Automatische Erzeugung von Programm-Templates mit Spezifikationen in integrierten Analysen

Thomas Wollseifen
inVentiv Health Germany GmbH
Grosse Hub 10d
65344 Eltville am Rhein
thomas.wollseifen@inventivhealth.com

Zusammenfassung

Analysis Data Model (ADaM) Daten werden als Teil von „Submission Packages“ von integrierten Analysen bei der Einreichung von klinischen Studien für die verschiedenen Behörden (FDA/EMA/PMDA) benötigt.
Damit die Daten den Vorgaben von Statistischen Analyse Plänen (SAP) und Guidelines (ADaM Implementation Guide) genügen, werden ADaM Daten üblicherweise mit Hilfe von Mapping-Programmen erstellt. Sie enthalten alle Ableitungen entsprechend der Metadaten, Codelisten und anderen Standards. Das Mapping-Programm, welches Studiendaten in die entsprechende Struktur einer integrierten Datenbank umsetzt, wird üblicherweise manuell für alle benötigten Domänen (ADSL, ADAE, ADDS, etc.) erstellt.
Der Prozess der Erstellung von Mapping-Programmen kann mit Hilfe einer Spezifikation automatisiert werden, die schon alle notwendigen Umwandlungsvorschriften enthält.
In diesem Beitrag wird ein Ansatz vorgestellt, eine Spezifikation basierend auf Metadaten der integrierten Datenbank und der Metadaten der einzelnen Studien automatisch zu erstellen. Informationen aus den Metadaten und Daten der Studie, die mit der integrierten Datenbank verglichen werden, fließen in die Spezifikation ein. Fehlende Mapping-Vorschriften werden dann manuell eingetragen. Es wird vorgestellt, wie mit der vollständigen Spezifikation automatisch Mapping-Programme für alle Domänen erzeugt werden können. Die generierten Mapping-Programme können später erweitert werden. Sie genügen den Anforderungen von Pharmafirmen und Behörden. Eine notwendige Dokumentation des Daten-Mappings kann aus der generierten Spezifikation erstellt werden. Der Vorteil dieses Ansatzes ist, dass der Datentransfer von klinischen Studien in die integrierte Datenbank automatisiert wird und transparenter ist.

Schlüsselwörter: Integrierte Datenbanken, Mapping, Metadaten

1 Einleitung: Integration von Studien in eine Datenbank

Die Integration und Analyse (Metaanalyse) von mehreren klinischen Studien eines Medikaments kann eine Herausforderung sein, da verschiedene Datenstrukturen in eine gemeinsame Datenstruktur überführt werden müssen. Globale Datenstandards sollen die Analyse vereinheitlichen.
Das Clinical Data Interchange Standards Consortium (CDISC) [1] hat verschiedene Datenstandards entwickelt, die einen effizienten Transfer, Zugriff, Review und Analyse von klinischen Studiendaten ermöglichen. Diese Standards beinhalten z.B. das Clinical Data Acquisition Standards Harmonization (CDASH) Modell, das Data Tabulation Mo-

del (SDTM) und das Analysis Data Model (ADaM). CDASH beschreibt die inhaltliche Struktur klinischer Datenbanken, die zur Aufnahme der Probandendaten aus den Case Report Forms gedacht sind. Die Transformation nach SDTM ist hierbei möglichst einfach. In SDTM wird die inhaltliche Struktur von Daten abgelegt, in denen die einzelnen Case Report Forms (CRF) aus klinischen Studien zusammengefasst und bei der FDA eingereicht werden können.
ADaM-Daten beinhalten schon alle notwendigen abgeleiteten Parameter, die für die statistische Auswertung und - bei mehreren klinischen Studien - für die Metaanalyse benötigt werden. ADaM soll es den Behörden ermöglichen, die von Pharmaunternehmen durchgeführten statistischen Analysen zu reproduzieren.
Diese Ausarbeitung beschreibt einen semi-automatischen Ansatz der Datenintegration, welcher auf dem oben genannten ADaM Standard basiert. Es können auch andere Datenstrukturen mit der vorgestellten Methode integriert werden.
In Abbildung 1 wird der Datenfluss von aufgezeichneten Daten einer klinischen Studie vom CRF oder eCRF (electronic case report form), Labordaten und anderer klinischer Daten dargestellt. Die aufgezeichneten Rohdaten werden vom Datenmanagement in SDTM Datensätze umgewandelt. Abgeleitete Variablen, zusätzliche Records (Summary Records) und Flags sowie Subgruppen werden in ADaM Datensätzen abgebildet. Die Analyse Datensätze (ADaM) werden von den statistischen Programmierern verwendet, um Tabellen, Listings und Grafiken zu erstellen, welche in den klinischen Bericht einfließen. Für Metaanalysen werden üblicherweise sogenannte integrierte Datenbanken (IDB) erstellt. Dabei werden die Daten von mehreren klinischen Studien zu einem Medikament in eine gemeinsame Datenstruktur (meist ADaM) überführt. Aus der integrierten Datenbank werden *Integrated Summaries of Safety* (ISS) und *Integrated Summaries of Efficacy* (ISE) für Einreichungen bei Behörden erstellt (sogenannte Submissions).
Dieses Paper stellt eine semi-automatische Methode vor, Daten aus verschiedenen klinischen Studien in eine integrierte Datenbank abzubilden. Die Datenstruktur der jeweiligen klinischen Studie, die integriert werden soll, kann hierbei unterschiedlich sein. Die Aufgabe ist, dass sogenannte Mapping zu automatisieren. Unterschiedliche Datenstrukturen werden hierbei auf eine gemeinsame Basis der integrierten Datenbank abgebildet. Zunächst muss eine Spezifikation erstellt werden, die den Mapping-Prozess der klinischen Daten von mehreren klinischen Studien in die integrierte Datenbank beschreibt.

