Drug Disease Model Resources

ddmore

Internal Release

Standard Output (SO)

Format Specification for Version 0.3.1

Nadia Terranova[1], Marc Lavielle[2], Mike K Smith[3], Emmanuelle Comets[4], Kajsa Harling[5], Rikard Nordgren[5], Duncan Edwards[6], Andrew Hooker[5], Celine Sarr[7], France Mentré[4], Florent Yvon[8], Maciej J Swat[8]

[1] *Merck Institute for Pharmacometrics, Merck Serono, Lausanne, Switzerland*
[2] *Inria Saclay, Paris, France*
[3] *Global Clinical Pharmacology, Pfizer, Sandwich, UK*
[4] *INSERM, IAME, UMR 1137, Paris, France &*
University Paris Diderot, IAME, UMR 1137, Sorbonne Paris Cité, Paris, France
[5] *Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden*
[6] *Simcyp (a Certara company), Sheffield, UK*
[7] *SGS Exprimo NV, Mechelen, Belgium*
[8] *EMBL - European Bioinformatics Institute, Cambridge, UK,*

March 31, 2016

Contents

Chapter 1

Overview of changes

The following table summarises the major changes in 0.3.1.

PharmML element or modelling aspect	version $\leq$ 0.3	version 0.3.1
Column type declaration Table 3.2	supported with `columnType` attribute	– new values introduced: *varParameter*, *correlation*, *covariance*, *stdev*, *variance* – removed *varParameter_corr*, *varParameter_cov*, *varParameter_stdev*, *varParameter_var* – multiple values allowed
Higher variability levels Section 3.4	IOV (aka BOV) supported only	arbitrary levels are supported with new `level` attribute
Integer declaration	`type="xs:integer"`	`type="xs:int"`, see detailed discussion in the PharmML 0.8.1 spec, [Swat et al., 2016b]

Table 1.1: Overview of major differences between versions 0.3.1 and 0.3

Chapter 2

Introduction

The definition and implementation of standards enabling a reliable exchange of pharmacometric models and data across software tools is one of the key goals for efficient promoting collaborative drug and disease modelling and simulation (M&S) research. The IMI project Drug and Disease Model Resources (DDMoRe) project is building such an interoperability platform [Harnisch et al., 2013].

Two XML based exchange formats create the backbone of the platform. The Pharmacometrics Markup Language, PharmML, [Swat et al., 2015b], has been designed as the exchange medium for mathematical and statistical models across various target tools. The Standard Output (SO) represents a complementary tool-independent format for storing typical output produced in a pharmacometric workflow.

As a generic output model, SO aims at:

- providing a flexible storage structure for typical results of M&S analyses performed in any DDMoRe target tool;
- enabling effective data flow across tasks to ensure optimal interactions among software tools and, then, extend the modelling capabilities of the workflow;
- facilitating information retrieval for post-processing and reporting, by allowing immediate access to M&S results.

SO is a tool-independent exchange format, intended for storage of results in standardised form. It enables effective data exchange within complex workflows as well as supports users in assessing, reviewing and reporting modelling steps.

The first SO schema and specification in version 0.1, [Terranova et al., 2015], was based on two initial proposals, [Smith, 2014] and [Lavielle, 2014]. The current specification is an extension of v0.3 released in January 2016, [Terranova et al., 2016].

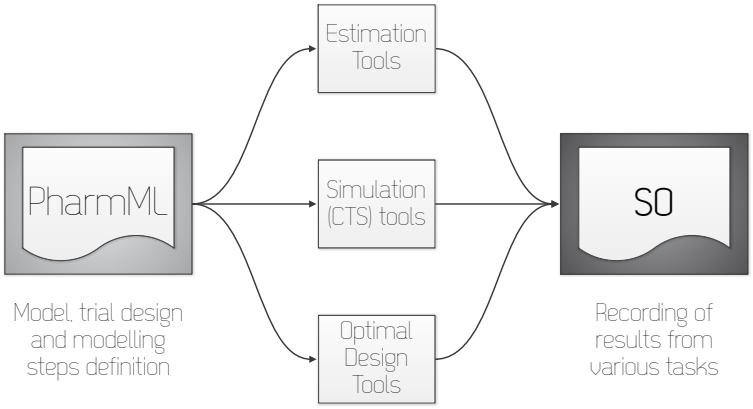

Figure 2.1: PharmML, target tools and SO.

2.1 Workflow support

One of the key benefits coming from SO usage will be the effective support for workflow interaction. As an example, it can be considered a typical pharmacometric workflow consisting of the following steps :

1. Data processing in R.
2. Model exploration in MlxPlore, [Lixoft Team, 2013].
3. Estimation of the model parameters and model refinement in Monolix[Lixoft Team, 2014] or NONMEM [Beal et al., 2009].
4. Performing bootstrap with PsN [Lindbom et al., 2004].
5. Exploring diferent trial designs in a clinical trial simulator using e.g. Simulx, [Lixoft Team, 2014].
6. Optimizing the design in PFIM [Bazzoli et al., 2010]/PopED[Nyberg et al., 2012].
7. Writing the report based on the well-documented steps of the workflow.

A simplified representation of this workflow is shown in Figure 2. Each step is accompanied by a PharmML file carrying on the entire input information and enabling the task execution as well as a SO file storing the relevant output coming from the respective pharmacometric analysis and making it available to any subsequent task.

