

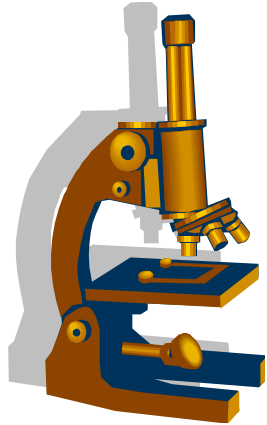


Use the define.xml Standard from the Start for CRF design

CDISC Interchange, Baltimore, 12 Oct 2011
Dr. Philippe Verplancke, XClinical GmbH



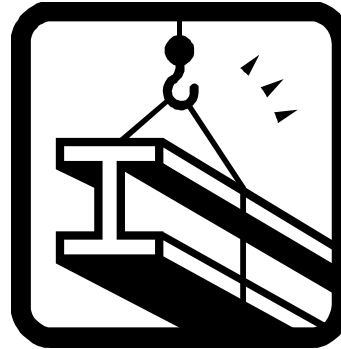
CDISC standards serve many purposes



Terminology

How do we call things?

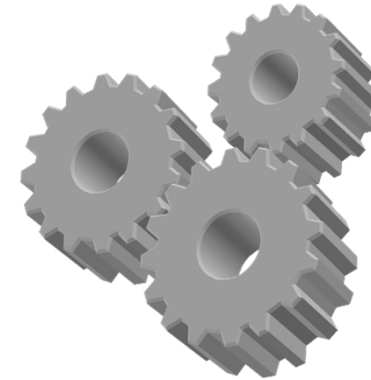
- **CDISC SHARE
Metadata Repository**



Structure

Where do we put things?

- Protocol (PRM)
- CRF pages (**ODM**)
- Lab forms (LAB)
- Data tables
(SDTM, **define.xml**)
- Analysis tables (ADaM)

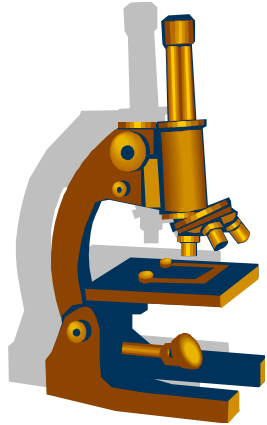


Format

How do computers store and exchange things?

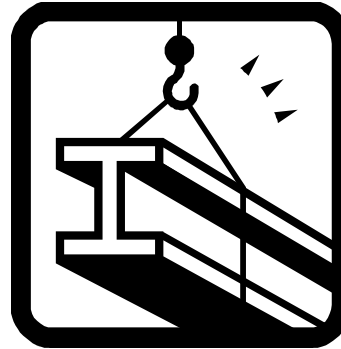
- **XML**
- ~~ASCII~~
- ~~SAS~~

But often we use only one or two of the CDISC standards...



Terminology

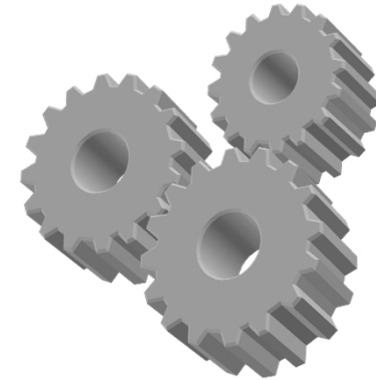
How do we call things?



Structure

Where do we put things?

- SDTM
- ADaM



Format

How do computers store and exchange things?

- SAS

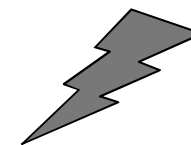
Something is wrong ...