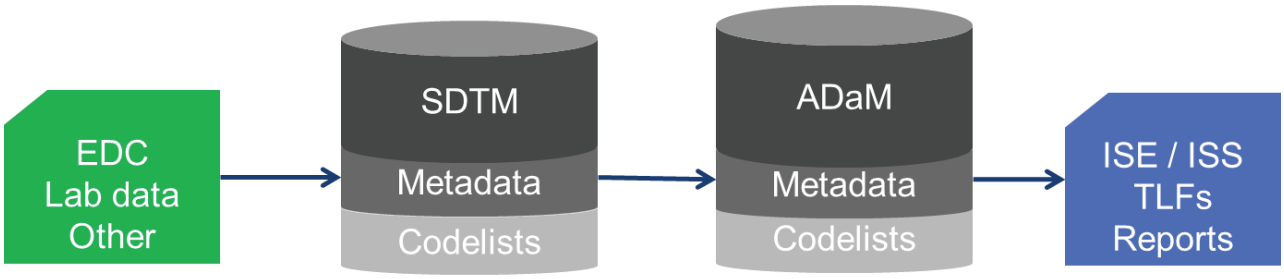

Abbildung 1: Datenfluss einer klinischen Studie über verschiedene Datenstrukturen bis zum Bericht

Abbildung 2 zeigt den typischen Mapping-Prozess von Daten klinischer Studien in eine gemeinsame Datenstruktur – eine integrierte Datenbank (IDB). Die zugehörigen Mapping-Programme werden manuell vom Programmierer basierend auf Spezifikationen (Klinisches Studien Protokoll, Statistischer Analyse Plan (SAP), Data Handling Document) erstellt.
Die Spezifikation beschreibt für jede Domäne und zu jeder Variable, wie sie für die einzelne klinische Studien abgeleitet werden soll. Damit die Daten der integrierten Datenbank einfach und effizient ausgewertet werden können, sind sie im ADaM-Format strukturiert – *Die Daten sind einen Schritt entfernt von der Analyse*. Alle abgeleiteten Parameter sollten in der IDB vorbereitet sein. Im Allgemeinen enthält die IDB verschiedene ADaM-Domänen – ADSL, ADAE, ADEX, ADCM, etc.). Die ADaM-Domänen beschreiben die inhaltliche Struktur der klinischen Studie. In ADSL (analysis subject-level dataset) werden Basisinformation zu jedem Patienten der klinischen Studie abgelegt. In den anderen Domänen sind z.B. Safety Informationen (ADAE, Adverse Events), die Medikation (ADEX, Exposure) und weitere Daten, die erhoben wurden. Die integrierte Datenbank besitzt eine Metadatenstruktur, die die Domänen beschreibt. Für das Mapping jeder Studie gibt es zu jeder Domäne ein Mapping-Programm. Die Studiendaten der einzelnen klinischen Studien besitzen ihrerseits auch wieder Metadatenstrukturen.
Damit die Mapping-Programme erstellt werden können, muss eine Spezifikation vorliegen. Diese wird meist parallel mit Erstellung des statistischen Analyse Plans für die integrierte Analyse vorbereitet.

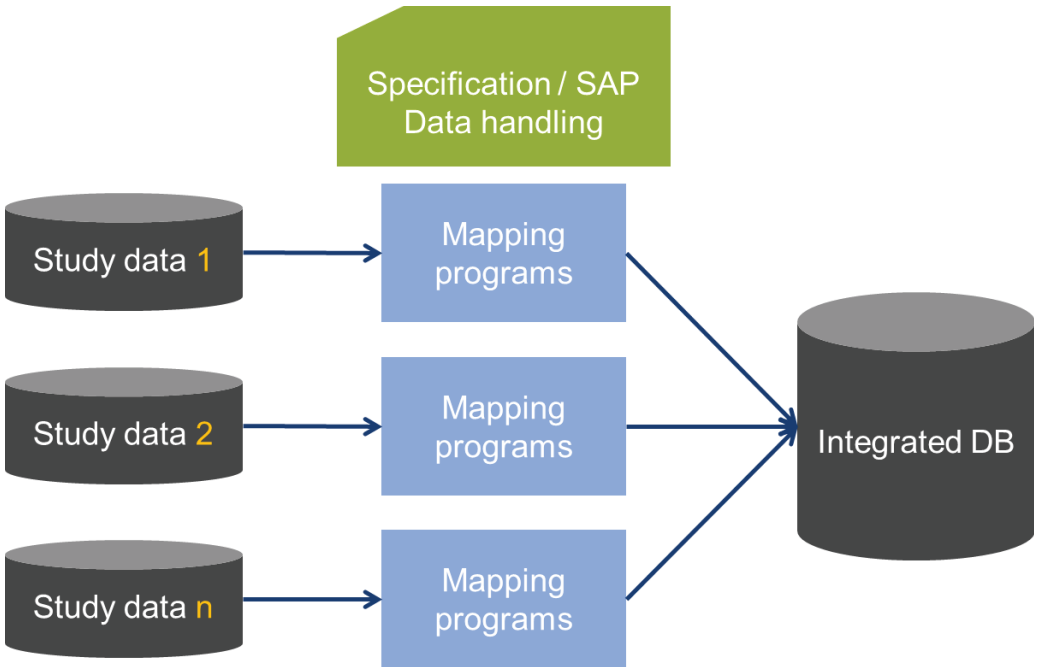

Abbildung 2: Mapping von mehreren klinischen Studien in eine integrierte Datenbank

Im folgenden Abschnitt wird der Prozess der semi-automatischen Erzeugung von Spezifikationen dargestellt, die auf den Metadaten der IDB, den Metadaten und Daten der jeweiligen klinischen Studie basieren.

2 Ansatz: Semi-automatische Spezifikation und Mapping-Programm-Generierung

Die grundsätzliche Vorgehensweise, eine Datenstruktur in eine andere Datenstruktur zu überführen, wird in einer Spezifikation festgehalten. Diese enthält alle Regeln und Definitionen, eine Struktur **A** in einer Struktur **B** abzubilden.
In klinischen Datenstrukturen, speziell in SDTM oder ADaM, gibt es Domänen mit einer vom Standard (CDISC) definierten Struktur. Der Standard gibt vor, wie Domänenamen, Variablennamen, Labels, Formate und der Inhalt der Daten aufgebaut sind. Diese Information fließt in die Metadaten einer integrierten Datenbank ein.
Die Metadaten beschreiben den Aufbau einer integrierten Datenbank zu einem Präparat. Sie können Regeln enthalten, wie beispielsweise Ersetzungsregeln, Definitionen der Visitenstruktur, Ableitungen spezieller Variablen oder Definitionen über zusätzliche Records im Datensatz für statistische Analysen (z.B. Compliance über vom Patienten genommene Medikation). Ableitungsregeln sind für verschiedene Präparate unterschiedlich. Jedes Pharmaunternehmen hat andere Datenstrukturen – die aber heutzutage dem CDISC Standard folgen sollten.

2.1 Beschreibung des Ansatzes

Im Folgenden wird ein semi-automatischer Ansatz beschrieben, eine Spezifikation aus den Metadaten der IDB und den klinischen Studiendaten und deren Metadaten zu erzeugen. Die Spezifikation wird manuell mit weiteren Ableitungsregeln (aus dem SAP oder Data Handling Document) aufgefüllt. Danach werden im nächsten Schritt SAS Mapping-Programme automatisch aus der Spezifikation generiert. Der gesamte Prozess ist in Abbildung 3 dargestellt.