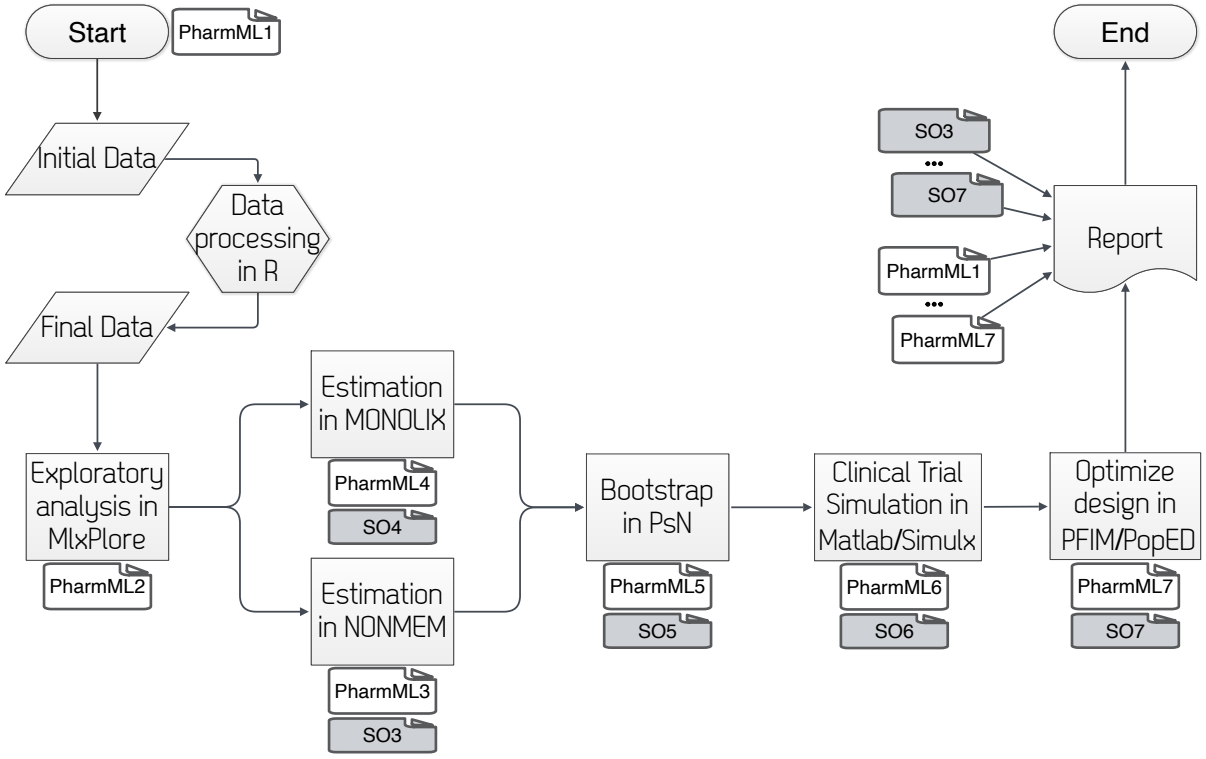

Figure 2.2: A schematic view of a typical workflow in pharmacometric analysis. PharmML/SO files (here defined as *PharmML1*, ..., *PharmML7* and *SO3*, ..., *SO7*) are accompanying every task by carrying the entire information necessary to execute it and to store the relevant outputs, respectively.

2.2 PharmML & SO

Despite being complementary to PharmML, SO doesn't have necessarily to refer to a related PharmML file. Nevertheless, even though the SO schema is a separate one, it is not fully independent from the PharmML schema from which it reuses structure elements for the specification of matrices, datasets, external file

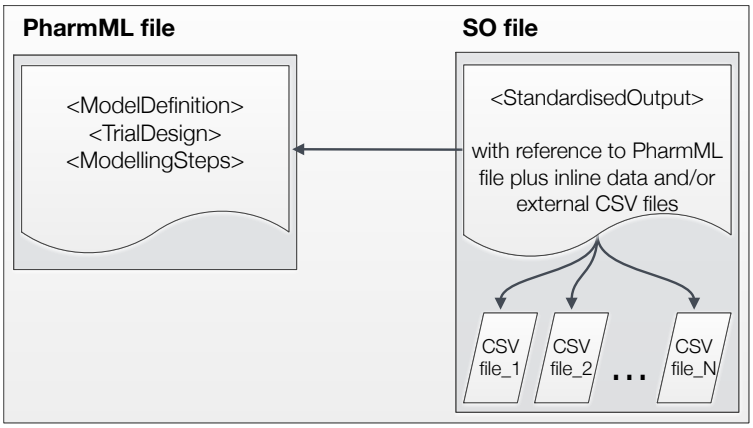

Figure 2.3: Connection between PharmML and SO.

references and other basic types such as *string* and *scalar*. A relationship between SO and PharmML is shown in figure 2.3.

Different versions of SO depend on different PharmML versions. The table 2.1 contains an overview of the relationship.

Version	Release date	Description	PharmML version
0.1	16 February 2015	Sections 1-6 covered	0.6
0.2	22 April 2015	Optimal Design support added	0.6
0.3	18 January 2016	Changes and improvements	0.8
0.3.1	16 March 2016	Support of higher variability levels	0.8.1

Table 2.1: SO-PharmML dependence overview.

2.3 Working with SO – libSO (API)

LibSO is a JAVA Application Programming Interface (API) for creation and validation of SO data XML documents. The API is available via SourceForge within the libPharmML project (`https://sourceforge.net/projects/libpharmml.ddmore.p`). This library depends on libPharmML and uses the same design pattern. A stand-alone validator is also provided, executable via command line. Documentation is available on the libPharmML wiki (`https://sourceforge.net/p/ddmore/libpharmml/wiki/Home/`).

Chapter 3

Population of SO

3.1 Data formats

The dataset support is based on the PharmML schema which is described in detail in the format specification, [Swat et al., 2015c]. The following list gives an overview of the data types required to populate a typical SO file

- `Table` with named columns using either
  - inline stored data or
  - reference to an external data file, e.g. *.csv or else.
- `Matrix` with named columns and rows.
- `File reference` – path to an external file relative to the current SO file (within the Combine archive).
- `Distribution` and its parameters using ProbOnto, [Swat et al., 2016a].
- `Sample` – empirical distributions using ProbOnto, see [Swat et al., 2016a] for more details.
- `Scalar` values – integer, real.
- `String` – any array of words.

Examples of the use of each format can be found in the next section.

3.2 Encoding of missing data

Encoding of missing data is supported in PharmML both for inline or external storage of data records since version 0.7, [Swat et al., 2015a]. Because the dataset support in SO is based on that in PharmML, the SO can reuse it in full extend as described in this section. For the sake of clarity, the description of features available in PharmML v0.7 is reported hereafter and slightly adapted for the use in SO.

3.2.1 Inline stored data

We reuse symbols for special numerical values from R (see is.finite{base}) and SAS (see SAS/IML(R) 9.22)[1] to provide means to encode missing data.

When data is stored inline within PharmML, few new predefined XML elements are required. Here the list of typical missing data types and corresponding elements introduced in this version:

- **NA** – not available/missing data, `<NA>`
- **NaN** – not a number – impossible values (e.g., dividing by zero), `<NaN>`
- **+Inf**/**-Inf** – positive/negative infinity, `<plusInf>` and `<minusInf>`

[1] Other possible codes are **DEE** or **-99** (SAS) – data entry error or **SRA** or **-999** (SAS) – subject refused answer, see `http://www.ats.ucla.edu/stat/sas/faq/how_to_code_missing_differently.htm`, but will not be introduced unless required

- **BLQ/ALQ** – below/above level of quantification[2], `<BLQ>` and `<ALQ>`

The following hypothetical dataset, Table 3.1, with several missing DV values can be encoded in PharmML

ID	TIME	DV
1	3.43	-99
1	5.2	48.03
1	42.13	L
1	52.63	+INF
1	57.53	72.3
...	...	...

