Construction of an ADaM-compliant data set for time-to-event analyses. Reexamining an approach by Bajamonte

Preamble

Author: Johannes Hüsing, KKS Heidelberg

Version: 0.1 as of 2009-09-22

Prerequisites

In 2003, Bajamonte suggested an approach along the ADaM model to convert a study data structure into an analysis-ready data set appropriate for time-to-event analysis. Since then, the AdaM standard has evolved: The Implementation Guide is at version 1.0 instead of hearsay version as in 2003, and some notions of what to expect from the source data sets have evolved. The following text is an outline of how to approach the problem outlined in the article from a 2009 perspective. This text is based on a discussion between Edelbert Arnold, Elke Sennewald and Johannes Hüsing on the German CDISC User Group Meeting on 2009-09-22.

As time-to-event analysis is in the focus of the current text, the term "event" will be used to describe a state-change in time, not necessarily unscheduled as in the SDTM interpretation.

Analysis tools are assumed to be more or less like SAS, but the usage of SAS is not implied. Users of different analysis tools are requested to interpret phrases like "one PROC step away" in the vernacular of their preferred software.

Relevant Documents

The article in question is referred to as Bajamonte (2003). The ADaM standard is referred to as presented in the documents "ADaM Implementation Guide" (ADam IG), version 1.0 as of 2008-05-30, and "Analysis Data Model" (ADaM 2.0) in version 2.0 as of 2006-02-15.

General Requirements

The source data in Bajamonte (2003) seem rather idealized, as a censoring status flag is already attached to the source data, and all types of events are already aggregated into one single file. We think that this outline sidesteps a main problem which arises with aggregating data coming from different domains (see the specific requirements from the example at the section titled " Study-specific requirements"), and therefore one of the key concepts in ADaM, i. e. less semantics in the contents than in the SDTM model. Also, it is safer to assume that the source data will be distributed over different SDTM domains, as many members interested in ADaM will have some experience with SDTM. SDTM is not required as structure for the input data according to ADaM IG, but is often mentioned as an example for input data.

The „One PROC step away" mantra is interpreted as the data being in a pre-analysis procedure step stage. We think of PROC REPORT, TABULATE, PRINT and the like as presenting rather than analyzing steps. There is, of course, room for interpretation as PROC TABULATE may include summarizing derivations. Therefore, we generally don't expect variables being the result of analysis steps in the analysis data sets.

Time-to-event analysis data sets should provide enough information to enable the user to do complex analyses such as multiple events, competing risks, time-dependent covariates, left-truncation, interval- and right-censoring. The example given does not require all of this information, but the generated data sets should be sufficiently generic.

There is some leeway between using more intermediate tables with simpler derivation steps or using more complex derivations in fewer steps. We choose the first approach, as the second approach can be more easily derived from it.

Study-specific requirements

The requirements are outlined in Bajamonte (2003) as excerpts from the study protocol and the statistical analysis plan. The protocol of a two-armed parallel-group trial is typical of oncological trials with different endpoints, such as tumor progression, death, or treatment failure. Endpoints may be elementary or composite. Specifically, treatment failure is defined as time to documented disease progression, death on study, start of different anti-cancer treatment regimen, or withdrawal due to toxicity, whatever comes first. Results required are tables comparing two treatments with respect to time to either event, expressed in median times with 95% confidence intervals in either treatment arm and as a relative risk estimate with 95% confidence interval.

General approach

The general approach is roughly visualized in Figure 1. First, the information on different endpoint relevant or censoring events will have to be collected from different domain tables and combined into one table containing all findings and events relevant to the time-to-event modelling of the different endpoints. Then, the table is aggregated into a table containing all subject-specific endpoints. From this table, the analyses can be generated (using, for example, the SAS procedure LIFETEST).