Der vorgestellte Ansatz verwendet grundsätzlich zwei SAS Makros: **%m_create_specification** und **%m_create_mapping**. %m_create_specification ist für die Erstellung der Spezifikation verantwortlich, das Makro %m_create_mapping erzeugt, basierend auf der Spezifikation, die Mapping-Programme.

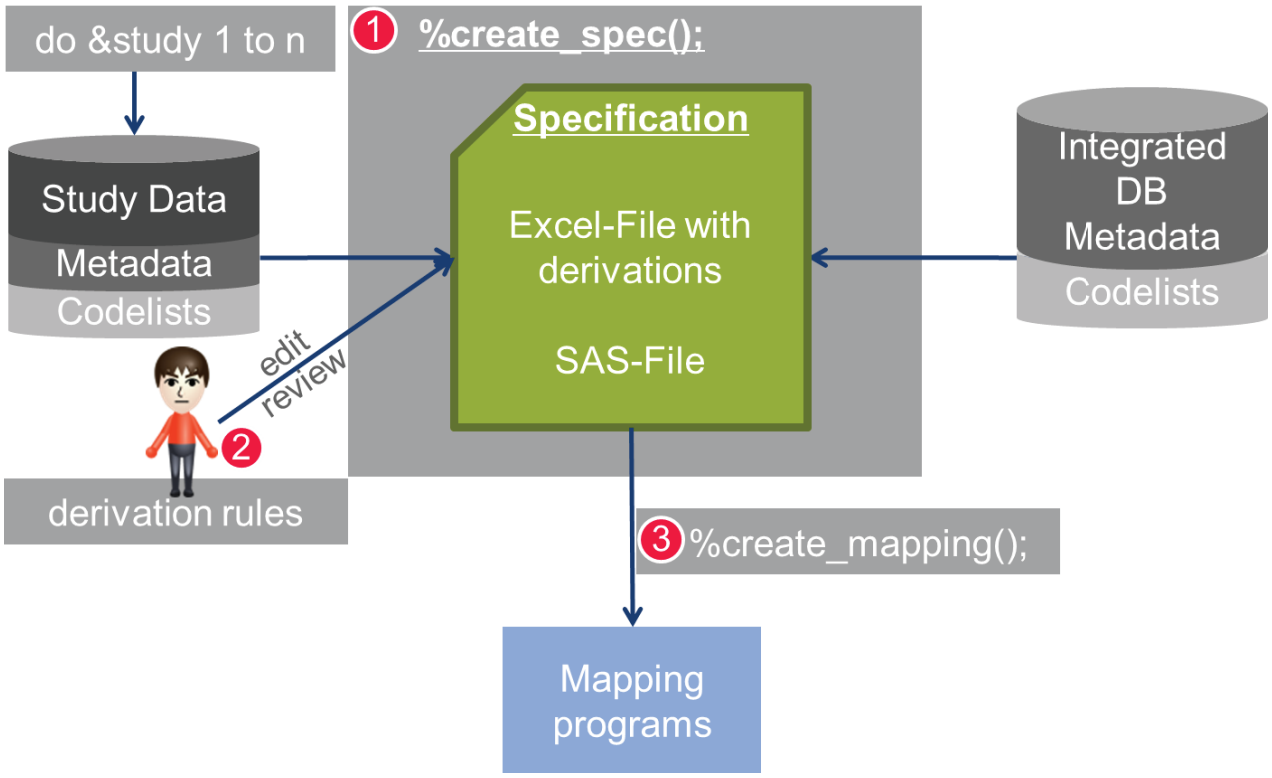

Abbildung 3: Erzeugung von Mapping-Programmen mit einer semi-automatisch erzeugten Spezifikation

MakroMakroSchauen wir uns an, wie die einzelnen Schritte (1) %m_create_specification, (2) manuelle Ableitungsregeln einführen und (3) %m_create_mapping ablaufen:

Schleife über alle zu integrierenden Studien:

1. Generiere eine Spezifikation mit **%m_create_specification** auf den zu integrierenden Daten klinischer Studien
2. **Manuelles** Auffüllen von fehlenden Ableitungsregeln basierend auf dem SAP, Data Handling Document und **Review** der erzeugten Spezifikation durch einen zweiten Programmierer oder Statistiker
3. Erzeugung von Mapping-Programmen mit **%m_create_mapping**

Ende

Der gesamte Prozess läuft auf den zu integrierenden Daten klinischer Studien in einer Schleife. Zunächst erzeugt das SAS Makro %m_create_specification eine Excel-Datei. Diese Datei stellt die Spezifikation für den gesamten Mappingprozess dar. Das Makro %m_create_specification geht alle Domänen der IDB durch und versucht gemeinsame Variablen in Domänen der klinischen Studie zu finden. Im Prinzip handelt es sich um einen *Merge*. Falls Variablen übereinstimmen, gibt es eine 1:1-Abbildung. Falls keine gemeinsamen Variablen gefunden werden, können spezielle vorgegebene Regeln eingefügt werden. Dies sind Makro-Aufrufe. Natürlich werden nicht alle Ableitungsregeln und Definitionen aus dem SAP oder Data Handling Document vom Makro automatisch eingefügt. Diese können manuell ergänzt werden. In einem weiteren Schritt muss die

Spezifikation von einem zweiten Programmierer oder Statistiker überprüft werden. Dieser genehmigt dann die Spezifikation, falls alle Ableitungsregeln dem SAP und Data Handling Document entsprechen.
Im dritten Schritt werden durch Aufruf des Makros %m_create_mapping Mapping-Programme generiert, die alle Mapping Regeln enthalten.
Das Makro %m_create_mapping liest die Spezifikation ein und erzeugt für jede Domäne einer klinischen Studie ein eigenes Mapping-Programm. Die Mapping-Programme sind vollständig lauffähig und enthalten alle notwendigen Header-Informationen und Kommentare, die den Programmierrichtlinien des Pharmaunternehmens genügen. Alle Ableitungsregeln, die in der Excel-Spezifikation gegeben sind, werden in SAS Code umgewandelt. Kommentare für eine Ableitung werden automatisch vor dem entsprechenden SAS Code eingefügt. Notwendige Einrückungen im Programm werden vom Makro %m_create_mapping vorgenommen. Falls zusätzliche Änderungen gemacht werden müssen, ist dies durch den statistischen Programmierer möglich. Jedes automatisch erzeugte Mapping-Programm wird in der Testumgebung überprüft und gestartet, validiert und für den produktiven Lauf freigegeben.