Table 3.1: A dataset with examples of missing data.

by using elements shown in the following snippet:

```
<Definition>
    <Column columnId="ID" columnType="id" valueType="id" columnNum="1"/>
    <Column columnId="TIME" columnType="idv" valueType="real" columnNum="2"/>
    <Column columnId="DV" columnType="dv" valueType="real" columnNum="3"/>
</Definition>
<Table>
    <!-- SUBJECT 1 -->
    <Row><ct:Id>i1</ct:Id><ct:Real>3.43</ct:Real><ct:NA/></Row>
    <Row><ct:Id>i1</ct:Id><ct:Real>5.3</ct:Real><ct:Real>48.03</ct:Real></Row>
    <Row><ct:Id>i1</ct:Id><ct:Real>42.13</ct:Real><ct:BLQ/></Row>
    <Row><ct:Id>i1</ct:Id><ct:Real>52.63</ct:Real><ct:plusInf/></Row>
    <Row><ct:Id>i1</ct:Id><ct:Real>57.53</ct:Real><ct:Real>72.3</ct:Real></Row>
```

Note 1 Two of the elements described above, NA and ∞, were available before as child elements of `<Constant>` – now merged with other *missing values* for consistency. The current version is simpler as it does allow to specify the missing values directly, without a parent element.

3.2.2 External datasets

Similar strategy is followed if an external dataset contains *missing data* records and this information needs to be passed to the target tool. The following new elements are available

- `<MissingData>` element with two attributes
  - `dataCode` – any symbol used in the referenced dataset
  - `missingDataType` – one of {*NA*, *NaN*, *plusInf*, *minusInf*, *BLQ*, *ALQ* }, consistent with elements introduced in the previous section.

The dataset used previously, table 3.1, is assumed now to be stored externally as *myFile.csv*, see `<path>` element below. The missing data codes used in the dataset are mapped to the allowed types as shown in the following code

```
<ds:Definition>
    <ds:Column columnId="ID" columnType="id" valueType="id" columnNum="1"/>
    <ds:Column columnId="TIME" columnType="idv" valueType="real" columnNum="2"/>
    <ds:Column columnId="DV" columnType="dv" valueType="real" columnNum="3"/>
</ds:Definition>
<ds:ExternalFile oid="extFile">
    <ds:path>myFile.csv</ds:path>
    <MissingData dataCode="-99" missingDataType="NA"/>
    <MissingData dataCode="+INF" missingDataType="plusInf"/>
    <MissingData dataCode="L" missingDataType="BLQ"/>
</ds:ExternalFile>
```

[2] **L/H** – symbols used in dataset in ADAPT5, [D'Argenio and Wang, 2009]

3.3 Data storage/encoding

This section is devoted to the most common case when data is stored in tabular form. Two options are available, either the data is stored directly in the XML format or in an external dataset. In both cases the columns along with optional headers, ignore line character and/or missing data symbols need to be defined. The declaration of the columns is in many cases SO specific and requires special attention. Other mentioned aspects are derived from PharmML and are described in the specification documents [Swat et al., 2015a] and [Swat, 2015].

3.3.1 Inline data

The example below shows how to declare and store typical estimation results in inline tables:

```
<Estimation>
    <PopulationEstimates>
        <MLE>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="POP_CL" columnType="popParameter" valueType="real" columnNum="1"/>
                <Column columnId="BETA_CL_WT" columnType="popParameter" valueType="real" columnNum="2"/>
                <Column columnId="PPV_CL" columnType="varParameter variance" valueType="real" columnNum="3"/>
                <!-- omitted other columns -->
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:Real>0.93882</ct:Real><ct:Real>1.63219</ct:Real><ct:Real>8.10703</ct:Real>
                    <!-- omitted other columns -->
                </Row>
            </Table>
        </MLE>
```

The dataset starts with `<Definition>` where the columns of the dataset table are defined. For each column the following attributes have to be specified

- `columnId` – the name of the column
- `columnType` (*optional*) – the type of the column identifies the role that the stored item plays in the model, see Table 3.2 for the explanation of allowed values. Multiple values are allowed.
- `valueType` – the type of each column value is specified, which complies with the PharmML type system.
- `columnNum` – the column number must start at 1 and each column must be numbered in consecutive order (i.e. 1,2,3,4... etc.).

Next the content of the dataset is held within the `<Table>` element and this consists of one or more `<Row>` elements. Each row must contain an entry for each column defined.

`columnType` value	Meaning
Applicable to PharmML & SO	
addl	number of additional doses
adm	type of administration
arm	study arm
censoring	left-censored data
covariate	covariates
cmt	target compartment
demographic	demographic type
dose	dose
duration	infusion duration
dv	dependent variable
dvid	dependent variable identifier
epoch	study epoch
evid	dose events
id	subject identifiers
idv (or time)	independent variable (time)
ii	inter-dose interval
limit	lower limit for interval-censored data

mdv	missing dependent variable
occasion	occasions
rate	infusion rate
reg	regression variable
replicate	in cases when subjects are assumed identical
ss	steady state
undefined	value for any undefined case
varLevel	level of variability
	Only SO related
indivParameter	individual parameter
popParameter	generic population parameter
randEffect	random effect
residual	residual
statPrecision	quantities expressing statistical precision
strataVariable	stratification variables
structParameter	structural parameters
varParameter	variability parameter
variance	variance
stdev	standard deviation
covariance	covariance
correlation	correlation

Table 3.2: A summary of allowed values for the `columnType` attribute in PharmML and SO. Note that the attribute is optional and should be used to provide information about the according column if required.

3.3.2 External datasets

As shown in the following listing, the reference to an external data set can be specified using the `<ExternalFile>` element, instead of `<Table>`, with a file path relative to the SO file. *Optional* parameters such as file format and delimiter can also be provided. The external file must be a valid input file for the target tool. The template for this reads:

```
<ds:ExternalFile oid="id1">
    <ds:path>FILE_PATH</ds:path>
    <ds:format>FORMAT</ds:format>
    <ds:delimiter>DELIMITER_TYPE</ds:delimiter>
</ds:ExternalFile>
```

As before the `<Definition>` defines the tabular content of the dataset. The listing below shows a simple but complete example how this works (optional attributed are omitted).

```
<ds:DataSet>
    <ds:Definition>
        <Column columnId="POP_CL" columnType="popParameter" valueType="real" columnNum="1"/>
        <Column columnId="BETA_CL_WT" columnType="popParameter" valueType="real" columnNum="2"/>
        <Column columnId="PPV_CL" columnType="varParameter variance" valueType="real" columnNum="3"/>
    </ds:Definition>
    <ds:ExternalFile oid="id1">
        <ds:path>warfarin_conc_pca.csv</ds:path>
    </ds:ExternalFile>
</ds:DataSet>
```

3.3.3 Using `columnType` attribute

Using the `columnType` attribute is often required for the downstream tools to makes sense out of the data stored in SO[3]. For example when storing parameters being variability measures, it is important to indicate the scale of such values. Then pair values such as `"varParam variance"`, `"varParam stdev"`, `"varParam covariance"`, `"varParam correlation"`, see Table 3.2, help to identify the value type.
Moreover, parameter related data might be stored in columns or rows, dependent on the case. In the following few typical examples are shown for the appropriate format and setting of the `columnType` attribute.

