phuse·connect
connect·share·advance
5-8 November
ICC Birmingham
EU 2023
The Clinical Data
Science Conference

Paper DH11

Challenges in Oncology Modular studies: The Fortrea approach

Francesca Dall'Acqua, Fortrea, Milan, Italy
Dubravka Georgijevic, Fortrea, Leiden, the Netherlands

ABSTRACT

An Umbrella study is a type of master protocol that is defined as "a unifying study design that includes multiple subgroups and sub-studies, with patients having same or different diseases and that employ one or multiple drugs to treat it."[1]. Such master protocol allows for testing multiple agents simultaneously and may include specified modifications while the trial is ongoing.

Several challenges are encountered with such designs: multiple deliveries with different purposes and scope, analysis outputs which may differ across modules, analysis of multiple modules simultaneously, a database that needs to be analyzed at a particular cut-off date and for only a particular module, preparation of final analysis for a specific module in an ongoing environment (as modules are completed at different times), modifications to the protocol – dealing with its implications, new module(s) introducing CRF change(s) requiring new analysis to be applied retrospectively to previous modules close to their delivery date.

Adapting programs to deal with these changes can make them very complex. How can this be handled, keeping consistency across modules and avoiding re-work? This presentation will share our experiences and solutions.

INTRODUCTION

At Fortrea, in the last years, we have had to deal more and more frequently with programming activities for umbrella studies which consist of many small sub-trials (or modules) to test multiple targeted therapies simultaneously for a single disease, in one large trial .

These studies operate under a master protocol, that allows for the inclusion of additional treatment arms or subgroups as new interventions become available and often employ an adaptive design, which allows for modifications in treatment arms based on ongoing data analysis and emerging information about treatment efficacy and safety.

Umbrella studies are characterized by the utilization of complex designs, made more arduous from the continually evolving nature and incremental of the trial, in a long period of time.

The scope of this paper is to present strategies we have adopted in Fortrea in order to make the analysis and reporting for such trials more efficient, presented through approaches adopted for real studies, bringing real examples from some of them.

As previously stated, one of the characteristics of these clinical studies is the long duration. As an example, one of the trials under investigation had First Subject In (FSI) Q4 2018, while Last Subject Last Visit (LSLV) is planned in Q3 2024. In the meantime, there have been – so far - 13 amendments of the Clinical Study Protocol (CSP), in which the initial four modules with two Investigational Products (IPs) have increased to fourteen modules with seven different IPs and twenty different study designs. These have been translated into eighteen different version of the Case Report form (CRF) and six different versions of the Statistical Analysis Plan (SAP) and related Tables, Figures and Listings (TFL) Mock Shells.

In terms of the database: at the database go live there were 98 raw datasets present and that number has more than doubled during the five years (so far) of the trial. One of the sponsor requirements is for SDTM datasets to be transferred daily to the sponsor's server, containing all the data present in the database (to facilitate safety monitoring).

This study presents characteristics common to all other umbrella studies we are facing in Fortrea: longer trial timelines, higher number of substantial protocol amendments and increased data requirements.

To manage the complexity of these trials, which increase in size and importance at a faster and faster rate, there is no one-size-fits-all solution, but instead a range of strategies must be put in place. Below we will describe some of the approaches used across these studies.

COPING STRATEGIES

CENTRALIZED PROGRAMMING

The most common strategy in place is to centralize, in each program, the logic for all modules present in the umbrella study, as they are added in the protocol.

This meets, at least for SDTM, the requirement to send to the sponsor daily transfer of the current version of SDTM data created from the data extracted from the database every night.

The centralized programming approach has been extended to ADaMs and TFLs, where the singular program implements derivations and layout for all modules as per the latest version of SAP and TFL Mock Shells.

For ADaM, this means that in each dataset program, baselines, analysis visit windows, exposure duration and imputations that can change from module to module depending on the design and schedules of assessments are implemented.

Regarding TFLs, the programs must include the logic to allow the reporting of all the differences due to the module specifications. The same table, depending on the module(s) to report, can contain differences in column headers, footnotes and also in the content. It can change depending on the module design and modules representation (single module or combination of modules on the same table), but also on the number of IPs administered.

For example, the “Table of Disposition of Subjects” can report in the same line, depending on whether randomization is used in the module or not, either “Subjects assigned to treatment” or “Subjects randomized”.