2.2 Semi-Automatische Spezifikation

Das Makro %m_create_specification erhält beim Aufruf alle notwendigen Informationen zum Projekt: den Projektnamen, die Studien ID, die Domänen, die abgebildet werden sollen, die Library zu den Studiendaten, die Metadaten der IDB und den Pfad und Filenamen der resultierenden Excel-Spezifikation.
Beispiel 1 zeigt die Parameterdefinition des Makros %m_create_specification. Im Weiteren werden alle Schritte des Makros beschrieben.

```
%macro m_create_specification(
     Project  =       /*Project name                    */
    ,studyid  =       /*Study ID                        */
    ,domain   =       /*Domains to map (A->B)           */
    ,path     =       /*Path to output specification    */
    ,filename =       /*Excel filename                  */
    ,metalib  =       /*Metadata of IDB                 */
    ,inlib    =       /*Study data                      */
    ,adslbase =adsl   /*ADSL base (DM for SDTM->ADaM)    */
);
```

Beispiel 1: Definition des Makros %m_create_specification

Das Makro %m_create_specification erzeugt eine Excel-Spezifikation, welche beschreibt, wie die Studiendaten in die integrierte Datenbank bezüglich der Metadaten der DB überführt werden sollen.
In der Makro-Definition wird ein Projektname (z.B. Medikamentenname oder therapeutisches Gebiet) definiert. Es ist notwendig Informationen über die IDB (*Libname* der IDB), die Studie (Studien ID, Libname der Studiendaten) und spezifische Informationen über die Ausgabedatei (Pfad, Name der Excel-Spezifikation) an das Makro zu übergeben. Die Mapping-Vorschrift wird auch an das Makro übergeben.

Die Definition des Mappings folgt der Regel: A→B (d.h. bilde A auf B ab, z.B. DM→ADSL#AE→ADAE). Hier kann die Ableitungsregel für jede Domäne angegeben werden. Im Makroaufruf werden die verschiedenen Domänen durch das Symbol ‚#' getrennt.
Die Grundregel, wie Domänen gemappt werden, ist folgendermaßen definiert:

Domain= <source_domain_A> → <target_domain_B>

Beispiel 2 zeigt einen Aufruf von %m_create_specification, in dem Studiendaten in SDTM in ADaM abgebildet werden sollen.

```
%m_create_specification( studyid  =99999
                        ,domain   =ae->adae #
                                   dm->adsl #
                                   ex->adex #
                                   lb->adlb
                        ,path     =&path
                        ,filename =9999_specification.xls
                        ,metalib  =meta
                        ,inlib    =in
                        ,adslbase =dm);
```

Beispiel 2: Beispielaufruf - Spezifikation von SDTM zu ADaM erzeugen

Falls eine unbekannte Domäne im Makro aufgerufen wird oder eine Library nicht existiert, gibt das Makro %m_create_specification eine Warnung zu den kritischen Makroparametern im SAS-Log zurück.
Im folgenden Abschnitt schauen wir uns die verschiedenen Schritte des Makros %m_create_specification an.

1. Vergleiche der Studien Domänen gegen die Metadaten der IDB
2. Generierung eine Excel-Spezifikation:
    a. Schreibe eine Domänenbeschreibung (Domänen, Key-Variablen, Sortierung)
    b. Schreibe die Pool-Metadaten und Studienmetadaten sowie Ableitungen in die Excel-Datei

Das Makro liest die Domänen der Studie ein und vergleicht die Information der Studienstruktur mit den Metadaten der integrierten Datenbank. Es durchläuft eine Schleife über alle Mappings, die in der Domänen-Makrovariablen gegeben sind.
Ein wesentlicher Bestandteil des Makros %m_create_specification ist das Check-Makro **%mcheck_variables_metadata**, welches die Information der Studien Metastruktur jeder angegebenen Domäne mit der Information der Metadatenstruktur der IDB *merged* und vergleicht. Dieser Schritt ist besonders wichtig für die resultierende Spezifikation, da alle strukturellen Elemente der Studien-Daten und –Metadaten mit den Metadaten der IDB verglichen werden. Diese Information wird für den Aufbau der Spezifikation verwendet.

Im Check-Makro %mcheck_variables_metadata werden alle relevanten Informationen der Domänen der Metadaten überprüft und verglichen:

- Variablenname
- Typ (alphanumerisch, numerisch)
- Länge der Variable
- Codeliste/Formate
- Labels (dies ist Optional, da schon kleine Unterschiede in den Schreibweisen entdeckt werden und zu Unterschieden führen)
- Keyvariable
- NotNull Information (d.h., der Inhalt einer Variable darf keine Missings enthalten – in keinem Record)

Das Check-Makro %mcheck_variables_metadata erzeugt mit diesen Informationen einen Datensatz. Dieser wird für den Aufbau der Excel-Spezifikation verwendet.

Im folgenden Beispiel 3 wird ein Aufruf des Makros %mcheck_variables_metadata gezeigt, welches in einer Schleife durch alle angegebenen Domänen (Mappingzuweisungen) läuft. Das Makro wird selbst nur innerhalb des Makros %m_create_specification aufgerufen und ist für den Benutzer nicht sichtbar.

```
%do d=1 %to &numdomains;

 /*************************************************************/
 /*check study data structure against metadata of the IDB     */
 /*************************************************************/
 %mcheck_variables_metadata( indata  = &inlib..&sourcedomain
                            ,meta    = &metalib..&outdomain
                            ,inadsl  = &inlib..&adslbase
                            ,outdata = &outdomain._check);
…
%end;
```

Beispiel 3: Makro %mcheck_variables_metadata vergleicht Studien- mit Metadaten der IDB

Nach Überprüfen und Sammeln von Informationen über die Studienmetadaten im Vergleich zur Metadatenstruktur der IDB, schreibt das Makro %m_create_specification einen Beschreibungsbogen in die Excel-Datei. In ihm sind die verwendeten Domänen, die Sortierungsvariablen, die Schlüsselvariablen innerhalb einer Domäne und die Quell- und Ziel-Domänen angegeben. Ein Beispiel eines Beschreibungsbogens der Domänen ist in Abbildung 4 angegeben.

	domain	sort_order	keyvar	source	target
2	ADSL	STUDYID USUBJID	STUDYID USUBJID	DM	ADSL
3	ADAE	STUDYID USUBJID ASEQ	STUDYID USUBJID ASEQ	AE	ADAE
4	ADEX	STUDYID USUBJID ASEQ	STUDYID USUBJID ASEQ	EX	ADEX
5	ADLB	STUDYID USUBJID PARAMCD	STUDYID USUBJID ASEQ	LB	ADLB
6	ADVS	STUDYID USUBJID PARAMCD	STUDYID USUBJID ASEQ	VS	ADVS
7	ADCM	STUDYID USUBJID CMSEQ	STUDYID USUBJID CMSEQ	CM	ADCM
8	ADEGM	STUDYID USUBJID PARAMCD	STUDYID USUBJID ASEQ	EG	ADEGM
9	ADMH	STUDYID USUBJID MHSEQ	STUDYID USUBJID ASEQ	MH	ADMH
10	ADTA	STUDYID TSPARMCD	STUDYID ARMCD TAETORD	TA	ADTA
11	ADTE	STUDYID ELEMENT	STUDYID ETCD ELEMENT	TE	ADTE
12	ADTS	STUDYID TSPARMCD	STUDYID TSPARMCD	TS	ADTS
13	ADTV	STUDYID VISITNUM	STUDYID VISITNUM	TV	ADTV

Abbildung 4: Beschreibungsbogen mit den abzubildenden Domänen

Nachdem das Makro %m_create_specification durch alle angegebenen Domänen gelaufen ist, erzeugt es für jede Domäne ein Excel-Sheet mit Informationen, die vom Makro über die Metadaten gesammelt wurden. Einen Ausschnitt der jeweiligen Domänenbögen ist in Abbildung 5 dargestellt.