How we worked before

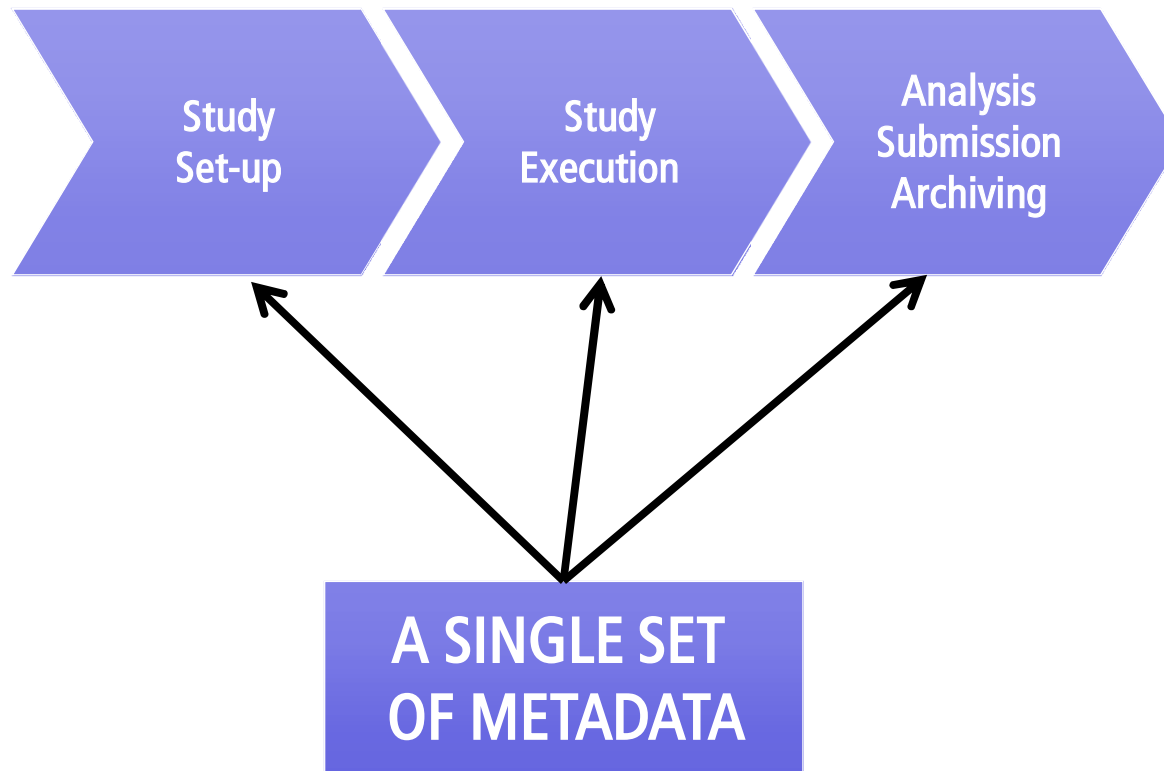


How we work now....