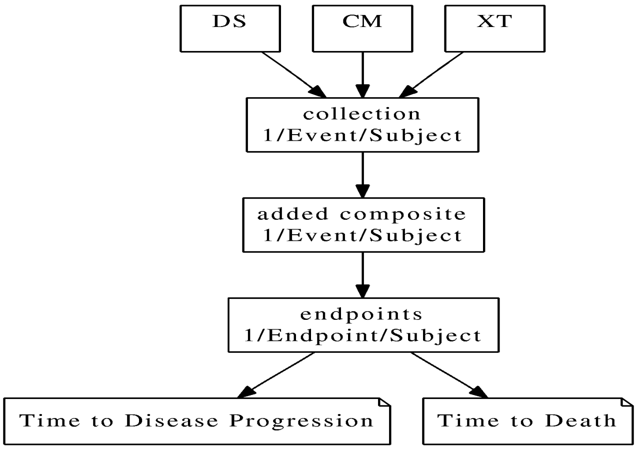


Figure 1: Workflow from source to report elements

Collecting the data from different tables

The relevant data are likely to be found in three different domains. The event “death” or “termination because of toxicity” can be attributed to the Disposition (DS) domain. The event “other chemotherapy” can be derived from the Concomitant Medication (CM) domain. For simplicity, it is assumed that there are CMCAT variable codes which directly correspond to tumor-specific chemotherapy outside the study treatments. The event “tumor progression” may require a sponsor-specific domain (abbreviated “XT” here). A possible censoring event is “loss to follow-up”, also found in the DS domain. All domains contain information on the time since the reference time point of the subject in the trial.

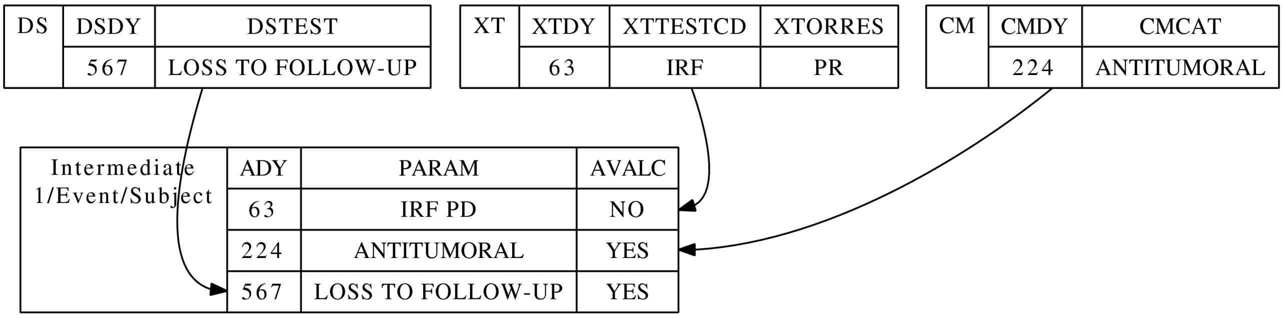
There has been discussion about whether the domain tables can be expected to be pure SDTM tables or their ADaM counterparts. The general consensus was to have as few transformation steps as possible from source to each report table, but above all to avoid redundant derivations. For instance, if the information on antitumoral therapy cannot be derived directly from the SDTM, but from a “forbidden CM” category derived in an ADaM domain table, the latter will be used as a source.

All the recorded events will be combined into one single table following the basic data structure (see ADaM IG, 2.2.2) and sorted by subject identifier and time. The timing variables will be derived into a single variable named ADY, the generic name for the timing variable. Traceability will be achieved using the source domain table name and the sequence number from the source tables.

The type of event that was assessed in the current record is described in PARAM, for instance IRF PROGRESSIVE DISEASE, LOSS TO FOLLOW-UP, TERMINATION FOR TOXICITY. The variable AVALC can assume the values YES, NO, or UNKNOWN, the expected code in AVAL being 1, 0 or 9. The value UNKNOWN can only be used if a relevant assessment has been done but the result was unclear.

The variables SRCDOM, SRCVAR, VISITNUM and SRCSEQ, may be used to uniquely define a variable. Derivation flags have to be set if the value in AVAL is not a direct copy of the source, which is almost always the case, as for instance DSDECOD=DEATH in the SDTM input data translated to PARAM=DEATH and AVAL=YES.

The transformation is rendered in a simplified form in Figure 2.



