PhUSE 2012

Paper DH08

Legacy Data Conversion: A Journey of Discovery

Pantaleo Nacci, Novartis Vaccines & Diagnostics Srl, Siena, Italy

ABSTRACT

This paper is about the experience gathered while heading a group of programmers in the effort to convert data for more than 150 studies from a legacy, proprietary format to an industry-standard one. This entailed several layers of complexity, from the challenge to recover all the original basic documentation (study protocol and all its amendments, CRFs, clinical study reports), to the heterogeneity of the data structures, to the need to quickly onboard and make effective external contractors with experience in the new standard but none on the company one(s).

INTRODUCTION

Novartis V&D has been working for quite some time now on a very challenging initiative, calling for the setting up of a brand new validated environment for clinical trial data collection, management, analysis, and reporting, as well as the remapping of all existing study data to a common standard. As with many similar projects, the final scope and complexity level proved much wider than initially envisioned. A consistent number of people, many of them SAS programmers, have been, and still are, working on various aspects of the project, and their geographical dispersion as well as the need to make sure that necessary information flows correctly have made things even more interesting.

First we will go through an overview of the CDR project, and then concentrate on various aspects of the work of the team of programmers in charge of remapping the first set of legacy study data into our own version of CDISC SDTM 3.1.2 (from now on NVD SDTM) and the many challenges they were faced with.

THE CLINICAL DATA REPOSITORY (CDR)

Clinical Data Repository is the name of the new NVD system for storing, managing and reporting on clinical studies. CDR has been developed to revolutionize our ability to:

- Address complex health authority questions quickly and completely
- Produce CDISC compliant submissions
- Review safety data in real-time, mine our overall database for scientific and commercial queries
- Improve overall productivity in Global Clinical Research & Development

BRINGING ALL TOGETHER

Figure 1 on the next page is the graphical representation of how the CDR will be placed in the context of our revised data flows, playing the role of central hub for many processes previously linked much more loosely.

One example is the coding of adverse events, medical history, medications, etc., previously managed in EDC on a study-by-study basis, and creating important issues when there was the need to unify dictionary terminology after pooling of data, since there was no central location containing all mappings of verbatims to preferred terms.

As you can also see, there are already further developments waiting for their turn, like a tighter integration of our Pharmacovigilance systems.