As well, depending on whether the therapy is mono or in combination, the “Table of Adverse events in any category” can report the relations with the main treatment only, or also the relation with second IP and/or the combination of them, resulting in a different number of records to report.

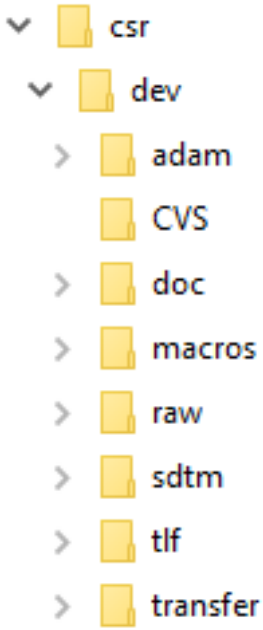

Figure 1 Structure of the Centralized Development Area

SUB-FOLDERS FOR ONGOING STUDY DELIVERIES

Centralized programs are developed and maintained in one central development folder within the study area in our SAS® environment (called csr/dev). Specific sub-folders are created for deliveries needed while the study is ongoing (dry-run, interim analysis, customized review (e.g., safety), clinical study report).

These delivery-dedicated subfolders contain the data needed for the delivery and programs copied from the central development folder. The best practice is to perform program updates only in the central development area, so that programs in delivery folders represent a picture of the version in use at the time of the delivery. There are, however, some circumstances in which programs get updated in the delivery areas as well:

- in some exceptional cases, due to critical delivery timelines, some ad-hoc changes can be made in the delivery folder.
- when multiple deliveries have overlapping timelines but need different data selection and/or different derivations and/or different layouts of the same output (in this case programs are prepared in separate delivery areas).
- in the case of a modular delivery, which must be re-run after a re-lock of the database for the module, when a new version of the programs needs to reflect the latest version of the database.

The program changes performed in specific delivery folders must be included in the corresponding programs in the central development area as soon as possible, making sure that the change applies only to the relevant module(s).

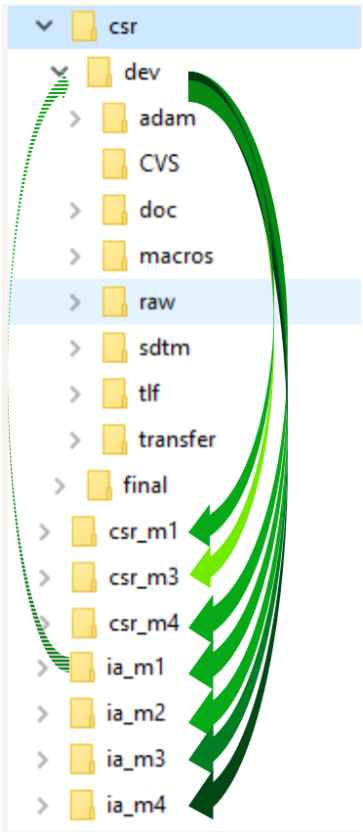

Figure 2 csr/dev contains centralized programs. At the time of the delivery, their last version is copied to the delivery folder.

CENTRALIZED DATA SELECTION

For ongoing study deliveries (dry-run, interim analysis, customized review (e.g., safety), clinical study report), it is common practice to analyze data snapshots according to a particular date i.e., the data cut-off date (DCO date) for a formal analysis. In the umbrella modular studies, typically analysis is performed on one study module only, or sometimes a combination of more than one module. Therefore, there is a need for each delivery to select data up to the DCO date, containing the desired selection of subjects based on the study modules they belong to.