- Using `columnType="popParameter"`

[3]Although optional, it should, whenever possible, always be used and populated with available values listed in Table 3.2.

– precision of population estimates

Parameter	SE
POP_CL	0.49
POP_V	0.5
...	...
PPV_CL	0.25
PPV_V	0.62

```
<StandardError>
    <ds:Definition>
        <ds:Column columnId="Parameter" columnType="popParameter" valueType="string" columnNum="1"/>
        <ds:Column columnId="SE" columnType="statPrecision" valueType="real" columnNum="2"/>
    </ds:Definition>
```

- Using `columnType="indivParameter"`

– mean of the individual estimates

ID	CL	V	Ka	...	Tlag
1	0.0049	0.43	0.26		6.6
2	0.0035	0.27	0.56		5.8
3	0.0028	0.35	0.39		2
...	...	...	...		...
N	0.0045	0.48	0.17		8.1

```
<IndividualEstimates>
    <Estimates>
        <Mean>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="CL" columnType="indivParameter" valueType="real" columnNum="2"/>
                <Column columnId="V" columnType="indivParameter" valueType="real" columnNum="3"/>
                <Column columnId="KA" columnType="indivParameter" valueType="real" columnNum="4"/>
                ...
                <Column columnId="TLAG" columnType="indivParameter" valueType="real" columnNum="8"/>
                ...
            </Definition>
```

– precision of individual estimates

ID	Parameter	SD
1	CL	0.04
1	V	0.15
1	...	...
1	F	0.05
1	KA	0.08
...	...	...
N	CL	0.06
N	V	0.13
N	...	...
N	F	0.2
N	KA	0.63

```
<StandardDeviation>
    <ds:Definition>
        <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
        <ds:Column columnId="Parameter" columnType="indivParameter" valueType="string" columnNum="2"/>
        <ds:Column columnId="SD" columnType="statPrecision" valueType="real" columnNum="3"/>
    </ds:Definition>
```

3.3.4 Multiple `columnType` attributes

Multiple `columnType` attributes might be in certain cases useful or required. This is now possible as the following code snippet shows. Any number of values is allowed separated by white space.

```
<Bayesian>
    <PosteriorMean>
        <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Column columnId="POP_CL" columnType="popParameter structParameter" valueType="real" columnNum="1"/>
            <Column columnId="POP_V" columnType="popParameter" valueType="real" columnNum="2"/>
```

See also examples in section 3.3.

3.4 Higher variability levels

3.4.1 Inter-occasion (aka between-occasion) variability

Standard Output is capable of storing of data coming from experiments with virtually an arbitrary design structure. When considering the parameter related level of variability, see [Swat et al., 2015c] for detailed discussion, the IIV (BSV) variability level is in the most cases the only source to be considered. However, when multiple occasions are part of the design, they are often considered as additional sources of unexplained variability handled with random effects.

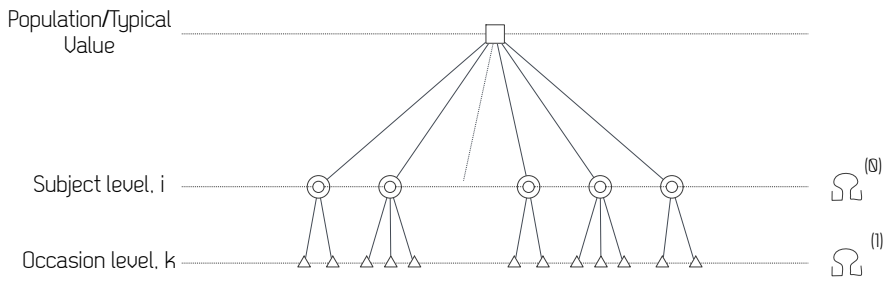

Figure 3.1: Variability structure with inter-individual and inter-occasion variability levels. In PharmML this is stored in the `<VariabilityModel>`.

Variability levels used to defined the hierarchy are indicated in the datasets headers, e.g.

- *OCC* – identifies occasions which are usually interpreted as an additional source of variability located typically below the subject, see Figure 3.1.

In SO, this is relevant information which needs to be indicated for the individual parameter estimates and their precision, see sections 4.4.3 and 4.4.4.

The first following example shows the encoding of individual estimates when no occasions are present

```
<IndividualEstimates>
    <Estimates>
        <Mean>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="CL" columnType="indivParameter" valueType="real" columnNum="2"/>
                <!-- omitted other parameters -->
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:String>i1</ct:String><ct:Real>0.28649</ct:Real>
                    <!-- omitted other parameters -->
                </Row>
                <!-- omitted other subjects -->
            </Table>
        </Mean>
```

In the second example, when occasions are considered, the additional column *OCC* is required to distinguish between these occasions. The indication of IOV is made via a combination of two attributes

- `columnType` – and assignment of *varLevel* value (new in 0.3.1)
- `level` – (new attribute in 0.3.1) which can be assigned any string.

```
<IndividualEstimates>
    <Estimates>
        <Mean>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="CL" columnType="indivParameter" valueType="real" columnNum="2"/>
                <!-- omitted other parameters -->
                <Column columnId="OCC" columnType="varLevel" level="occasion" valueType="int" columnNum="10"/>
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:String>i1</ct:String><ct:Real>0.28649</ct:Real>
                    <!-- omitted other parameters -->
                    <ct:Int>1</ct:Int>
                </Row>
                <!-- omitted other subjects -->
            </Table>
        </Mean>
```

If wanted and/or available, occasion start and end times can be captured as well with columns *OccasionStart* or *OccasionEnd*.

3.4.2 Study centre levels and beyond

Higher variability levels related to study, geographic location etc. are relatively frequent in pharmacometrics. Figure 3.2 illustrates a variability structure resulting from a design scenario with an additional, centre related, level, see [Swat et al., 2015c] for detailed discussion.

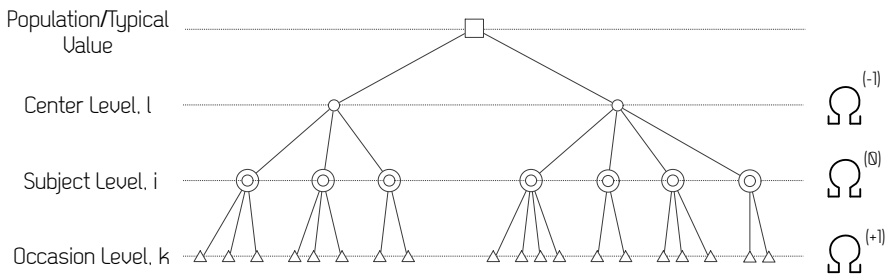

Figure 3.2: Higher levels of variability, centre and occasion related. In PharmML this is stored in the `<VariabilityModel>`.