CDISC SDTM
CDISC define.xml

The way we should work... applying CDISC end-to-end

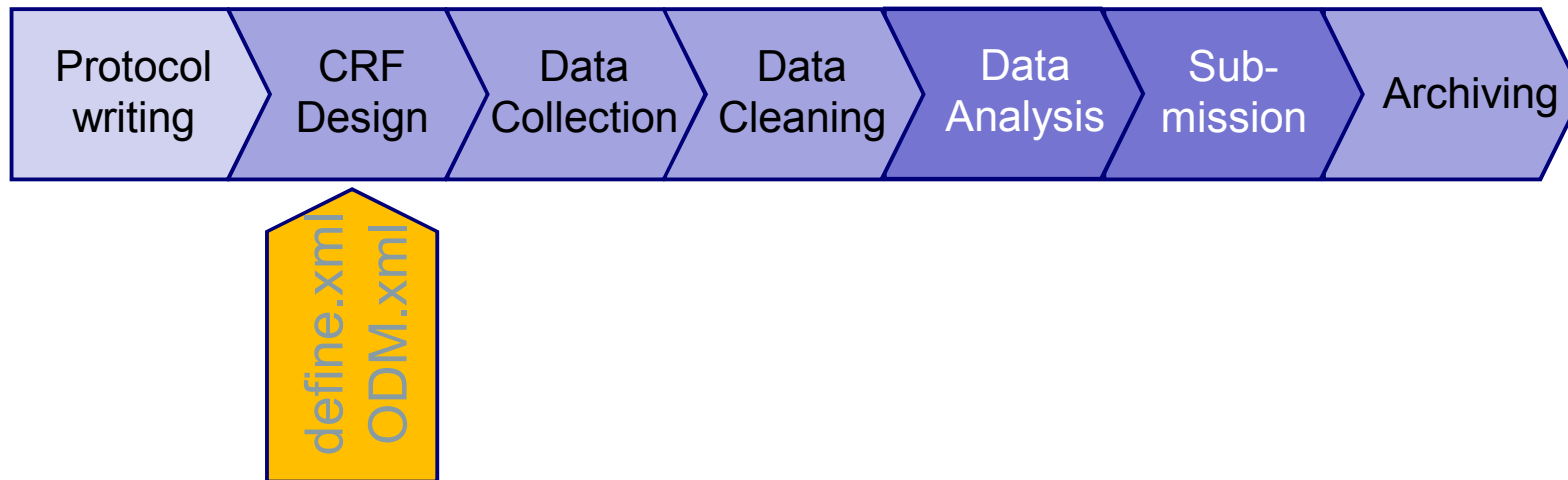


The way we should work... applying CDISC end-to-end



How?

The way we should work...
inject CDISC at the start of the process!



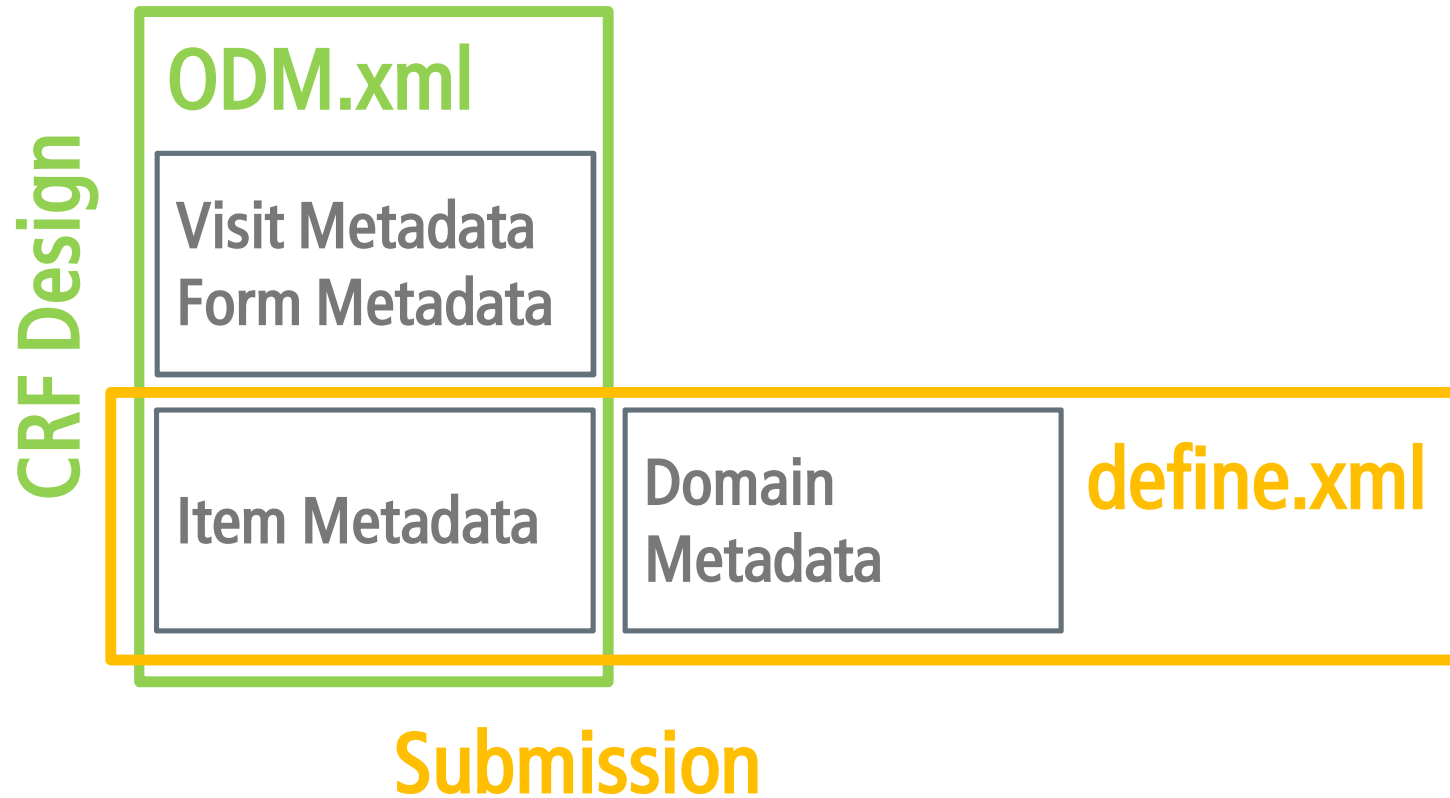
define.xml is not an output, it is an input of the CDISC process!