74	24	ADSL	BRTH_Y		4	N	Birth			M	Y	99999
75	25	ADSL	COUNTRY		3	C	Country	COUNTRY		M	RY	99999
76	26	ADSL	ENRLFL		15	C	Population			O	ENRLFN	99999

Description | ADSL | ADAE | ADEX | ADLB | ADVS | ADCM | ADEGM | ADMH | ADTA | ADTE | ADTS | ADTV

Abbildung 5: Spezifikation mit Beschreibungsbogen und Bögen für jede Domäne

Im nächsten Abschnitt betrachten wir die Struktur der Domänen-Bögen. Die Spezifikationsdatei besteht aus folgenden Bögen:

- Beschreibungsbogen (Abb. 4) mit Basisinformationen (Domänen, Sortierung, Schlüsselvariablen, Quell-Domänen, Ziel-Domänen)
- Bögen für jede Domäne der integrierten Datenbank

In den Domänenbögen ist die Mapping-Information der jeweiligen Studiendaten in die integrierte Datenbank beschrieben. Sie enthalten die Information über die Quell- (Studie) und Zielstruktur (IDB) sowie über die jeweiligen Ableitungsregeln je Domäne und Variable. Ein Bogen einer Domäne besteht grundsätzlich aus drei Elementen: Zielstruktur, Quellstruktur und Ableitungsregeln. Diese Elemente sind farblich im Header abgesetzt. In Abbildung 6 ist ein Ausschnitt zu ADSL dargestellt. Hier wird von DM zu ADSL abgebildet.

Project: 99999 - Pool Metadata							Source Study Metadata							Derivation	
Domain Target	sasname	Keyvar	Outform	Type	Label	Codelst	Studyid	Domain Source	Var_study	Label Source	Type study	Format study	Length	Comment	Derivation
	SUBPIM						99999	DM	SUBPIM	inflammator	C	$	200	[not in metadata]	
	SUBPPK						99999	DM	SUBPPK	pharmacoki	C	$	200	[not in metadata]	
	TRT01A						99999	DM	TRT01A	Treatment	C	$	200	[not in metadata]	
	TRT01P						99999	DM	TRT01P	Treatment	C	$	200	[not in metadata]	
ADSL	ADSNAME		8	C	Name	X_ADSNM	99999	DM	ADSNAME	Name	C	$X_ADSNM	8		DM.ADSNAME
ADSL	STUDYID	1	10	C	Identifier		99999	DM	STUDYID	Identifier	C	$	10		DM.STUDYID
ADSL	USUBJID	2	25	C	Subject		99999	DM	USUBJID	Subject	C	$	20		DM.USUBJID
ADSL	POOLSEQ	3	8	N	Sequence		99999							key	
ADSL	SUBJIDN		9	N	Identifier for		99999	DM	SUBJIDN	Identifier for	N		8		DM.SUBJIDN
ADSL	SUBJID		9	C	Identifier for		99999	DM	SUBJID	Identifier for	C	$	9		DM.SUBJID
ADSL	TREATNO		6	N	Number		99999	DM	TREATNO	Number	N		8	[includes only missings]	DM.TREATNO
ADSL	RANDNO		6	N	tion		99999	DM	RANDNO	on Number	N		8		DM.RANDNO
ADSL	UASR		40	C	Subject		99999	DM	UASR	Subject	C	$	40		DM.UASR
ADSL	RASR		20	C	tion		99999	DM	RASR	on	C	$	20		DM.RASR
ADSL	ITTFN		3	N	Treat	NY	99999	DM	ITTFN	Treat	N	NY	8	[includes only missings]	DM.ITTFN
ADSL	PPROTFN		3	N	Protocol	NY	99999	DM	PPROTFN	Set Flag (N)	N	NY	8		DM.PPROTFN
ADSL	SAFFN		3	N	Population	NY	99999	DM	SAFFN	Population	N	NY	8		DM.SAFFN
ADSL	FASFN		3	N	Analysis	NY	99999	DM	FASFN	Analysis	N	NY	8		ADSL.FASFN
ADSL	LOSFN		3	N	Set (N)	NY	99999	DM	LOSFN	Set (N)	N	NY	8	[includes only missings]	ADSL.LOSFN

Abbildung 6: Spezifikation mit %m_create_specification erstellt (Auszug)

Die farblich codierte Spalte zu Studienvariablen zeigt, ob die entsprechende Metadatenstruktur der IDB mit den Studienmetadaten zusammenpasst. Abbildung 6 zeigt ein Beispiel zum Abbilden von SDTM zu ADaM. Rote Zellen beschreiben Variablen, die nicht in den IDB Metadaten vorkommen, aber in den Studienmetadaten. Grüne Zellen deuten eine Übereinstimmung an, das heißt, die Variable kommt sowohl in den Pool-Metadaten als auch in den Studienmetadaten vor. Falls die Zelle orange ist, muss die Variable abgeleitet werden. Sie ist nicht in den Studiendaten vorhanden. Ableitungsregeln können in der entsprechenden Spalte zu *Derivation* eingetragen werden.
Es ist möglich, SAS Code (Befehle im Data Step) und Makro-Aufrufe einzufügen. Falls Variablen mit derselben Bedeutung, aber unterschiedlicher Bezeichnung, in den Studiendaten auftauchen, können sie mit dem Befehl: *"= variable name study"* umbenannt werden. Die Umbenennung sollte in der entsprechenden Zeile, der Zielvariablen, der IDB vorgenommen werden. Oft kommen auch Code- und Decode-Paare in der IDB vor. Das Makro fügt automatisch PUT-Statements in die Zellen ein:

variable_decode=put(variable_code, codelist.);

Informationen über Missings (vollständig leere Variableninhalte) werden vom Check-Makro gefunden und in einer Kommentarspalte eingefügt.
In einigen Fällen muss eine bestimmte Reihenfolge beim Ausführen des späteren SAS Codes eingehalten werden. Dies kann in einer entsprechenden Order-Spalte berücksichtigt werden. Die Schritte werden entsprechend nummeriert. Zusätzlich beschreibt eine weitere Spalte *Derivation Method*, ob es sich bei der Ableitungsregel um einen Data Step oder Makroaufruf handelt.
Nachdem alle Ableitungsregeln in allen Domänen vervollständigt wurden, erfolgt ein Review-Schritt. Ein zweiter statistischer Programmierer bzw. ein Statistiker kontrolliert die Spezifikation gegen den SAP.
Mit der fertigen Spezifikation kann die Erzeugung der Mapping-Programme gestartet werden. Der folgende Abschnitt beschreibt, wie die Excel-Spezifikation wieder in SAS

eingelesen wird und wie aus den einzelnen Domänenbögen per Makro SAS-Programme erzeugt werden. Sie vollführen später das eigentliche Mapping.