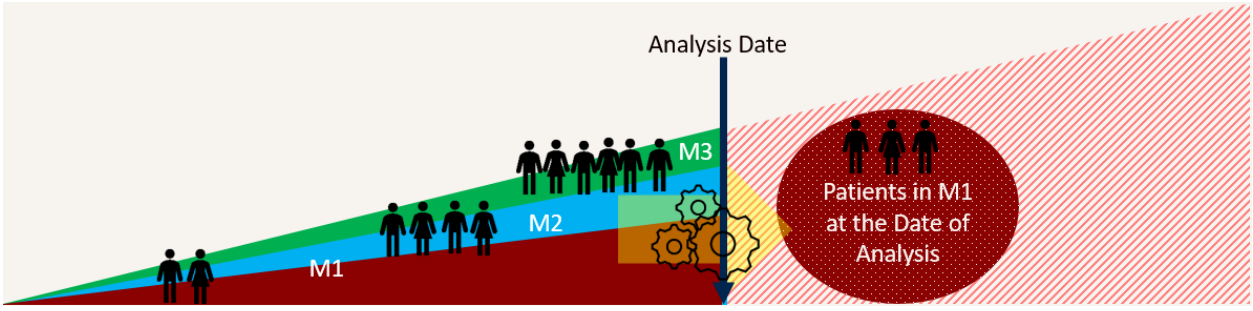

Figure 3 Centralized Data Selection

Our preferred option is to perform the data selection at the raw data level, in one central program, prior to creating SDTM datasets, so that SDTM datasets, ADaM datasets and transport files will contain the same version of data.
In this way, even if the programs contain the logic applicable to all modules, the raw datasets, SDTMs, ADaMs and related transport files, will contain only the data related to the module(s) of interest for that specific delivery.

Referring to Figure 3, on the DCO date for the Clinical Study report of the first module, in the database patients from Modules 2 and 3 were also present. The processed raw datasets resulting from the DCO program contained only the patients enrolled to Module 1.

TABLES, FIGURES AND LISTINGS FLEXIBILITIES

Our goal is to develop centralized output programs that contain the logic applicable to all modules. At the TFLs level, centralized programming is implemented through an extensive use of macros that grant flexibility in customizing columns, analysis categories (e.g., "Subjects assigned to treatment" versus "Subjects randomized") and footnotes which might differ from module to module.

As an example, because modules investigate different study drugs in possibly different subject groups, tables need to present different treatment columns between the modules:

Module 1

	Number (%) of subjects					
	Group 1.A (N = xxx)	Group 1.B (N = xxx)	Group 1.C (N = xxx)	Group 1.D (N = xxx)	Group 1.E (N = xxx)	Total (N = xxx)
Subjects assigned to treatment	xx (x.x)	xx (x.x)	xx (x.x)	xx (x.x)	xx (x.x)	xx (x.x)

Module 2

	Number (%) of subjects					
	Stratification Group 1				Stratification Group 2	
	Group 2.A (N = xxx)	Group 2.B (N = xxx)	Group 2.C (N = xxx)	Total (N = xxx)	Group 2.D (N = xxx)	Overall Total (N = xxx)
Subjects randomised	xx (x.x)	xxx (x.x)	x (x.x)	xx (x.x)	xx (x.x)	xx (x.x)

Modules 4+5

	Number (%) of subjects					
	Module 4					Module 5
	Group 4.A (N = xxx)	Group 4.B (N = xxx)	Group 4.C (N = xxx)	Group 4.D (N = xxx)	Total (N = xxx)	Group 5.A (N = xxx)
Subjects assigned to treatment/randomised	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)	xx (xx.x)

To be able to use the same table program, we create macros that can be called for a specific module and that derive applicable treatment groups per module, analysis set and output (%tlf_reptrt), number of subjects in each treatment group (%tlf_denom), list of treatment columns to report (%tlf_repcol) and report define statements for treatment groups (%tlf_repcol).

```
%macro by_module(out= , module = );
   %tlf_reptrt(dsin=xxx, pop=&popfl, module=&module);
   %tlf_denom(pop=&popfl, module=&module);
   …
   proc report data = xxx_pop;
      column a b c %tlf_repcol(txt="Number (%) of subjects");
      %tlf_repdef(width=16, col_on_page=4);
   run;
%mend by_module;
```

For the content of TFLs, one of the challenges we have is to select the study modules/treatment groups applicable for the delivery in question and applicable to the analysis set used in each output. We have a requirement to display them all, even in outputs where there is no data for a module (e.g., in an output for AE of special interest, it can happen that for a module no subject experienced the AE), in which case an empty page will be produced with the message "No data to report" for that module.