Why use CDISC standards at the start, for CRF design?



- Save a lot of time and costs by
 - Avoiding legacy data transformation at the end
 - Setting up the CRF based on standards
 - Auto-configuring an EDC system
(no need for database programmers)
- Generate all kinds of documentation (annotated CRF, list of items, list of codes, etc.) automatically
- Make the CRF design independent of any EDC system
- Specify the CRF design such that multiple CROs can work with it and you can be sure they'll deliver data in exactly the format you want

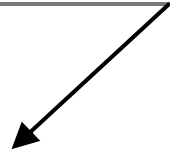
It's easy to start: define.xml and ODM.xml overlap!



Variable-level Metadata in define.xml



```
<ItemDef OID="VS.VSTESTCD" Name="VSTESTCD" DataType="text"
  Length="8"
  Origin="CRF Page"
  def:Label="VITAL SIGNS TEST SHORT NAME">
  <def:ValueListRef ValueListOID="VL.VSTESTCD"/>
</ItemDef>
```



```
<def:ValueListDef OID="VL.VSTESTCD">
  <ItemRef ItemOID="VS.HEIGHT" OrderNumber="1" Mandatory="Yes"/>
  <ItemRef ItemOID="VS.WEIGHT" OrderNumber="2" Mandatory="Yes"/>
  <ItemRef ItemOID="VS.BMI" OrderNumber="3" Mandatory="Yes"/>
  <ItemRef ItemOID="VS.SYSBP" OrderNumber="4" Mandatory="Yes"/>
  <ItemRef ItemOID="VS.DIABP" OrderNumber="5" Mandatory="Yes"/>
  <ItemRef ItemOID="VS.POS" OrderNumber="6" Mandatory="Yes"/>
  <ItemRef ItemOID="VS.PULSE" OrderNumber="7" Mandatory="Yes"/>
</def:ValueListDef>
```

Item Metadata in define.xml is identical with Item Metadata in odm.xml

```
<ItemDef OID="VS.SYSBP"
```

```
  Name="Systolic Blood Pressure"
```

```
  Name="SYSBP"
```

define.xml

```
  def:Label="Systolic Blood Pressure" SDSVarName="SYSBP"
```

```
  DataType="integer"
```

```
  Length="3"
```

ODM.xml

```
  SASFieldName="SYSBP">
```

```
    <Question>
```

```
      <TranslatedText xml:lang="en">Systolic Blood Pressure</TranslatedText>
```