In SO handling of additional variability levels is possible in multiple ways, as the examples below show:

```
<Column columnId="STUDY" columnType="varLevel" level="-2" valueType="real" columnNum="5"/>
<Column columnId="COUNTRY" columnType="varLevel" level="-1" valueType="real" columnNum="6"/>
<Column columnId="OCCASION" columnType="varLevel" level="+1" valueType="real" columnNum="7"/>
```

or

```
<Column columnId="STUDY" columnType="varLevel" level="study" valueType="real" columnNum="5"/>
<Column columnId="COUNTRY" columnType="varLevel" level="country" valueType="real" columnNum="6"/>
<Column columnId="OCCASION" columnType="varLevel" level="occasion" valueType="real" columnNum="7"/>
```

3.5 Covariates versus regressors

(This section is adapted from the PharmML 0.8 specification, [Swat et al., 2016b].) There are different types of covariates and we distinguish between following types

- constant covariates
- occasion dependent covariates

- time dependent covariates – *aka* regressors

Due to the various interpretation and naming conventions for covariates and regressors, in PharmML they can be identified either in the `<ModelDefinition>` or in the `<CovariateModel>`.

To account for this SO allows, in this version, the storage of both covariates and/or regressors is supported for both estimation and simulation tasks as explained in the sections 4.5, 4.6 and Table 4.13.

3.6 Samples in SO

When formulating models it is often not possible to specify a probability distribution using a parametric distribution because its defining functions are not known. Similarly, the exact type/name of the posterior distribution of population parameters is in most cases unknown. Instead, a given dataset or values range can be considered as determining a distribution model with or without additional parameters. There are various options how such distribution can be defined, see the ProbOnto specification for more details, [Swat et al., 2016a].

We show here how the posterior distribution of individual parameters estimates is captured using an empirical distribution *RandomSample*

```
<PosteriorDistribution>
   <Distribution>
      <po:ProbOnto name="RandomSample">
         <ds:DataSet>
            <ds:Definition>
               <ds:Column columnId="K" columnType="popParameter" valueType="real" columnNum="1"/>
               <ds:Column columnId="V" columnType="popParameter" valueType="real" columnNum="2"/>
               <ds:Column columnId="CL" columnType="popParameter" valueType="real" columnNum="3"/>
            </ds:Definition>
            <!-- option 1: inline table -->
            <ds:Table>
               <ds:Row><ct:Real>2.006510</ct:Real><ct:Real>5.050013</ct:Real><ct:Real>0.033333</ct:Real></ds:Row>
               <ds:Row><ct:Real>2.045465</ct:Real><ct:Real>5.050013</ct:Real><ct:Real>0.100000</ct:Real></ds:Row>
               <ds:Row><ct:Real>2.084421</ct:Real><ct:Real>5.050013</ct:Real><ct:Real>0.100000</ct:Real></ds:Row>
               <ds:Row><ct:Real>2.123377</ct:Real><ct:Real>5.050013</ct:Real><ct:Real>0.200000</ct:Real></ds:Row>
               <ds:Row><ct:Real>2.162333</ct:Real><ct:Real>5.064166</ct:Real><ct:Real>0.100000</ct:Real></ds:Row>
               <ds:Row><ct:Real>2.201288</ct:Real><ct:Real>5.064166</ct:Real><ct:Real>0.066667</ct:Real></ds:Row>
               <!-- other samples skipped -->
            </ds:Table>
            <!-- option 2: alternatively in an external dataset
               <ds:ExternalFile oid="extDataId">
                  <ds:path>samples_KVCL.csv</ds:path>
               </ds:ExternalFile>-->
         </ds:DataSet>
      </po:ProbOnto>
   </Distribution>
</PosteriorDistribution>
```

Chapter 4

SO structure

The SO structure consists of the following seven main sections:

1. Tool settings
2. Raw results
3. Task Information
4. Estimation
5. Model Diagnostic
6. Simulation
7. Optimal design

As shown in Figure 4.1, each section comprises specialised SO elements, with varying number of levels. The SO organisation as well as the format definition of each component, with XML examples, is described in this chapter. In few cases, for example for the purpose of bootstrap analysis, the reuse of certain sections listed above might be required. These are organised in blocks and the `<SOBlock>` ... `</SOBlock>` layer is always required. An unlimited number of blocks is permitted. Similar block structures are used for simulation, `<SimulationBlock>`, and optimal design results, `<OptimalDesignBlock>`, see sections 4.6 and 4.7, respectively.

SO structure, conceived as a tool-independent storage format, has been designed to allow flexibility in populating its elements. Among these features, it is worth to mention that:

- if any of the defined sections/elements are not available for a given target software, then, these sections/elements can be omitted;
- SO elements can be populated through the defined XML structure or by referring to an external file (any extension is allowed) coherently constructed.

For the sake of clarity, the column names used in the table templates are reported as examples for a better understanding of the related element. Any column name can be used and it is responsibility of the connector developers to make sure that, when applicable, these names are consistent with those originally defined by the user.

4.1 Tool Settings

The `<ToolSettings>` element keeps the location for storing the reference to any file containing the tool settings (e.g., task related, graphical) of the performed task. References to multiple files are allowed.