DS	DSDY	DSTEST
	567	LOSS TO FOLLOW-UP

XT	XTDY	XTTESTCD	XTORRES
	63	IRF	PR

CM	CMDY	CMCAT
	224	ANTITUMORAL

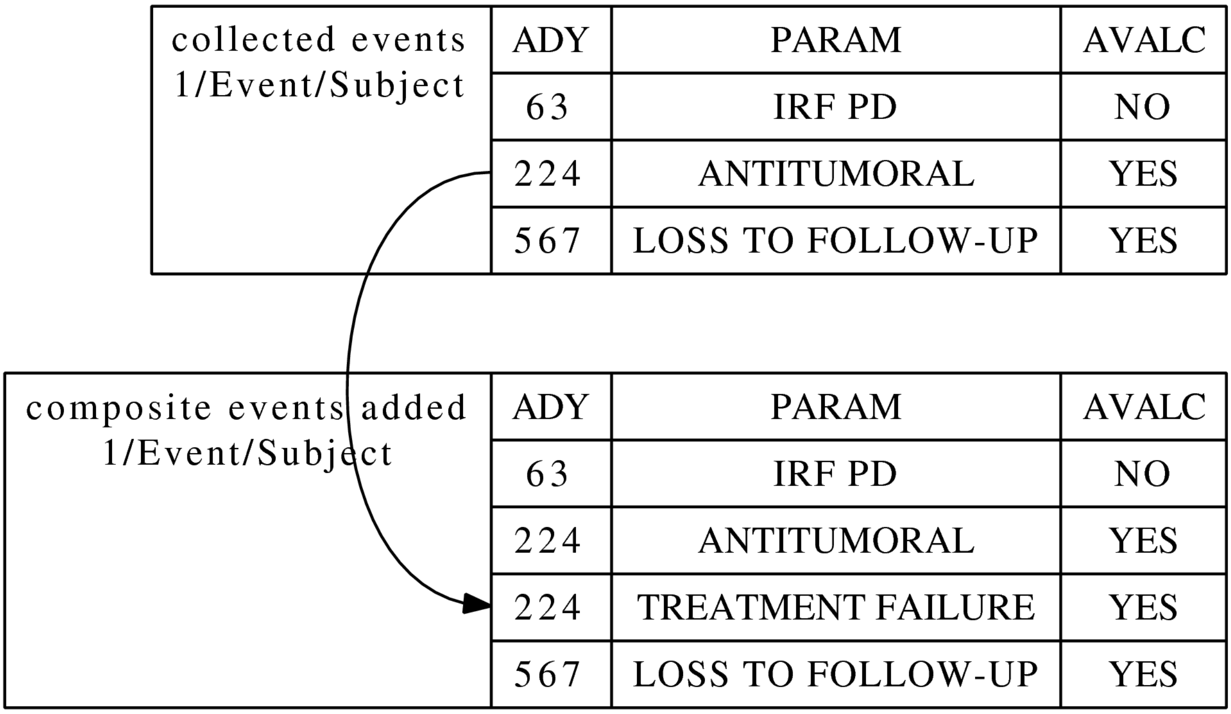
Intermediate 1/Event/Subject	ADY	PARAM	AVALC
	63	IRF PD	NO
	224	ANTITUMORAL	YES
	567	LOSS TO FOLLOW-UP	YES

Figure 2: Collecting all relevant events into a single table

Adding composite events to table

The table is now completed by adding lines wherever a composite event is reached. It is assumed that the event can be completely derived from the information in the source table. For instance, a line containing the entry PARAM=TREATMENT FAILURE and AVALC=YES would be added after the line with the first occurrence of any of TERMINATION FOR TOXICITY, DEATH, IRF PROGRESSIVE DISEASE, DEATH, or ANTITUMORAL THERAPY events with AVALC=YES. The process is illustrated in simplified form in Figure 3. Note that any further occurrence of one of these conditions, for instance death after different antitumoral therapy, does not cause a re-occurrence of the composite event in this example. In general, a reoccurrence of a composite event may be recorded if a repeated event analysis is meaningful in this context.

The occurrences of composite events would have to be marked as derived variables, so PARAMTYP is set to DERIVED and DTYPE can assume the value COMPOSITE.



collected events 1/Event/Subject

ADY	PARAM	AVALC
63	IRF PD	NO
224	ANTITUMORAL	YES
567	LOSS TO FOLLOW-UP	YES

composite events added 1/Event/Subject

ADY	PARAM	AVALC
63	IRF PD	NO
224	ANTITUMORAL	YES
224	TREATMENT FAILURE	YES
567	LOSS TO FOLLOW-UP	YES

Figure 3: Adding composite events

Transformation into time-to-event analysis table

The time-to-event analysis table would contain one record per subject per event for single-event analyses, and one record per subject per event type per event, recording time intervals instead of full time since RFSTDT. The events may be simple or composite.