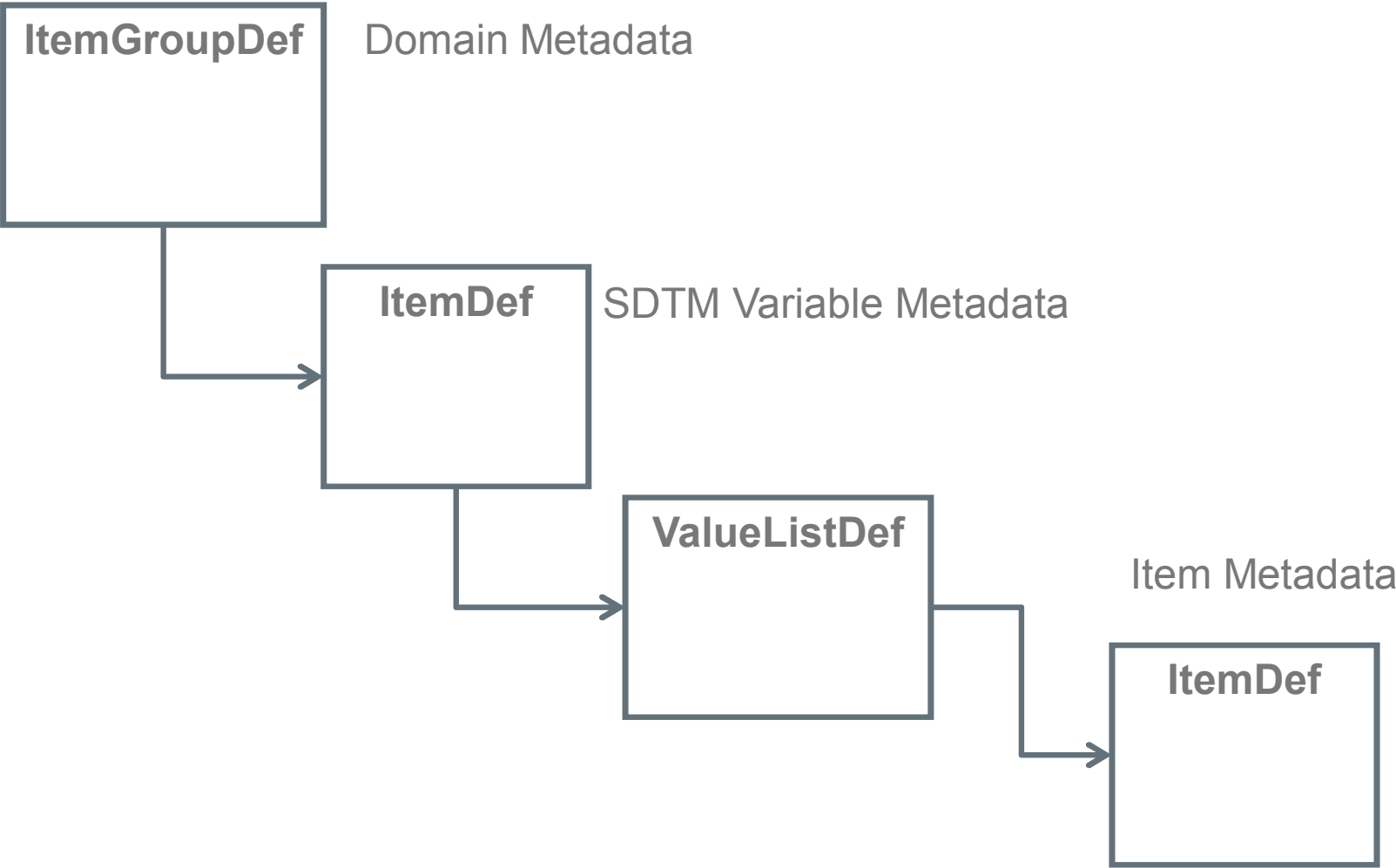
```
      <TranslatedText xml:lang="de">Systolischer Blutdruck</TranslatedText>
```

```
    </Question>
```

```
    <MeasurementUnitRef MeasurementUnitOID="MU.mmHg"/>
```

```
</ItemDef>
```