There are a few approaches used:

1. **Data driven:** As we apply subject selection at raw data level, SDTM and ADaM datasets contain only the applicable subjects/study modules. The ADaM dataset ADSL, which contains all subjects relevant for the delivery, is used in each TFL program to get the list of modules applicable for the analysis set in question, and then the further TFL code is executed for those modules only.

```
proc sql noprint;
   create table allmodules as
      select distinct group01c, group01
```

```
    from adam.adsl
         where SAFFL = 'Y'
    order by group01c;
quit;
data _null_;
   set allmodules;
   file "&work_lib/create_report.sas";
   put '%by_module(out=' group01c ', module=' group01c ');';
run;
%inc "&work_lib/create_report.sas";
```

2. **Metadata driven:** In this second approach modules are defined in a central Microsoft ® Excel ® file and that information is read into SAS to create a SAS macro variable that is used in each TLF program. This method allows some flexibility if modules must be combined in different ways into the outputs.
   In the example below the singular tables will report modules in combination M1+M2 and M3+M4 (see Figure 5), while the singular listings will report the content of modules M1, M2, M3 and M4 (Figure 6).

macro_var_name	macro_var_value
out_parts	M1 M2 M3 M4
out_parts2	M1+M2$M3+M4$

Figure 4 Macro variable definition in the Excel file

Page 1 of 2

Table xx.x.x

Title

Module 1+2

	Module 1					Module 2			
	GROUP 1.A N=xxx		GROUP 1.B N=xxx		Total N=xxx		GROUP 2.A N=xxx		Total N=xxx	
	n	(%)	n	(%)	n	(%)	n	(%)	n	(%)
Subjects who completed treatment	xxx	(xx.x)	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Subjects who discontinued treatment due to global/country/area situation	x	(x.x)	xx	(xx.x)	x	(x.x)	x	(x.x)	x	(x.x)
<<Reason1>>	x	(x.x)	xx	(xx.x)	x	(x.x)	x	(x.x)	x	(x.x)
<<Reason2>>	x	(x.x)	0		x	(x.x)	x	(x.x)	x	(x.x)
…										

Page 2 of 2

Table xx.x.x

Title

Module 3+4

	Module 3					Module 4			
	GROUP 3.A N=xxx		GROUP 3.B N=xxx		Total N=xxx		GROUP 4.A N=xxx		Total N=xxx	
	n	(%)	n	(%)	n	(%)	n	(%)	n	(%)
Subjects who completed treatment	xxx	(xx.x)	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)	xx	(xx.x)
Subjects who discontinued treatment due to global/country/area situation	x	(x.x)	xx	(xx.x)	x	(x.x)	x	(x.x)	x	(x.x)
<<Reason1>>	x	(x.x)	xx	(xx.x)	x	(x.x)	x	(x.x)	x	(x.x)
<<Reason2>>	x	(x.x)	0		x	(x.x)	x	(x.x)	x	(x.x)

Figure 5 Example of how Tables will appear as per macro variable definition in the Excel file provided in Figure 4

Appendix xx.x.x.x
Title

Module 1

Subject identifier	Preferred term (MedDRA version xx.x)	Start date (Study day at start)	End date (Study day at end)	Duration (<<unit>>)	Ongoing at time of enrollment
E0001011	Xxxxxxxxx	2021-04 (-500*)	2022-03 (-136*)	NC	N
	Xxxxxxxxx	2022-03-17 (-150)		NC	Y
	Xxxxxxxxx	2022-03-27 (-140)	2022-06-25 (-50)	90	N
	Xxxxxxxxx	2022-07-10 (-35)	2022-09-02 (20)	55	Y
E0001012	Xxxxxxxxx	2022-07-10 (-35)	2022-09 (48*)	NC	Y
	Xxxxxxxxx	2022-07-15 (-30)	2022-07-20 (-25)	6	Y

Appendix xx.x.x.x
Title

Module 2

Subject identifier	Preferred term (MedDRA version xx.x)	Start date (Study day at start)	End date (Study day at end)	Duration (<<unit>>)	Ongoing at time of enrollment
E0001021	Xxxxxxxxx	2021-04 (-500*)	2022-03 (-136*)	NC	N
	Xxxxxxxxx	2022-03-17 (-150)		NC	Y
	Xxxxxxxxx	2022-03-27 (-140)	2022-06-25 (-50)	90	N
	Xxxxxxxxx	2022-07-10 (-35)	2022-09-02 (20)	55	Y
E0001022	Xxxxxxxxx	2022-07-10 (-35)	2022-09 (48*)	NC	Y
	Xxxxxxxxx	2022-07-15 (-30)	2022-07-20 (-25)	6	Y