2.3 Mapping-Programmerzeugung

Der finale Schritt des vorgestellten Konzepts ist die automatische Erzeugung von Mapping-Programmen. Diese Programme werden *Programm-Templates* genannt, da sie in einem weiteren Bearbeitungsschritt angepasst werden können, falls nicht alle Ableitungsregeln detailliert (per Programmcode) in der Spezifikation beschrieben wurden. Der eingefügte Code könnte später nochmals in die Spezifikation eingetragen werden. Dies wäre bei einer Erstellung der Programmdokumentation wichtig, da die gesamte Ableitung (Mapping) für die Behörden dokumentiert werden muss.
Aufgrund von Validierungs-SOPs wird jedes Mapping-Programm validiert – entweder per Code-Review oder durch Doppelprogrammierung.

Im nächsten Abschnitt schauen wir uns an, wie die einzelnen Mapping-Programme aus der Excel-Spezifikation mit Hilfe des Makros %m_create_mapping erzeugt werden. Der Header des Makros %m_create_mapping ist mit den verschiedenen Makroparametern in Beispiel 4 dargestellt.

```
%macro m_create_mapping (
     project=          /*Project name                              */
    ,studyid=          /*Study ID                                  */
    ,domain=  /*Domains (separated by '#',e.g. ADSL#ADEX#ADTS)*/
    ,path=    /*Path to output folder of the programs             */
    ,userid=          /*User ID                                   */
    ,specification=   /*Specification filename                    */
    ,spec_path=       /*Path to specification                     */
    ,filenamepattern= /*Pattern of the filnename                   */
);
```

Beispiel 4: Makro Definition %m_create_mapping

Im Makroaufruf zu %m_create_mapping werden projektrelavante Informationen (Projekname und Studie) übergeben sowie auch die Angaben, für welche Domänen Mapping-Programme erzeugt werden sollen und welche Spezifikation (Name der Excel-Spezifikation und Pfad) eingebunden werden soll. Die aufzurufenden Domänen werden per ‚#'-Symbol getrennt. Die UserID des Programmierers wird eingetragen. Über den Parameter *filenampattern* kann ein Muster für den zu generierenden Programmnamen zu jeder Domäne definiert werden.

Das Makro %m_create_mapping besteht grundsätzlich aus einer Schleife durch alle Domänen, die im Makroaufruf zugewiesen wurden. Es ist möglich, alle in der Excel-Spezifikation gegebenen Domänen zu durchlaufen. Entsprechender SAS Programmcode wird automatisch erstellt.

Schleife über alle <*Domänen*>

1. Lies die Excel Spezifikation
2. Erzeuge einen Header des Mapping-Programms
3. Einfügen von Ableitungen (Data Step / Makros) und Kommentare in das Mapping-Programm
4. Nachbearbeitung

Ende

In einer Schleife wird zunächst die Excel-Spezifikation in SAS eingelesen und für jede Domäne ein Datensatz erstellt, der alle Informationen des entsprechenden Excel-Bogens enthält. Ein Auszug aus dem Makro %m_create_mapping ist in Beispiel 5 wiedergegeben.

In den folgenden Schritten des Makros %m_create_mapping werden die verschiedenen Teile des Mapping-Programms erstellt. Der Programm-Header wird über das Makro %template_header erzeugt. Falls Änderungen am Programm-Header aufgrund von SOP Vorgaben oder Programmierrichtlinien notwendig werden, kann dies einfach im Makro implementiert werden.

Alle Ableitungen und Mappingvorschriften werden über das Makro %template_derivations in das Mapping-Programm als SAS Code eingefügt. Es durchläuft hauptsächlich die verschiedenen Zeilen der Derivation Spalte und fügt den SAS Code in das künstlich erstellte Programm ein. Zusätzlich werden Kommentare zu jedem SAS Code eingefügt, die eine Beschreibung der Ableitung enthalten. Dies vereinfacht das spätere Review des Programmcodes.

Makroaufrufe in der Derivation Spalte erzeugen auch Makroaufrufe im resultierenden Mapping-Programm. Das Makro %template_derivations folgt hierbei der Reihenfolge der Ableitungen, die in der Spezifikation eingetragen wurden bzw. die in der Order Spalte hinterlegt ist. Einfache Umbenennungen von Variablen, werden am Anfang des Mapping-Programms als Rename-Befehle eingefügt.

Einer der letzten Schritte des Makros ist das Hinzufügen der Metadatenstruktur der jeweiligen Domäne mit finalen Sortierbefehlen mit einer Bereinigung des Work-Verzeichnisses und des Löschens von verwendeten Makrovariablen.

Grundsätzlich kann die Struktur eines Mapping-Programms an Firmenrichtlinien und Kundenwünsche angepasst werden.

```
%varcount(list=&domain);
%do d=1 %to &var_num;
        %let _domain= %left(%qscan(&domain.,&d,"#"));
        %read_meta_spec(domain=&_domain,
                        path=&spec_path.,
                        filename=&specification);
        %template_header(domain=&_domain,
                         studyid=&studyid,
                         project=&project,
                         path=&path,
                         userid=&userid,
```

```
                        filenamepattern=&filenamepattern.);
        %template_set_ds(domain=&_domain);
        %template_set_keyvar(domain=&_domain);
        %template_in_data(domain=&_domain);
        %template_derivation(domain=&_domain);
        %template_sortorder(domain=&_domain);
        %template_drop_in_var(domain=&_domain);
        %template_add_dummy;
        %template_clean_lib;
        %template_eof;
%end;
```

Beispiel 5: Makro %m_create_mapping (Auszug)