SO Element	Format
○ Tool-setting file(s)	`File Reference`

Table 4.1: Tool Settings

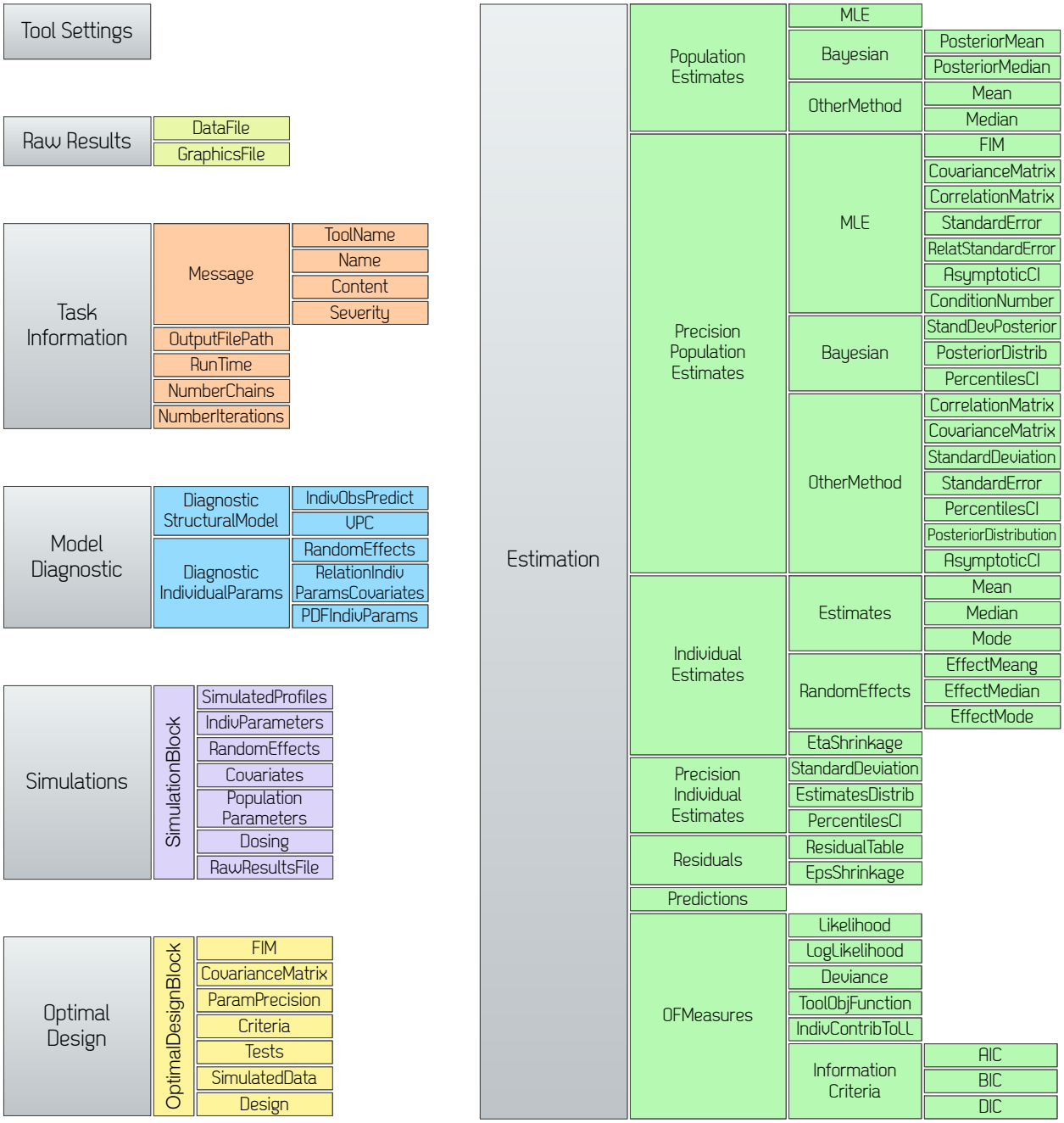

Figure 4.1: Overview of the hierarchical SO structure.

Example

```
<ToolSettings>
    <File oid="ts1">
        <ds:path>algorithms.xmlx</ds:path>
    </File>
    <File oid="ts2">
        <ds:path>solver.xmlx</ds:path>
    </File>
    <File oid="ts3">
        <ds:path>andNowForSomethingCompletelyDifferent.xmlx</ds:path>
    </File>
    <!-- any number of files is permited -->
</ToolSettings>
```

4.2 Raw Results

Within this section the reference to original output files, both data and graphical, produced by any target tool (e.g., WINBUGS, Monolix, NONMEM, PsN) are stored. No limits in the number of original files are imposed.

SO Element	Format
○ Original data file(s)	`File Reference`
○ Original plots	`File Reference`

Table 4.2: Raw Results.

Template

```
<RawResults>
    <DataFile>...</DataFile>
    <GraphicsFile oid="g1">...</GraphicsFile>
</RawResults>
```

Example

```
<RawResults>
    <DataFile>
        <ds:ExternalFile oid="d1">
            <ds:path>dataFolder/indiv_parameters.csv</ds:path>
        </ds:ExternalFile>
    </DataFile>

    <GraphicsFile oid="g1">
        <ds:path>graphicsFolder/ObservationVSprediction.png</ds:path>
    </GraphicsFile>

    <GraphicsFile oid="g2">
        <ds:path>graphicsFolder/figure2.png</ds:path>
    </GraphicsFile>

    <!-- any number of files is permited -->
</RawResults>
```

Note, that the definition of data columns and missing values is also possible (compared to v0.2) for the raw results as the following snippet demonstrates

```
<RawResults>
    <DataFile>
        <ds:Definition>
            <ds:Column columnId="MY_ID" columnType="id" valueType="int" columnNum="1"/>
            <ds:Column columnId="MY_TIME" columnType="idv" valueType="int" columnNum="2"/>
        </ds:Definition>
        <ds:ExternalFile oid="d2">
            <ds:path>dataFolder/warfarin_conc.pca</ds:path>
            <ds:MissingData dataCode="nan" missingDataType="NaN"/>
        </ds:ExternalFile>
    </DataFile>

        ...
```

4.3 Task Information

This section is composed of five elements holding information about the modelling step execution: tool message and diagnostic information, execution time, the path pointing to the original output file, number of chains (in case of MCMC execution) and iterations. Depending on how the target tool has been designed to inform the user about warning contents, errors and program termination, the `<Message>` element has been defined to embed any of these messages by specifying its `type` attribute. It is structured for providing the name of the tool generating it, the message name, a content description and the severity value.

SO Element	Format
○ Target Software Messages	Allowed type: `ERROR`, `WARNING`, `TERMINATION`, `INFORMATION`
– Tool name	`String`
– Message name	`String` (e.g. nmtran error)
– Message description	`String`
– Severity value	`Integer` from 1 (low) to 10 (high)
○ Run time	`Real`
○ Original output file path information	`File Reference`
○ Number of MCMC chains	`Real`
○ Number of iterations	`Real`

Table 4.3: Task Information.

Complete template/example

```
<TaskInformation>
    <!-- Messages from Target Software -->
    <!-- any number of messages is permitted -->
    <Message type="ERROR">
        <Toolname>NONMEM</Toolname>
        <Name>nmtran_error</Name>
        <Content>Possibly_long_text...</Content>
        <Severity>2</Severity>
    </Message>
    <OutputFilePath oid="o1">
        <ds:path>/results/mainResultFile.txt</ds:path>
    </OutputFilePath>
    <RunTime>23232.5</RunTime>
    <NumberChains>3</NumberChains>
    <NumberIterations>1000000</NumberIterations>
</TaskInformation>
```

4.4 Estimation

This section is designed to store typical output of interest resulting from an estimation task and it is organized into seven elements:

- `<PopulationEstimates>` – placeholder for population estimates,
- `<PrecisionPopulationEstimates>` – placeholder for precision of population estimates
- `<IndividualEstimates>` – placeholder for individual estimates
- `<PrecisionIndividualEstimates>` – placeholder for precision of individual estimates
- `<Residuals>` – placeholder for multiple types of residuals
- `<Predictions>` – placeholder for individual and population predictions
- `<Likelihood>` – placeholder for estimated log-likelihood

located within the `<Estimation>` tag. These elements are in turn composed of child elements allowing if required the separation among different estimation techniques (e.g., MLE, Bayesian, Bootstrap).