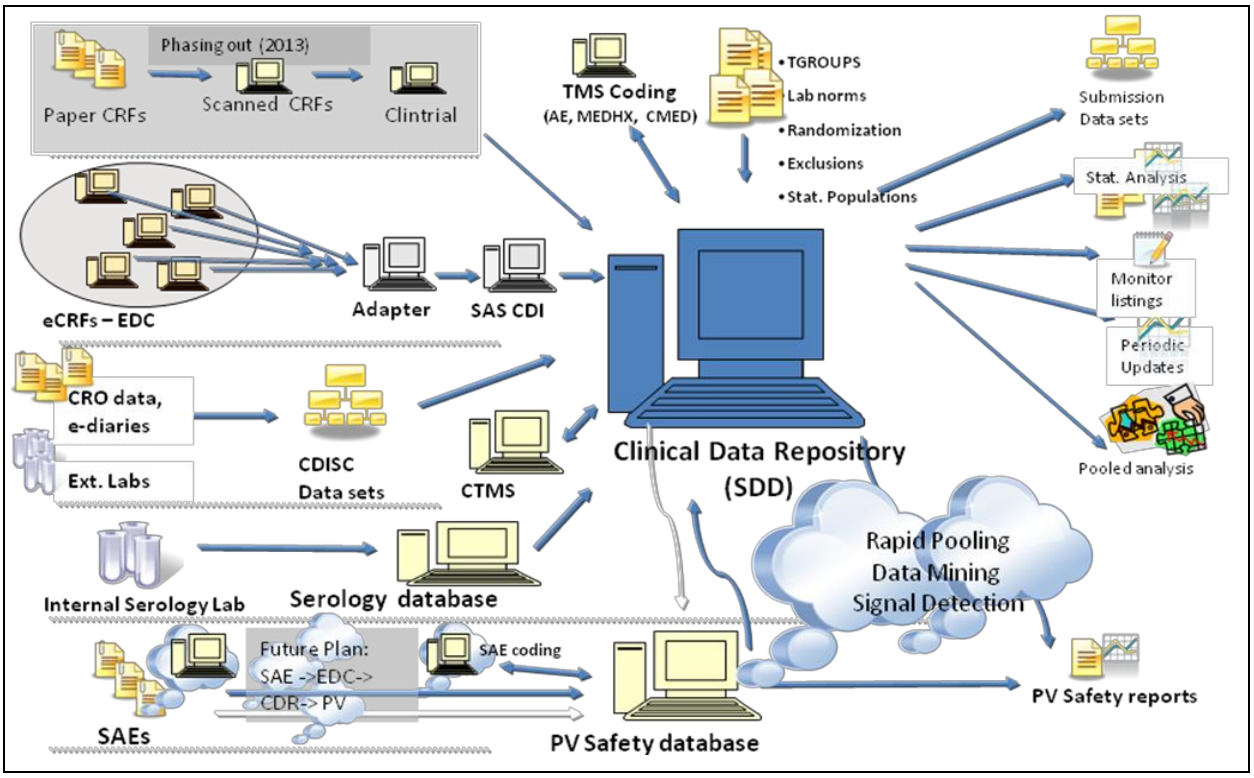

Figure 1: The CDR as a hub for multiple processes

BUSINESS DRIVERS

The business drivers for its implementation were:

- Increase in volume and complexity of Health Authorities expectations and reduced turnaround times
- Number of studies and subjects going up, increasing the workload
- Push to reduce time needed to develop a new vaccine while maintaining quality
- All this brings the need to get rid of old, non-scalable processes & systems

MORE THAN MEETS THE EYE

As you can easily understand, CDR is actually an umbrella, under which seven major projects have been identified:

1. Design, configure, & validate two SAS applications (SDD and CDI)
2. Definition of two new CDISC-based data standards (CDASH and SDTM)
3. Multiple interfaces & database connections setup
4. New dictionary coding system (Oracle TMS)
5. Development/update of SOPs, Work Instructions, and SOP related documents
6. Legacy Data Conversion (LDC)
7. SAS programs conversion, including those in Standard Reporting Software

DELIVERABLES & RESOURCES

Until today, the list of deliverables and resourced needed to work on them includes:

- Implementation of two CDISC-based data standards
- Three new validated IT applications
- Twenty-eight months
- Almost 50 team members involved from various functions
- Almost 60 SAS programs adapted or rewritten to run in CDR with the new data structures
- More than one hundred controlled documents created or revised
- Data remapped for 153 legacy studies, spanning over 19 years and enrolling more than 76,000 subjects
- Almost 5,000 pages of validation documentation
- More than 250 million of legacy data points converted and validated

LEGACY DATA CONVERSION

Novartis Vaccines (previously Chiron Vaccines) has accumulated electronic data from clinical trials for two decades, the earliest ones dating back to (at least) 1992, and spanning over a considerable number of vaccines (influenza, meningitis, rabies, tick-borne encephalitis, Hib, HIV, TDaP, HBV, etc.) as well as adjuvants and preservatives.
For our exercise we defined as 'legacy' all the study for which data were not collected using the new CDASH-compliant EDC eCRFs, thus needing remapping to SDTM: consequently this definition includes several studies still running today.

THE CHOICE OF THE LEGACY STUDIES TO INCLUDE IN PHASE 1

The number of legacy studies included in Phase 1 is the result of a compromise: we have collected evidence of data from roughly 400 legacy studies. This number was too big to run such an exercise on a brand new, still unknown system, so we asked our Clinical colleagues to come up with a more manageable subset.
This selection process was based on flagging as many completed studies as possible in a few high priority projects or containing a certain chemical entity, and so the final list of 153 studies was agreed, and the real work could start.

