>Clinical Endpoint ((ESRD))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(XX) ((XX.X))</td>
<td>(0.XXXX)</td>
</tr>
</tbody>
</table>
...versus flow diagram
Trade-offs

- Flow diagram *may* be easier to interpret (debatable)...

- ...but more complicated and time-consuming to produce (in fact, just how *do* you program a flow chart?)
The Institute of Imaginary SAS Procedures
Purpose:

Present three alternative methods for disposition flow diagrams.

- Annotate (purely SAS-based)
- Annotate/Excel/Visual Basic (hybrid)
- Word template/script-driven transcription (hybrid)
Keeping in mind...

- Validation implications
- Time/effort/cost
- Flexibility
- ‘Interoperability’ (operating system, programming environment, programmer knowledge)
The Annotate Facility

- SAS/Graph ‘drawing tool’
  - Objects (line, rectangle, text)
  - Attributes (color, font)
  - Functions (move, draw)
- Verbose (but consider Annotate macros (%line, %label, etc.))
- A flow diagram is just a collection of objects & attributes...
Annotate

Screened
N=xxx

Randomized
N=xxx

Not Randomized
N=xxx

%rect(x,y,width,height,...);
%label(x,y,text,...);
%line(x1,y1,x2,y2,...);
%line(x1,y1,x2,y2,...);
%line(x1,y1,x2,y2,...);
%line(x1,y1,x2,y2,...);
%rect(x,y,width,height,...);
%label(x,y,text,...);
%rect(x,y,width,height,...);
%label(x,y,text,...);
%annomac;
%macro add_element(x=,y=,width=,height=,text=);

%rect(&x,&y,&x+&width,&y+&height,black,1,0.2);

%let i=1;
%do %until (%qscan(&text,&i,'|')=);

%label(&x+0.5,&y+&height-&i,
"%qscan(&text,&i','|'),
BLACK,0,0,1,SIMPLEX,6);

%let i=%eval(&i+1);

%end;

%mend add_element;
Annotate

data final;
   length function color style $8. text $60.;
   retain xsys '6' ysys '6' hsys '6' when 'a'
   line 1 function 'label';
   %add_element(x=20,y=35,width=12,height=3,
                    text=%str(  Entered |  N=xxx));
   %add_element(x=5,  y=27,width=12,height=3,
                    text=%str(  Randomized|  N=xxx));
   ...(etc.)...
   %line(26,35,  26,32.5,BLACK,1,1);
   %line(10,32.5,52,32.5,BLACK,1,1);
   ...(etc.)...
### Annotate

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>COLOR</th>
<th>STYLE</th>
<th>TEXT</th>
<th>XSYS</th>
<th>YSYS</th>
<th>HSYS</th>
<th>WHEN</th>
<th>LINE</th>
<th>SIZE</th>
<th>X</th>
<th>Y</th>
<th>ANGLE</th>
<th>ROTATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLYLINE</td>
<td>black</td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>0.2</td>
<td>0.0</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>POLYCONT</td>
<td>black</td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>0.2</td>
<td>12.0</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>POLYCONT</td>
<td>black</td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>0.2</td>
<td>12.0</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>POLYCONT</td>
<td>black</td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>0.2</td>
<td>0.0</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>POLYCONT</td>
<td>black</td>
<td></td>
<td></td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>0.2</td>
<td>0.0</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LABEL</td>
<td>BLACK</td>
<td>SIMPLEX</td>
<td>Entered</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>1.0</td>
<td>0.5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LABEL</td>
<td>BLACK</td>
<td>SIMPLEX</td>
<td>N=xxx</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>a</td>
<td>1</td>
<td>1.0</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Annotate

ods rtf file="diagram.rtf";

   proc gslide annotate=final;
   run;
   quit;

ods rtf close;
Entered
N=xxx

Randomized
N=xxx

Not Randomized
N=xxx
Reasons Not Randomized
- Protocol Entry Criteria Not Met n=xxx
- Personal Conflict or Other Patient Decision n=xxx
- Physician Decision n=xxx
- Unable to Contact Patient (Lost to Follow-Up) n=xxx
- Sponsor Decision n=xxx
- Adverse Event n=xxx

Placebo
N=xxx

Treatment
N=xxx
This is ok, but...

- ‘Dynamic’ positioning of elements? (see instead Saradha & Veeravel (2008))
- Output file is ‘static’
- Increase in diagram complexity = increase in code size/complexity...
‘Hybrid’ approach #1:

- Use Annotate to create rectangle/text elements (as separate graphics files)
- VBA macro or VB script to import, position, and link elements in Excel
Annotate/Excel/VB

SAS

`Screened
N=xxx`

VBScript

`%rect(x,y,width,height,...);
%label(x,y,text,...);`

`Import, position, link`

Excel

```
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enter</td>
<td>N=xxx</td>
</tr>
</tbody>
</table>
```

- `Screened N=xxx` in SAS
- `%rect(x,y,width,height,...);` and `%label(x,y,text,...);` in VBScript
- `Import, position, link` in VBScript
- Excel sheet with `Entered N=xxx`
Annotate/Excel/VB

... 

Set objXLS = WScript.CreateObject("Excel.Application.11")

...

with objXLS

.Range("B2").Select

ActiveSheet.Pictures.Insert(f & "\1.emf").Select

...(etc.)...

AddLine(76.5, 46.5, 76.5, 76.5).Select
Annotate/Excel/VB

Entered
N=xxx

Randomized
N=xxx

Not Randomized
N=xxx

Reasons Not Randomized
- Protocol Entry Criteria Not Met n=xxx
- Personal Conflict or Other Patient Decision n=xxx
- Physician Decision n=xxx
- Unable to Contact Patient (Lost to Follow-Up) n=xxx
- Sponsor Decision n=xxx
- Adverse Event n=xxx

Placebo
N=xxx

Treatment
N=xxx
Annotate/Excel/VB

- This is OK, but...
  - Determining target cells for placing rectangle/text elements is a pain
  - Determining coordinates for connecting lines is a pain
- Generate code automatically by recording a VBA macro while importing, positioning and linking.
‘Hybrid’ approach #2

- Calculate values in SAS, and output to a text file
- Use the flow-diagram mock-up as a template, replacing ‘N=xxx’ placeholders with the actual values
Word template, transcription
data _null_;  
    set disposition;  
    file "replacevalue.txt";  
    put "VAR"||put(_n_,best.)||""||""||put(count,best.);  
run;

VAR1 | 200  
VAR2 | 180  
VAR3 |  20  
VAR4 | 150  
VAR5 |  30  
---(etc.)---
Word template, transcription

- *Script*: read the varname/value list, and do an iterative search/replace in Word...
Open the mockup in Word

Open the text file, read the first line,

(Grab ‘VAR1’ as the search-string, and ‘200’ as the replace-string)

Execute the search-replace

Repeat...

Save the final product
Figure 14.1.1
Patient Disposition

- Enrolled: N=180
- Randomized: N=150
- Entered: N=200
- Non-Enrolled: N=20
- Non-Randomized: N=30
In Summary...

- Annotate – avoids ‘hybrid’ solution, is platform-independent, but may require a lot of code
In Summary...

- Annotate/Excel/VB – reduces coding overhead, but sacrifices platform independence.
- Word template/transcription - demands a well-developed template but little code; sacrifices platform independence.

fairfieldcarterbrian@gmail.com