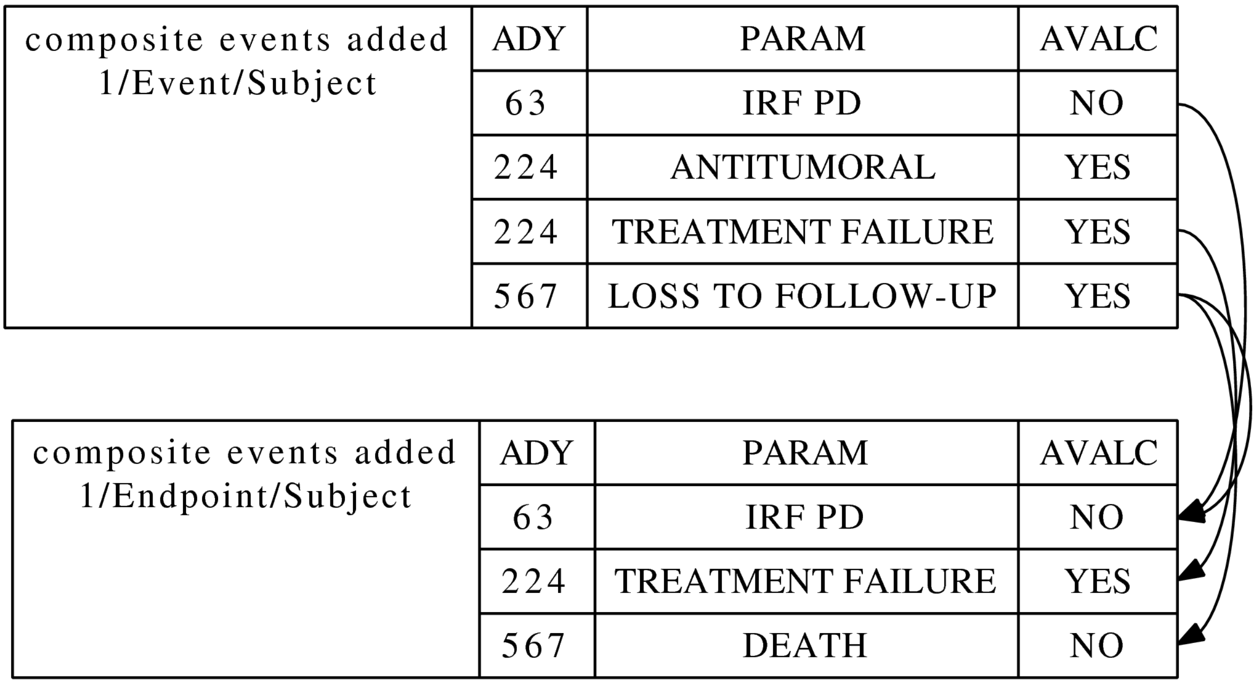
In a single event analysis, every endpoint occurs exactly once. The time is stored in the variable ADY, AVALC can only be YES and NO for event or censoring times. This approach is illustrated in .

In a multiple event analysis, it may be a valid approach to introduce two supportive date variables (see Table 3.3.1 in the ADaM IG) named ASTDY and AENDY.

Note that the process of derivation is not straightforward here. Some event times, such as time at loss to follow-up, serve as censoring times for different events, e. g. death. In the case of non-occurrence of progressive disease, however, the time of last IRF assessment before the censoring event is taken as censoring time.

Problems

As variables named with the suffix –DY cannot be 0, they cannot be used straightforwardly in analyses where left-truncation is involved and intervals start before RFSTDT. In this case, they have to be treated as additional analysis variables.



composite events added 1/Event/Subject	ADY	PARAM	AVALC
	63	IRF PD	NO
	224	ANTITUMORAL	YES
	224	TREATMENT FAILURE	YES
	567	LOSS TO FOLLOW-UP	YES

composite events added 1/Endpoint/Subject	ADY	PARAM	AVALC
	63	IRF PD	NO
	224	TREATMENT FAILURE	YES
	567	DEATH	NO

Figure 4: Assembling the table of endpoints