DATA INSPECTION

Once the initial subset of studies had been identified, a data inspection was undertaken, with the aim to identify all the datasets used in those studies and the variables in each of them.
In our case we could further split the selected studies in pre- and post-Clintrial adoption:

- Clintrial studies are those which had been managed using Clintrial 3 or 4, and are based on minor variations of the internal non-CDISC standard inherited from Chiron and known and well understood by most members of the programming group
- Pre-Clintrial ones on the contrary were managed by external CROs in the first half of the 90's, so that existing knowledge on them was extremely limited

NVD CDM is still using Clintrial 4 today, so retrieving data for those studies posed no issue. After successfully restoring an instance of Clintrial 3 from tape (quite a challenging task), study data were either extracted again from Clintrial or retrieved from our electronic archives.
In the first case extensive checks were run against the same data extracted at the time of the latest analysis available on our legacy server, and all discrepancies thoroughly investigated, addressed and fully documented. This activity took in the end a huge amount of time, since it was often very difficult to trace back the original change documentation which could be used to validate any deviations: e.g., in one case we had to add back one adverse event, since no positive trace of the reason for its removal was found.
At the same time we tried to get hold of as much as possible of the original study documentation (protocols and all amendments, CRFs, clinical study reports, etc.), but in some cases we failed, completely or in part. Since in the early 90's protocols in Chiron once issued were not amended, but only self-standing incremental amendments were created (so that you need to read them all in the right order to understand what was planned to happen), looking at the data warned us that we were missing some amendments, like when we found data related to a previously unknown visit and two distinct amendments linked to it.
As a result we were sometimes forced to guess the meaning of some uncommon variables by running comparisons with similar studies for which more documentation was available. This approach proved mostly ok, but in a few cases the dangers of such an approach were evident, like when the same variable (RSGPT) was used in three studies with the same vaccine, but in two representing a repeated test result (Figure 2), in the other a binary flag (Figure 3). The CRF for the latter study was not available, and so this specific discrepancy was only identified when looking at the ranges and distribution of the pooled data.

RSERUMD	RSGPT	RSGPTCS	COMMENT
11/25/92	68		C

Figure 2: RSGPT contains a test result

SGPTDT	RSGPT	VSGPT	COMMENT
	2		
	2		
	2		
	2		
	2		
11/25/92	1	46	C
02/01/93	1	57	

Figure 3: RSGPT contains a yes/no variable

The initial idea was to create a full list of all variables met in every dataset, with some basic info on type, format, label, etc. (like the output from PROC CONTENTS) plus a comment field explaining where the variable was mapped, and, if not, why. This step was never completed due to other competing priorities, and we realized only much later what a big mistake that had been.

What we did create is a list of all datasets and variables as yet unmapped, so that when a new domain becomes available or a new study gives indication on where to map some of them, we will know where to look for the data and what to do with them.

THE REQUIRED SKILLSET FOR LDC TEAM MEMBERS

The optimal skill set of the SAS programmers working in the LDC team included both a good knowledge of the SDTM standard, and a lot of experience with the old one(s). Unfortunately very few of the NVD staff had had any exposure to CDISC before we embarked in this adventure, so we had to rely heavily on external consultants whenever we had a question on where to map some legacy variable, and the consultants did not have a full understanding of the information we were talking about, at least at the beginning. Things started to get better quickly though, as soon as everybody involved started accumulating more experience with the opposite standard.

On top of that, the NVD SDTM standard, to be used for all future studies and used as our golden reference, was still being refined at the same time we were planning the remapping, so in some cases we ended up going in opposite directions, e.g., leaving out some legacy information only to discover too late that we should have placed it in a certain variable when actual data from the first CDR studies started to be available. That was also when we realized that another domain, QS, had been added to the standard set without warning.

CREATING THE MAPPING SHEETS FOR EACH DOMAIN

To facilitate the work of the programmers, mapping datasets for each domain were developed. They contained a variable number of records for each study to be remapped, and tried to establish a connection on a study level between legacy and SDTM variables, indicating in which legacy panel the variables could be found and where to look for the legacy dataset (Figure 4). So, e.g., when developing the EX domain in most cases STUDYID could be derived from PROT or PROTOCOL, USUBJID from PTNO or SUBJECT, EXLOC from SITEW or INJSITE, and so on. A blank cell would mean that no legacy variable had been identified as containing that information: further investigation led in some cases to the identification of another variable, but in others the info had simply not been collected.