4.4.1 Population estimates

The `<PopulationEstimates>` element stores the population parameter estimates obtained from MLE, Bayesian and other techniques (e.g. sampling methods). The child elements within Bayesian and Other methods allow to account for two measures of central tendency describing the posterior distributions of the population parameters.

SO Element	Format
◦ Maximum likelihood estimate	`Table` with columns: `POP_PARAM_1`, . . . , `POP_PARAM_K`
◦ Bayesian	
– Posterior Mean	`Table` with columns: `POP_PARAM_1`, . . . , `POP_PARAM_K`
– Posterior Median	`Table` with columns: `POP_PARAM_1`, . . . , `POP_PARAM_K`
◦ Other methods – with attribute *method*♠,♡	
– Mean	`Table` with columns: `POP_PARAM_1`, . . . , `POP_PARAM_K`
– Median	`Table` with columns: `POP_PARAM_1`, . . . , `POP_PARAM_K`

Table 4.4: Population estimates. ♠The build-in methods are: *Bootstrap*, *LLP*, *SIR* or *MultiDimLLP* but any string without white-spaces is allowed which means that adding new methods is possible without changing the PharmML schema. ♡Multiple `<OtherMethod>` elements are allowed.

Complete template

```
<Estimation>
    <PopulationEstimates>
        <MLE>...</MLE>
        <Bayesian>
            <PosteriorMean>...</PosteriorMean>
            <PosteriorMedian>...</PosteriorMedian>
        </Bayesian>
        <OtherMethod method="...">
            <Mean>...</Mean>
            <Median>...</Median>
        </OtherMethod>
    </PopulationEstimates>
```

Selected examples

The first example is the table with MLE population estimates

```
<Estimation>
    <PopulationEstimates>
        <MLE>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="POP_CL" columnType="popParameter" valueType="real" columnNum="1"/>
                <Column columnId="POP_V" columnType="popParameter" valueType="real" columnNum="2"/>
                <Column columnId="POP_KA" columnType="popParameter" valueType="real" columnNum="3"/>
                <Column columnId="POP_TLAG" columnType="popParameter" valueType="real" columnNum="4"/>
                <Column columnId="BETA_CL_WT" columnType="popParameter" valueType="real" columnNum="5"/>
                <Column columnId="BETA_V_WT" columnType="popParameter" valueType="real" columnNum="6"/>
                <Column columnId="PPV_CL" columnType="varParameter variance" valueType="real" columnNum="7"/>
                <Column columnId="PPV_V" columnType="varParameter variance" valueType="real" columnNum="8"/>
                <Column columnId="PPV_KA" columnType="varParameter variance" valueType="real" columnNum="9"/>
                <Column columnId="PPV_TLAG" columnType="varParameter variance" valueType="real" columnNum="10"/>
                <Column columnId="RUV_PROP" columnType="popParameter" valueType="real" columnNum="11"/>
                <Column columnId="RUV_ADD" columnType="popParameter" valueType="real" columnNum="12"/>
                <Column columnId="CORR_PPV_CL_V" columnType="varParameter correlation" valueType="real" columnNum="13"/>
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row><ct:Real>0.93882</ct:Real><ct:Real>1.63219</ct:Real><ct:Real>8.10703</ct:Real>
                    <ct:Real>1</ct:Real><ct:Real>0.13426</ct:Real><ct:Real>0.75</ct:Real>
                    <ct:Real>0.54806</ct:Real><ct:Real>0.76833</ct:Real><ct:Real>0.13334</ct:Real>
                    <ct:Real>0.26875</ct:Real><ct:Real>0.16612</ct:Real><ct:Real>0.32349</ct:Real>
                    <ct:Real>0.07736</ct:Real>
                </Row>
            </Table>
                <!--ALTERNATIVE: external data file reference. -->
                <!-- <ds:ExternalFile oid="ext1">
                <ds:path>myData.csv</ds:path>
            </ds:ExternalFile>-->
        </MLE>
```

The second example shows the mean of population estimates obtained using *Bootstrap*

```
<Estimation>
```

```
<PopulationEstimates>
    <!-- ... -->
    <OtherMethod method="Bootstrap">
        <Mean>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="POP_CL" columnType="popParameter" valueType="real" columnNum="1"/>
                <Column columnId="POP_V" columnType="popParameter" valueType="real" columnNum="2"/>
                <Column columnId="POP_KA" columnType="popParameter" valueType="real" columnNum="3"/>
                <Column columnId="POP_TLAG" columnType="popParameter" valueType="real" columnNum="4"/>
                <Column columnId="BETA_CL_WT" columnType="popParameter" valueType="real" columnNum="5"/>
                <Column columnId="BETA_V_WT" columnType="popParameter" valueType="real" columnNum="6"/>
                <Column columnId="PPV_CL" columnType="varParameter variance" valueType="real" columnNum="7"/>
                <Column columnId="PPV_V" columnType="varParameter variance" valueType="real" columnNum="8"/>
                <Column columnId="PPV_KA" columnType="varParameter variance" valueType="real" columnNum="9"/>
                <Column columnId="PPV_TLAG" columnType="varParameter variance" valueType="real" columnNum="10"/>
                <Column columnId="RUV_PROP" columnType="popParameter" valueType="real" columnNum="11"/>
                <Column columnId="RUV_ADD" columnType="popParameter" valueType="real" columnNum="12"/>
                <Column columnId="CORR_PPV_CL_V" columnType="varParameter correlation" valueType="real" columnNum="13"/>
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row><ct:Real>0.93882</ct:Real><ct:Real>1.63219</ct:Real><ct:Real>8.10703</ct:Real>
                    <ct:Real>1</ct:Real><ct:Real>0.13426</ct:Real><ct:Real>0.75</ct:Real>
                    <ct:Real>0.54806</ct:Real><ct:Real>0.76833</ct:Real><ct:Real>0.13334</ct:Real>
                    <ct:Real>0.26875</ct:Real><ct:Real>0.16612</ct:Real><ct:Real>0.32349</ct:Real>
                    <ct:Real>0.07736</ct:Real>
                </Row>
            </Table>
        </Mean>
```

4.4.2 Precision of population estimates

Uncertainty measures of population parameter estimates are stored within this SO element.

The `<PrecisionPopulationEstimates>` is structured for accounting of typical information and statistics calculated from MLE and Bayesian estimation as well as techniques (e.g., Bootstrap, LLP, SIR, MultiDim-LLP) to assess the precision of population estimates. For this purpose, uncertainty measures for each considered method have been considered and defined as child elements (e.g., `<FIM>`, `<CovarianceMatrix>`, `<CorrelationMatrix>`, `<StandardError>`, `<RelativeStandardError>`, `<AsymptoticCI>` and `<ConditionNumber>` within the `<MLE>` element).