Im nächsten Abschnitt werden einige der verwendeten Sub-Makros beschrieben, die vom Makro %m_create_mapping aufgerufen werden.
Das erste Makro liest die aktuelle Domäne (&_domain) aus der Excel Spezifikation ein. Hierbei wird der Domänen-Bogen mit der Metadatenstruktur der IDB (*target*), den Studienmetadaten (*source*) und den Ableitungen (*derivations*) eingelesen. Das Makro %read_meta_spec erzeugt einen SAS Datensatz mit diesen Informationen.
Über das Makro %template_header (Beispiel 6) wird das Mapping-Programm angelegt und notwendige Header-Informationen über das Projekt, die Studie, den Autor, den Programmnamen und den Pfad zum Ausgabeordner eingefügt. Der Header sollte an spezielle Firmen- oder Kundenvorgaben angepasst sein. Insgesamt besteht der Header im Mapping-Programm aus SAS Kommentaren.
Diese Kommentare werden über simple DATA _NULL_ und PUT-Befehle erzeugt. Die verschiedenen PUT-Befehle sind in Beispiel 6 dargestellt.

```
%macro template_header(
    domain= /*domain                                        */
   ,studyid=/*Study ID                                      */
   ,project=/*Project/Therapeutic area/Compound             */
   ,path=   /*Path to output file                           */
   ,validation=double programming   /*Validation method     */
   ,userid= /*User ID or Name of the programmer             */
  , filenamepattern=p-template- /*Filename pattern          */
);

 filename _templ "&path/&filenamepattern.&domain..sas";

 data _null_;
    file _templ old;
    put '/*-------------------------------------------------*/';
    put '/* The Pharma Company                               */';
    put '/*-------------------------------------------------*/';
    put "/* Studyname   : &studyid                          */";
    put "/* Project     : &project                          */";
    put "/* Program name: &p-&targetdomain..sas;           */";
    put "/* Purpose     : map dataset &targetdomain. to IDB*/";
    put '/* Template    :                                   */';
```

```
    put '/* Validation method  : &validation               */';
    put "/* Author name        : (&userid.)                */";
    put "/* Date completed     : &sysdate9.                */";
    put '/* Updated by         : (Name) - (Date):          */';
    put '/*-------------------------------------------------*/';
 run;
%mend;
```

Beispiel 6: Makro %_template_header

Im nächsten Schritt geht das Makro durch alle Ableitungsregeln, die in der Excel-Spezifikation angegeben sind. Diese wurden im vorherigen Schritt schon als SAS Datensatz für die aktuell zu bearbeitende Domäne erzeugt.
Die Ableitungsregeln sind mit einem derivation_flag=‘Y‘ markiert. Die orange hervorgehobenen Felder zeigen auch an, dass die entsprechende Variable der Domäne abgeleitet werden muss. Der Ableitungs-Flag (*derivation_flag*) zeigt dem Makro an, dass der spezielle SAS Code der Derivation Spalte in das Mapping-Programm eingefügt werden muss. Vor jeder Ableitung wird vom Makro die Information über den Variablennamen, das Label und ein Kommentar im Mapping-Programm eingefügt. Es wird als SAS Kommentar eingetragen. Dies geschieht wiederum mit Hilfe von DATA _NULL_ und PUT-Befehlen. Einen Auszug aus dem Makro ist in Beispiel 7 wiedergegeben.

```
data _null_;
    set _deriv01;
    file _templ mod;
    put '/*************************************************/';
    put '/*' sasname '(' label '): ' descript '*/';
    put '/*************************************************/';
    put  derivation;
run;
```

Beispiel 7: PUT-Befehle erzeugen Kommentare und die Ableitungsregeln

In Abbildung 7 sind Ableitungsregeln (*Derivation*) für die Domäne ADSL auszugsweise dargestellt. Für den Benutzer der Spezifikation zeigen die grünen Felder an, dass die Variablen schon in den Quelldaten vorhanden sind.
Manchmal haben die Variablen der Quelldaten denselben Namen wie in der Zieldatenstruktur, aber sie können eine andere Bedeutung haben. In diesem Fall müssten die Variablen der Quelldaten verändert werden, obwohl sie existieren. Dies könnte mit dem *Derivation-Flag* angezeigt werden. Orange Farben zeigen an, dass die Variable nicht in Quelldaten existiert und abgeleitet werden muss. Falls eine Variable der Quelldaten nicht in der Zieldatenstruktur vorkommt, wird sie rot markiert. Dies könnte bedeuten, dass die Variable umbenannt werden muss. Umbenennungen werden nicht automatisch erkannt. Ein Umbenennungsbefehl (*Rename*) sollte manuell in die entsprechende Derivation Zeile der Zielvariablen eingetragen werden. Es könnte aber auch bedeutet, dass die rot markierte Variable nicht mit in die Zieldatenstruktur übernommen werden muss.

L	M	N	O	P	Q	R	S	U	V	W	X
Source Study Metadata							Derivation				
Studyid	Domain Source	Var_study	Label Source	Type study	Format study	Length	Comment	Derivation	Derivation_order	Derivation_type	derivation_flag
99999	DM	SAFETY	Population	C	$	2	[not in metadata]		.		
99999	DM	SAFETYLT	Safety	C	$	2	[not in metadata]		.		
99999	DM	SAFETYN	Population	N	NY	8	[not in metadata]		.		
99999	DM	SAFSUR	Population	C	$	2	[not in metadata]		.		
99999	DM	SAFSURN	Population	N	NY	8	[not in metadata]		.		
99999	DM	SFETYLTN	Safety	N	NY	8	[not in metadata]		.		
99999						.			.		Y
99999	DM	STUDYID	Identifier	C	$	10		ADSL.STUDYID	.		
99999	DM	USUBJID	Subject	C	$	20		ADSL.USUBJID	.		
99999	DM	AGE	Age	N		8		ADSL.AGE	.		
99999						.	[AGEGR1N]	AGEGR1=put(AGEGR1N, Z_AGEGRP.);	2	[data step]	Y
99999						.		%_adsl_agegr1(age);	1	[macro]	Y
99999						.	[AGEGR2N]	AGEGR2=put(AGEGR2N, Z_AGEGRP.);			Y
99999						.		%_adsl_agegr2(age);		[macro]	Y
99999						.	[AGEGR3N]	AGEGR3=put(AGEGR3N, Z_AGEGRP.);	3		Y
99999						.		%_adsl_agegr3(age);		[macro]	Y
99999						.	[AGEGR4N]	AGEGR4=put(AGEGR4N, Z_AGEGRP.);	.		Y
99999						.		%_adsl_agegr4(age);		[macro]	Y