	DOMAIN	VAC	LBN	DSN	LOC	VERS	STUDYID	USUBJID	EXLOC	VISIT	VISITNUM	EXGRPID	EXSTDTC	EXSTDTC2	EXENDTC
23	EX	HIV	V24P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V24\V24P1\	1	PROT	PTNO	SITEW	VISIT	VISIT	INJNO	STARTDT	STARTTM	STARTDT
24	EX	FLUN	V25P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V25\V25P1\	1	PROT	PTNO		VISIT	VISIT	INJNO	STARTDT	STARTTM	STARTDT
25	EX	HIV	V26V6P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V26V6\V26V6P1\	1	PROT	PTNO	SITEW	VISIT	VISIT	INJNO	STARTDT	STARTTM	STARTDT
26	EX	HCV	V35P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V35\V35P1\	1	PROT	PTNO	SITEW	VISIT	VISIT	INJNO	STARTDT	STARTTM	STARTDT
27	EX	HBV	V42P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V42\V42P1\	1	PROT	PTNO	SITEW	VISIT	VISIT	INJNO	STARTDT	STARTTM	STARTDT
28	EX	HBV	V42P2	DOSING	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V42\V42P2\	1	PROT	PTNO	SITEW		VISNUM		STARTDT	STARTTM	STARTDT
29	EX	HIV	V46P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V46\V46P1\	5	PROTOCOL	SUBJECT	INJSITE	VISIT	VISIT		DATET	INJTIME	DATET
30	EX	HIV	V4P3	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V4\V4P3\	1	PROT	PTNO	SITEW	VISIT	VISIT	INJNO	IMMUNDT	IMMUNTM	IMMUNDT
31	EX	FLUN	V57P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V57\V57P1\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
32	EX	MENACWY	V59P1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P1\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
33	EX	MENACWY	V59P10	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P10\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
34	EX	MENACWY	V59P11	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P11\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
35	EX	MENACWY	V59P13	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P13\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
36	EX	MENACWY	V59P14	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P14\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
37	EX	MENACWY	V59P16	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P16\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
38	EX	MENACWY	V59P17	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P17\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
39	EX	MENACWY	V59P18	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P18\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
40	EX	MENACWY	V59P1E1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P1E1\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
41	EX	MENACWY	V59P2	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P2\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
42	EX	MENACWY	V59P20	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P20\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
43	EX	MENACWY	V59P21	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P21\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
44	EX	MENACWY	V59P22	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P22\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
45	EX	MENACWY	V59P2E1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P2E1\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
46	EX	MENACWY	V59P2E2	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P2E2\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
47	EX	MENACWY	V59P3	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P3\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
48	EX	MENACWY	V59P4	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P4\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
49	EX	MENACWY	V59P5	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P5\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
50	EX	MENACWY	V59P5E1	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P5E1\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
51	EX	MENACWY	V59P6	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P6\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
52	EX	MENACWY	V59P7	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P7\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
53	EX	MENACWY	V59P8	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P8\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
54	EX	MENACWY	V59P9	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V59\V59P9\	8	PROT	PTNO	SITEW	VISNUM	VISNUM	INJNO	STARTDT	STARTTM	STARTDT
55	EX	HSV	V5P1	VACCINE	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V5\V5P1\	5	PROTOCOL	SUBJECT	INJSITE	VISIT	VISIT		DATE		DATE
56	EX	HSV	V5P10	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V5\V5P10\	6	PROTOCOL	SUBJECT	INJSITE	VISIT	VISIT		DATET	INJTIME	DATET
57	EX	HSV	V5P11	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V5\V5P11\	6	PROTOCOL	SUBJECT	INJSITE	VISIT	VISIT		DATET	INJTIME	DATET
58	EX	HSV	V5P12	IMMUN	E:\CDR\SOURCE_DATA\PHASE1\DEVEL\OUTPUT\V5\V5P12\	6	PROTOCOL	SUBJECT	INJSITE	VISIT	VISIT		DATET	INJTIME	DATET

Figure 4

These files were then used by the programmers developing the SAS programs for each domain in various ways according to personal coding style and previous knowledge of the data.

PROGRAMMING APPROACH

One of the main decisions we had to make early on was how to structure the remapping exercise: two main options were on the table:

- Assign small sets of studies (5-10) to each programmer for them to develop code to remap all study data
- Assign one or more domains to each programmer, who would then develop programs converting legacy data from all 153 studies in one go

Since we were going to deal mostly with data which were already sharing a largely standard structure, initially we had planned to use SAS CDI, and the whole team was trained on it. But as soon as we started testing how it would work in reality we agreed that given the already high level of SAS programming skills in the team it would be much faster developing 'normal' SAS programs. We also decided to assign to programmers domains rather than studies, reasoning that it would be mush easier that way to leverage existing code when adding new studies.