Appendix xx.x.x.x
Title

Module 3

identifier	(MedDRA version xx.x)	at start)	end)	Duration (<<unit>>)	enrollment
E0001031	Xxxxxxxxx	2021-04 (-500*)	2022-03 (-136*)	NC	N
	Xxxxxxxxx	2022-03-17 (-150)		NC	Y
	Xxxxxxxxx	2022-03-27 (-140)	2022-06-25 (-50)	90	N
	Xxxxxxxxx	2022-07-10 (-35)	2022-09-02 (20)	55	Y
E0001022	Xxxxxxxxx	2022-07-10 (-35)	2022-09 (48*)	NC	Y
	Xxxxxxxxx	2022-07-15 (-30)	2022-07-20 (-25)	6	Y

Appendix xx.x.x.x
Title

Module 4

Subject identifier	Preferred term (MedDRA version xx.x)	Start date (Study day at start)	End date (Study day at end)	Duration (<<unit>>)	Ongoing at time of enrollment
E0001041	Xxxxxxxxx	2021-04 (-500*)	2022-03 (-136*)	NC	N
	Xxxxxxxxx	2022-03-17 (-150)		NC	Y
	Xxxxxxxxx	2022-03-27 (-140)	2022-06-25 (-50)	90	N
	Xxxxxxxxx	2022-07-10 (-35)	2022-09-02 (20)	55	Y
E0001042	Xxxxxxxxx	2022-07-10 (-35)	2022-09 (48*)	NC	Y

Figure 6 Example of how Listings will appear as per macro variable definition in the Excel file provided in Figure 4

3. **Treatment group based metadata driven:** The third is a variant of the second, metadata driven as well, but based on the list of treatment groups present in the TRT codelist in the ADaM specs file, which gets adjusted to each delivery, and read into a SAS macro variable.

ADaM Specs in Central Development Area			
ID	Name	Term	Decoded
TRT	Treatment	M1	Module 1
TRT	Treatment	M2	Module 2
TRT	Treatment	M3	Module 3

ADaM Specs in Delivery Folder for M1			
ID	Name	Term	Decoded
TRT	Treatment	M1	Module 1

CONCLUSION

The centralized programming approach obviously has some disadvantages: some of the programs are very large and complex since they must handle different study designs. For example, for ADaMs - different baseline, analysis windows, and exposure duration definitions, while for TFLs different column headers and/or different content.

On the other perspective, this approach has the big advantage of implementing in one place the latest version of the program, which is definitely easier to maintain than having disseminated different versions of the same program which we need to then keep consistent.

The other advantage is that this approach permits flexibility when outputs for combined modules are needed. We have summarized in the table below some pros and cons of our approach:

PROS	CONS
• Meets the requirement of SDTM daily transfer containing all the data collected in the database	• Programs are very large in order to implement complex logic (i.e. more study designs that require different Analysis Visit Windows, Baseline definitions, different header blocks or layout for Table)
• Centralized code allows us to implement in one place the latest version of the program and ensures it is easier to maintain different disseminated versions of the same program.	• If any change in the middle of the process (SDTM, ADaM or TLFs) you have to check backwards and forwards to evaluate whether this has an impact on other modules and programs.
• Allows the development of a single program for simultaneous and parallel deliveries for different modules.	• Stability in the team is very important. Otherwise, a longer onboarding period needs to be planned, due to the high complexity of the programs
• In term of resources, one dedicated team is enough versus a team per module	

REFERENCES

[1] Bogin V. Master protocols: New directions in drug discovery. Contemp Clin Trials Commun. 2020 Apr 25;18:100568. doi: 10.1016/j.conctc.2020.100568. PMID: 32395664; PMCID: PMC7205752.

ACKNOWLEDGMENTS

We would like to express our warm gratitude to Carla Santillan for taking the time to review this paper and provide helpful feedback and for precious tips to the presentation.

CONTACT INFORMATION

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

Francesca Dall'Acqua
Fortrea Development Ltd
Via Giorgio Washington, 70,
20146 Milano MI, Italia
francesca.dallacqua@fortrea.com

Dubravka Georgijevic
Fortrea Netherlands BV
Schipholweg, 111 2316 XC
Leiden, The Netherlands
dubravka.georgijevic@fortrea.com