Abbildung 7: Ableitungen mit Data Step und Makro-Aufrufen in ADSL

Beispiel 8 zeigt einen Auszug aus einem automatisch generierten Mapping-Programm mit einfachen Ableitungen im Data Step. Vor jedem SAS Befehl wird vom Makro ein Kommentar eingetragen. Der Variablenname und die zugehörige Beschreibung aus der Spezifikation werden in das Mapping-Programm eingefügt.

```
data &dsn.;
    set &dsn.;

 /*************************************************************/
 /*AGEGR1 (Age Group 1 ): decode of AGEGR1N                   */
 /*************************************************************/
 AGEGR1=put(AGEGR1N, Z_AGEGRP.);

 /*************************************************************/
 /*UASR (Unique Subject Identifier/Age/Sex/Race ):            */
 /*************************************************************/
 uasr=cat(scan(uasre,1,'/'),'/',scan(uasre,2,'/'));
run;
```

Beispiel 8: Generierter SAS Code basierend auf der Spezifikation

Grundsätzlich kann man komplizierte Ableitungen in Makros auslagern, da diese einfacher in der Spezifikation eingetragen werden können. Das ist auch für den Nutzer einfacher zu lesen. Im resultierenden Mapping-Programm wird nur der Makro-Aufruf automatisch eingefügt. Natürlich müssen die verwendeten Makros validiert werden. Ebenso könnte man generelle Makros für das Projekt erstellen, die auch in anderen Studien für das Mapping verwendet werden können. Es ist auch möglich, nur eine beschreibende Erklärung zur Ableitung in die Spezifikation einzutragen. Dies würde dann als reiner Kommentar im Mapping-Programm eingetragen. Der Programmierer müsste diesen Kommentar später manuell in Programmcode umsetzten. Das folgende Beispiel zeigt einen Ausschnitt aus einem generierten Mapping-Programm, wobei hier SAS Makro-Aufrufe mit den zugehörigen Kommentaren eingefügt wurden.

```
/*****************************************************************/
/*AGEGR2N (Age Group 2 (N) ): Age group2  is derived from age in  */
/*        categories specified in the SAP                        */
/*****************************************************************/
%_adsl_agegr2(age);
/*****************************************************************/
/*BASEBMI (Baseline Body Mass Index (kg/m2) ):  is baseline BMI   */
/*        given in vital signs                                   */
/*****************************************************************/
%_adsl_bmi;
/*****************************************************************/
/*BASEHEIG (Baseline Height (cm) ):   is baseline height given in */
/*         vital signs                                           */
/*****************************************************************/
%_adsl_heigth;
/*****************************************************************/
/*BRTHDTL (Birth Date ): Birth Date for Listings                  */
/*****************************************************************/
%_adsl_brthdtl;
```

Beispiel 9: SAS Makros automatisch im Mapping-Programm eingefügt

3 Diskussion

In einigen Beispielprojekten konnte der vorgestellte semi-automatische Prozess der Spezifikationserstellung mit folgender Mapping-Programmerzeugung den Programmieraufwand verkürzen. Üblicherweise arbeiten mindestens zwei Programmierer am Mapping der Daten in die integrierte Datenbank. Ebenso konnte die Zeit verkürzt werden, eine Spezifikation zu erstellen. Die meiste Arbeit, die Zusammenstellung der Metadaten und der Vergleich von Studien- und IDB Datenstruktur wird vom Makro %m_create_specification übernommen. Es übernimmt die ermüdende Arbeit, alle Domänen und Variablen mit den Attributen in die Spezifikation einzutragen. Dies geschieht vollständig automatisch.

Ein Check-Makro, welches die Studiendatenstruktur mit den Metadaten der IDB vergleicht, findet Treffer zwischen den Variablen und zeigt an, welche Variablen abgeleitet und welche möglicherweise umbenannt werden müssen. Projektspezifische Makros, die man in verschiedenen Studien verwendet, können automatisch in die Spezifikation eingebunden werden. Schließlich sollte eine Statistiker oder zweiter statistischer Programmierer die Spezifikation gegen die Vorgaben (SAP, Data Handling Document) vergleichen. Nachdem alle fehlenden Ableitungen in die Spezifikation eingetragen wurden, ist die Generierung der Mapping-Programme ein automatischer Prozess. Das Makro **%m_create_mapping** erzeugt alle Programme, der in der Spezifikation angegebenen Domänen und fügt jeglichen SAS Programmcode mit entsprechenden Kommentaren ein. Ein weiterer Vorteil ist, dass die verwendeten Sub-Makros (von %m_create_mapping) leicht an Firmenvorgaben, SOPs oder Programmierrichtlinien angepasst werden können.

Die erzeugten Mapping-Programme haben den gewünschten Header mit allen Projektinformationen, die Ableitungen auch notwendige Kommentare. Der generierte SAS Code

ist einfach zu lesen, da oft auch Makroaufrufe verwendet werden, die schwierigere Ableitungen auslagern. Diese Makros müssen allerdings konventionell programmiert werden.
Falls Änderungen im Programm notwendig sind, könnten diese direkt in der Spezifikation vorgenommen werden. Die Spezifikation könnte somit auch für eine Dokumentation des gesamten Mapping-Prozesses verwendet werden.
Der Mapping-Prozess wird transparenter, ist einfacher nachzuvollziehen und zu überprüfen.
Natürlich gibt es schon verschiedene Ansätze der Automatisierung des Mapping-Prozesses. Ein Vorteil des vorgestellten Verfahrens ist, dass nicht nur die Mapping-Programme erstellt werden, sondern auch die Spezifikation in einem semi-automatischen Prozess. Manche Methoden können auch nur bestimmte Datenstrukturen abbilden, wie z.B. von SDTM zu ADaM. Mit der vorgestellten Methode kann man jegliche Quelldatenstruktur in irgendeine Zieldatenstruktur abbilden. Hiermit können wir Datenstrukturen von A nach B überführen.

4 Zusammenfassung und Ausblick

Es wurde ein Ansatz vorgestellt, wie man den Mapping-Prozess von klinischen Studiendaten in eine integrierte Datenbank automatisieren kann. Ein Ergebnis dieses Prozesses ist eine semi-automatisch erzeugte Spezifikation. Zusätzlich werden die Mapping-Programme vollständig automatisch generiert.

In einer weiteren Stufe des Prozesses könnten weitere Daten-Checks implementiert werden. Dem Nutzer können Informationen über notwendige Änderungen, Rückmeldungen zu Problemen oder Auffälligkeiten in den Daten gemeldet werden. Auch könnte die Implementation der Define.xml oder Define.pdf Dokumente für eine Submission eingebaut werden. Zusätzlich wäre es möglich, den Prozess der Spezifikationserstellung und Anpassung der Mapping-Regel in SAS per Makro-Aufruf zu steuern. Dies wird von Pharmaunternehmen favorisiert, da sich Prozesse in SAS einfacher validieren lassen.

Literatur

[1] http://www.cdisc.org.