Several macros were developed to cover common tasks like creating ISO-compliant dates and standard variables (STUDYID, USUBJID, VISITNUM) or calculating intervals and ages, so that the programmers could concentrate just on the data they had to remap.

In the end the logical structure used by the various programmers was basically always the same, with minor variations: they developed study-specific modules to reshape input data as needed, and then a core program, common to all studies, creating the actual SDTM data. In a few cases post-processing study-specific modules were used too.

VALIDATION APPROACH

We included QA in our discussion from the very beginning, since everything in CDR needed to be properly validated. The approach to validation we agreed initially was based on a quality control concept: remap the data first, then sample a subset and compare to the original data, if no discrepancies are detected the program is validated, otherwise select another subset and loop until no more differences are found.

We planned our programming resources according to this approach, but then QA had second thoughts, and asked that we use a double-programming approach instead, doubling our needs for senior programmers in one swift move.

Looking back now the final decision was definitely right, but the delayed timing created issues for the team, since everybody had to divide their time and attention to study and look after data not always logically related.

A minimum set of necessary validation documents was also defined for each program, described in a validation plan: we started with functional specifications all the way to the final IQ, OQ and PQ steps. At the end a validation report has been prepared.

FILLING IN THE T-DOMAINS

The major improvement I personally found when looking at SDTM 3.1.2 is that there is a designated place to store the study metadata, i.e., those information which refer to the single subjects enrolled in it. Since our final goal was to be able to slice and dice all available data quickly according to possibly very complex specifications (e.g., all children in non-US studies less then 6 month of age at enrollment exposed to at least one dose of any thiomersal-containing flu vaccine during 2009), this aspect was critical. The incomplete status of the original documentation for some early studies made their completion (especially TI, TS, and TV) problematic, for example when only an abbreviated CSR was available.

To complement the existing t-domains we designed a couple of custom metadata domains, actually rather lookup tables, to easily identify what lies behind a TA.ELEMENT value. The first one, VC (Vaccine Components), contains the 'building blocks' of all the vaccines we have used in our clinical trials (Figure 5), while the other, VF (Vaccine Formulations), uses sets of records from VC to summarise the exact composition of a certain vaccine (Figure 6). This structure allows for a good degree of flexibility, since VC elements can be reused as much as needed in unlimited permutations.

DOMAIN	COMPCD	COMPTY	COMPSNM	COMPLNM	COMPDS	COMPUN
VC	C0067	OTHER	THIOMERSAL	Thiomersal	0.05	mg
VC	C0068	OTHER	THIOMERSAL	Thiomersal traces		
VC	C0069	ADJUVANT	LTK63	LTK63 + SMVB	3	mcg
VC	C0070	ADJUVANT	LTK63	LTK63 + SMVB	10	mcg
VC	C0071	ADJUVANT	LTK63	LTK63 + SMVB	30	mcg
VC	C0072	ADJUVANT	LTK63	LTK63	30	mcg
VC	C0073	ANTIGEN	FLU_H3N2	A/Moscow/10/99	7.5	mcg
VC	C0074	ANTIGEN	FLU_B	B/Hong Kong/330/2001	7.5	mcg
VC	C0075	ANTIGEN	FLU_H1N1	A/New Caledonia/20/99	7.5	mcg
VC	C0076	ANTIGEN	FLU_H3N2	A/Wisconsin/67/2005	7.5	mcg
VC	C0077	ANTIGEN	FLU_H1N1	A/Solomon Islands/3/2006	7.5	mcg
VC	C0078	ANTIGEN	FLU_B	B/Brisbane/60/2008	7.5	mcg
VC	C0079	ANTIGEN	MEN_C	MenC-CRM197, lyophilised	10	mcg
VC	C0080	ADJUVANT	ALUM	Aluminium hydroxyde		
VC	C0081	ANTIGEN	TBE	TBE, K23 strain	0.75	mcg
VC	C0082	OTHER	POLYGELINE	Polygeline		
VC	C0083	ANTIGEN	FLU_B	B/Beijing/184/93	7.5	mcg
VC	C0084	ANTIGEN	CMV_GB	Glycoprotein B	5	mcg
VC	C0085	ANTIGEN	CMV_GB	Glycoprotein B	30	mcg
VC	C0086	ANTIGEN	CMV_GB	Glycoprotein B	100	mcg