Metadata are represented as XML elements

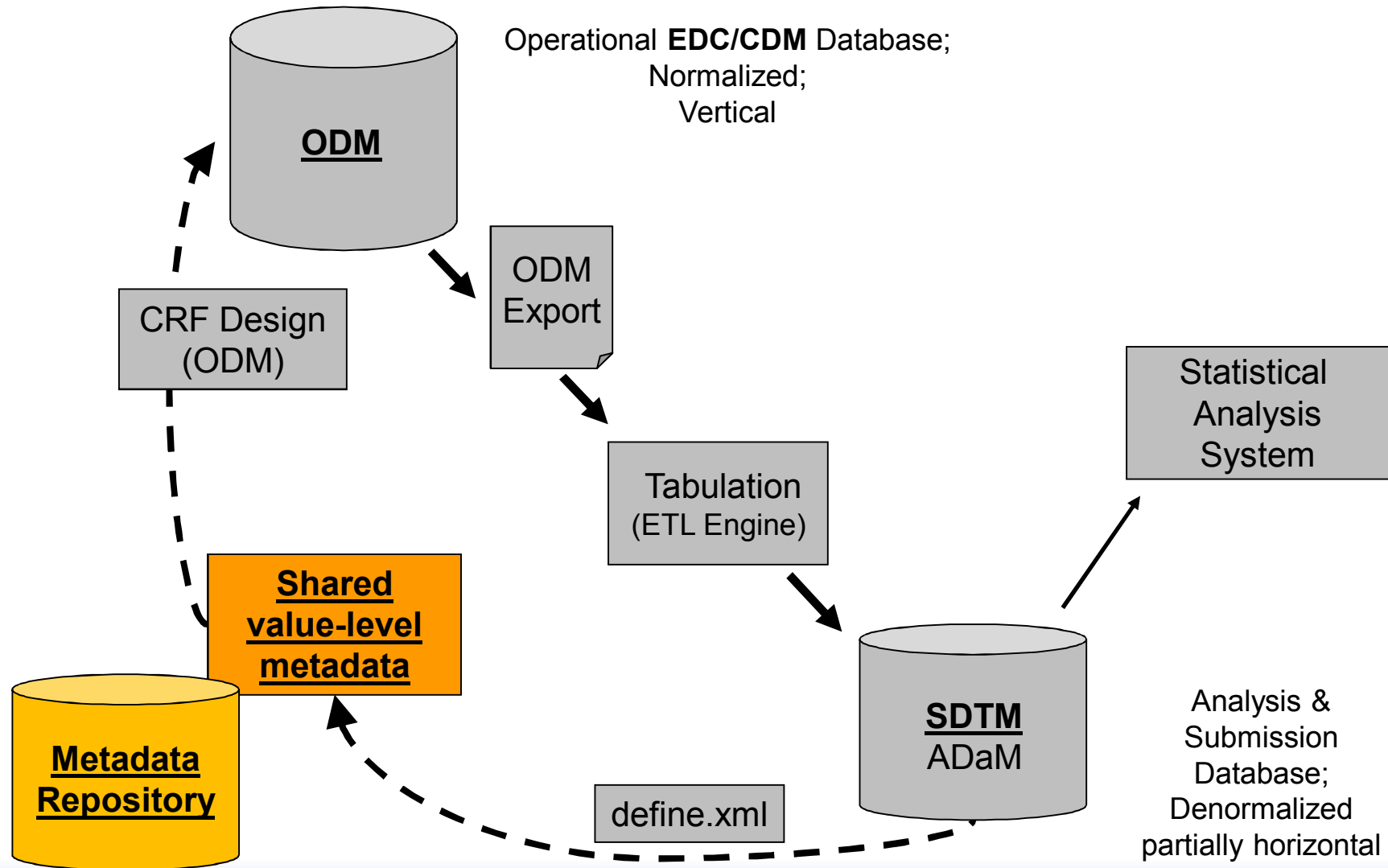


Checklist for define.xml



- Does your define.xml contain the right datatypes for value-level metadata?
 - E.g., VS.SYSBP = Integer, not Text!
- Does your define.xml use normalized Measurement Units and are those actually attached to the value-level ItemDefs?
 - E.g., MU.mmHg instead of MU.SYSBP?
- Does your define.xml attach codelists to value-level ItemDefs?
- Does your define.xml contain question texts for value-level metadata?

End-to-end Flow of Data and Metadata



So what about CDASH?

- Re-use your define.xml value-level metadata to build the CRFs for your next trial and that trial will be CDASH compliant!
- Why? Because your value-level metadata will comply with the SDTM standard.
- P.S.: a number of EDC systems are certified by CDISC to be able to import XML metadata such that the CRF building is automatic.

Ideas about how to help with CDISC SHARE



- Donate your define.xml files to CDISC, they contain valuable metadata which is already SDTM-compliant
- So the only work that is needed will be the harmonization between donated sets of metadata
- Automatic tools can be applied to the XML repository to find probable duplicates, which will very much speed up the process.
- An XML-based metadata repository is technically extremely simple; it is just a file
(see e.g., <http://lillyodmlibrary.codeplex.com/>)
- P.S. The odm.xml file with CDISC controlled terminology that you can download from the NCI EVS does not contain ItemDefs yet...



Questions?

pv@xclinical.com

Philippe Verplancke

XClinical GmbH
Arnulfstrasse 19
80335 Munich

Germany

Tel: +49 (0)89 / 45 22 77 – 50 00

Fax: +49 (0)89 / 45 22 77 – 59 00

www.xclinical.com
info@xclinical.com