Figure 5: Example of the VC custom domain

DOMAIN	FORMCD	FORMSNM	FORMLNM	FORMCMP	FORMDS	FORMUN	GENERIC
VF	F0019	FLUAD	Fluad Northern Hemisphere 2002/03 w/ thiomersal	C0004,C0014,C0025,C0043,C0067	0.5	mL	MF59-eTIV
VF	F0020	FLUAD	Fluad Northern Hemisphere 2002/03 w/ thiomersal traces	C0004,C0014,C0025,C0043,C0068	0.5	mL	MF59-eTIV
VF	F0021	FLUAD	Fluad Northern Hemisphere 2002/03	C0004,C0014,C0025,C0043	0.5	mL	MF59-eTIV
VF	F0022	FLUAD	Fluad Southern Hemisphere 2003	C0004,C0014,C0025,C0043	0.5	mL	MF59-eTIV
VF	F0023	FLUAD	Fluad Northern Hemisphere 2003/04	C0004,C0014,C0025,C0043	0.5	mL	MF59-eTIV
VF	F0024	FLUAD	Fluad Southern Hemisphere 2004	C0004,C0015,C0025,C0043	0.5	mL	MF59-eTIV
VF	F0025	FLUAD	Fluad Northern Hemisphere 2004/05	C0004,C0015,C0026,C0043	0.5	mL	MF59-eTIV

Figure 6: Example of the VF custom domain

Values in VF are then used to fill in TA.ELEMENT (Figure 7), so that a program can easily identify what each treatment arm was planned to be exposed to. Applying the same logic to EX.EXTRT we can find out what actually happened to each subject.

STUDYID	DOMAIN	ARMCD	ARM	TAETORD	ETCD	ELEMENT	TABRANCH
V70P1	TA	A	FLUAD_N	1	SCR	SCREENING	RANDOMIZED TO A
V70P1	TA	A	FLUAD_N	2	FLUAD21	FLUAD0021	
V70P1	TA	B	FLUAD_O	1	SCR	SCREENING	RANDOMIZED TO B
V70P1	TA	B	FLUAD_O	2	FLUAD20	FLUAD0020	

Figure 7: Example of a TA domain

RECODING SAFETY DATA

The whole point of pooling data is to be able to look at them together in a meaningful way, so the dictionaries used must be harmonized. We decided to recode only adverse events, which over time had been coded using a variety of dictionaries ranging from COSTART to various versions of MedDRA. Since, as shown above, at the same time we moved all our coding activities to TMS, we had to identify all unique verbatims ever used in the selected studies, spell-correct them when needed and then fool the coding system into thinking that these terms were coming from running studies, so that the normal coding process could be used.

DEVELOPING IN A NEW ENVIRONMENT

One thing we found out soon is that when coming from a SAS server installation, programming in SAS Drug Development feels very different (read: slower). Just copying over programs and trying to run them 'as is' proced quickly not to be a good strategy, even if they did actually work, because of the new architecture and philosophy behind SDD.

The first hurdle lies in the way libnames are defined, and then you need to remember that whenever a library is defined all data sets present in there will be copied to the execution server, even if they will never be used. While in some programs it is feasible to proactively select only those data sets you will actually need, when dealing with general use programs that's not always possible or desirable. Another major time saving came from switching on data compression: on a modern high-end server it takes much less time to compress and uncompress data on the fly than to copy the same data back and forth in uncompressed format.

Since the delivery of an SDD environment with a complete implementation of the new directory structure (first QA, then production) took longer than planned, we ended up developing most of the code on our legacy SAS server anyway, so that when the new system became available we needed to work almost exclusively on the libnames and filenames to make the programs run there too. In the specific case of the double program I wrote for EX the running time was initially 8 minutes vs. 1 hour, which was later reduced to roughly 15-20 minutes after the code was optimized for SDD. The remaining difference is in my opinion offset by SDD offering a complete audit trail fully integrated into it, the capability to e-sign any object, and so on.

CONSISTENCY CHECKS

Among other very useful functionalities, SDD comes natively with a full set of consistency checks. Initially we planned to use them extensively in our development work to check that what we were obtaining as an output was actually SDTM, but an issue with performance forced us to go for the OpenCDISC Validator instead. For one of our larger studies the time to run a full set of checks went from hours to minutes, but if in the meantime SAS has identified and fixed the cause of that difference, so we will probably go back to using the SDD checks next time. For now though we have been forced to copy all the SDTM domains back and forth to our legacy SAS server, where the Validator has been installed.

MANAGING EXPECTATIONS

The business case and presentation material developed for the CDR included also an economic aspect, talking about expected savings, easier scalability and overall shortening of timelines.

These were then and still remain some of the main objectives of the whole project, but the positive effects of so many changes applied at the same time will only be fully evident once the CDR reaches a certain degree of stability, when all the new or hugely modified processes we came up with, until recently existing only on paper, will have been tested and refined.

The LDC Phase 1 component has been delivered much later than other pieces, due to multiple reasons not always dependent on the programming: especially creating and debugging LB proved an incredibly challenging activity due to the variety of data structures and the sheer quantity of data errors we identified along the way, but the bottom line was that from the point of view of management we missed one deadline after the other.

OVERALL NUMBERS

A few facts on Phase 1 of the LDC exercise:

- 153 legacy studies (spanning 19 years from 1991-2010; 76,581 subjects in total)
- Number of SAS datasets: 2,837 (~18 datasets per trial)
- Number of variables per dataset (on average): 37
- Number of records per dataset (on average): 2341
- Total number of data points: 245,542,542 (Every point has been validated!!)
- Number of SAS programs written for the Legacy Data Conversion: 56
    - Cross-domain programs: 20
    - Domain-specific (original + double-program): 36
    - "...they're not 'lightweight' programs", "Very, very complex"
- Number of Validation Packages (UVRs): 23
    - Each package includes a Functional Specification, IQ, OQ, and some have PQs
- Retrieval of study documents (protocol, CSR, CRF, SAP; mostly on paper in archives), scanning, uploading into our EDMS
- 314 T-domains spreadsheets populated by 92 people from Clinical and 28 from BCDM

NEXT STEPS

Now that Phase 1 is finally completed we are already planning Phase 2, trying to take into consideration all the learnings we collected until now.

CONCLUSION

Converting legacy data can easily prove a nightmare, and be at least a very challenging task. Decisions taken early with incomplete or draft information can lead to a blind alley, and parallel development with other processes which will

ultimately provide input into the one you are working on should be avoided if at all possible, e.g., the reference data standard should be stable and already used for studies before you execute any data conversion.

Knowing the legacy data to the last variable is critical, and the same holds for the availability of the original documentation: unless there is clear indication of what has been originally stored in a variable, porting it on a 'best guess' basis will always entail a certain level of risk.

Old data from last century tend to contain on average more mistakes than what we are used to these days, due to both less intensive checks performed at data entry at the time and a higher acceptability threshold used then: plan plenty of buffer time to identify, investigate, address and fully document them.

Documentation and validation activities require a lot of time and attention, and the average SAS programmer is not very keen on spending too much time on them. Identify people who are willing and able to look after them properly and you will never regret it. Keep QA people in the loop at all times, and especially so when making important decisions; always strive to maintain a good working relationships with them, since they are there to help.

Document all decisions and the reasoning behind them, so that in the future it will be possible to understand clearly why a certain route was taken.

Manage expectations of top management carefully, in many cases they believe that all programmers have a 'big red button', so it's just a question of pushing it to see the desired results.

ACKNOWLEDGMENTS

I would like to thank all the programmers at NV&D who worked on this lengthy and often excruciating exercise, and who never gave up. I would also like to thank Todd Miller and Aldo Schepers for preparing and maintaining the material I used as a source for the introduction.

CONTACT INFORMATION

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

Pantaleo Nacci
Novartis Vaccines & Diagnostics Srl
Via Fiorentina, 1
53100 Siena, Italy
Phone: +39 0577 243554
Fax: +39 0577 278443
Email: pantaleo.nacci@novartis.com
Web: www.novartisvaccines.com