SO Element	Format
◦ MLE	
– Fisher Information Matrix (FIM)	`Matrix` with `RowNames`, `ColumnNames`
– Covariance matrix	`Matrix` with `RowNames`, `ColumnNames`
– Correlation matrix	`Matrix` with `RowNames`, `ColumnNames`
– Standard Error	`Table` with columns: `Parameter`, `SE`
– Relative standard error (%)	`Table` with columns: `Parameter`, `RSE`
– Asymptotic CI	`Table` with columns: `Parameter`, `CI`, `LowerBound`, `UpperBound`
– Condition number	`Real`
◦ Bayesian	
– Standard deviation of posterior distribution	`Table` with columns: `Parameter`, `SDP`
– Distributions of the population parameters	`Sample` or `Table` with columns: `POP_PARAM1`, ..., `POP_PARAM_K`
– Percentiles CI	`Table` with columns: `Percentile`, `POP_PARAM1`, ..., `POP_PARAM_K`
◦ Other methods – with attribute *method*$^{\spadesuit,\heartsuit}$	
– Covariance matrix$^{\diamondsuit}$	`Matrix` with `RowNames`, `ColumnNames`
– Correlation matrix	`Matrix` with `RowNames`, `ColumnNames`
– Standard deviation	`Table` with columns: `Parameter`, `SD`
– Standard error	`Table` with columns: `Parameter`, `SE`

– Asymptotic CI	`Table` with columns: `Parameter`, `CI`, `LowerBound`, `UpperBound`
– Distributions of the population parameters	`Sample` or `Table` with columns: `Parameter`, `PD`
– Percentiles CI	`Table` with columns: `Percentile`, `POP_PARAM1`, ..., `POP_PARAM_K`

Table 4.5: Precision of population estimates. ♠,♡ See previous table. ♢ The matrix element should be populated only when the SIR method is applied.

Complete template

```
<PrecisionPopulationEstimates>
    <MLE>
        <FIM>...</FIM>
        <CovarianceMatrix>...</CovarianceMatrix>
        <CorrelationMatrix>...</CorrelationMatrix>
        <StandardError>...</StandardError>
        <RelativeStandardError>...</RelativeStandardError>
        <AsymptoticCI>...</AsymptoticCI>
        <ConditionNumber>...</ConditionNumber>
    </MLE>
    <Bayesian>
        <StandardDeviation>...</StandardDeviation>
        <PosteriorDistribution>...</PosteriorDistribution>
        <PercentilesCI>...</PercentilesCI>
    </Bayesian>
    <OtherMethod method="...">
        <CovarianceMatrix>...</CovarianceMatrix>
        <CorrelationMatrix>...</CorrelationMatrix>
        <StandardDeviation>...</StandardDeviation>
        <StandardError>...</StandardError>
        <AsymptoticCI>...</AsymptoticCI>
        <PosteriorDistribution>...</PosteriorDistribution>
        <PercentilesCI>...</PercentilesCI>
    </OtherMethod>
</PrecisionPopulationEstimates>
```

Selected examples

The first is an *MLE* example showing how the FIM matrix is implemented

```
<PrecisionPopulationEstimates>
    <MLE>
        <FIM>
            <ct:Matrix matrixType="Any">
                <ct:RowNames>
                    <ct:String>POP_CL</ct:String><ct:String>POP_V</ct:String><ct:String>POP_KA</ct:String>
                    <ct:String>POP_TLAG</ct:String><ct:String>BETA_CL_WT</ct:String><ct:String>BETA_V_WT</ct:String>
                    <ct:String>PPV_CL</ct:String><ct:String>PPV_V</ct:String><ct:String>PPV_KA</ct:String>
                    <ct:String>PPV_TLAG</ct:String><ct:String>RUV_PROP</ct:String><ct:String>RUV_ADD</ct:String>
                    <ct:String>CORR_PPV_CL_V</ct:String>
                </ct:RowNames>
                <ct:ColumnNames>
                    <!-- omitted names as identical to those above -->
                </ct:ColumnNames>
                <ct:MatrixRow>
                    <ct:Real>4.21e-05</ct:Real><ct:Real>2.36e-04</ct:Real><ct:Real>-0.000124</ct:Real>
                    <ct:Real>-1.01e-04</ct:Real><ct:Real>1.15e-04</ct:Real><ct:Real>-1.74e-06</ct:Real>
                    <ct:Real>0</ct:Real><ct:Real>0</ct:Real><ct:Real>0</ct:Real><ct:Real>0</ct:Real>
                    <ct:Real>0</ct:Real><ct:Real>0</ct:Real>
                </ct:MatrixRow>
                <ct:MatrixRow>
                    <ct:Real>2.36e-04</ct:Real><ct:Real>4.95e-02</ct:Real><ct:Real>-0.006150</ct:Real>
                    <ct:Real>-1.24e-03</ct:Real><ct:Real>-1.42e-04</ct:Real><ct:Real>1.12e-04</ct:Real>
                    <ct:Real>0</ct:Real><ct:Real>0</ct:Real><ct:Real>0</ct:Real><ct:Real>0</ct:Real>
                    <ct:Real>0</ct:Real><ct:Real>0</ct:Real>
                </ct:MatrixRow>
                <!-- omitted other matrix rows -->
            </ct:Matrix>
        </FIM>
```

The second is a *Bayesian* example showing how standard deviation of posterior distribution is stored

```
<PrecisionPopulationEstimates>
    <!-- ... -->
    <Bayesian>
        <StandardDeviation>
            <ds:Definition>
                <ds:Column columnId="Parameter" columnType="popParameter" valueType="string" columnNum="1"/>
                <ds:Column columnId="SDP" columnType="statPrecision" valueType="real" columnNum="2"/>
            </ds:Definition>
            <ds:Table>
                <ds:Row><ct:String>POP_CL</ct:String><ct:Real>0.1</ct:Real></ds:Row>
                <ds:Row><ct:String>POP_V</ct:String><ct:Real>0.2</ct:Real></ds:Row>
                <!-- omitted other parameters -->
            </ds:Table>
        </StandardDeviation>
```

The third is an *OtherMethod* example showing how confidence intervals are stored

```
<PrecisionPopulationEstimates>
    <!-- ... -->
    <OtherMethod method="LLP">
        <AsymptoticCI>
            <ds:Definition>
                <ds:Column columnId="Parameter" columnType="popParameter" valueType="string" columnNum="1"/>
                <ds:Column columnId="CI" columnType="statPrecision" valueType="real" columnNum="2"/>
                <ds:Column columnId="LowerBound" columnType="statPrecision" valueType="real" columnNum="3"/>
                <ds:Column columnId="UpperBound" columnType="statPrecision" valueType="real" columnNum="4"/>
            </ds:Definition>
            <ds:Table>
                <ds:Row>
                    <ct:String>POP_CL</ct:String><ct:String>95\%</ct:String><ct:Real>0.1</ct:Real><ct:Real>0.5</ct:Real>
                </ds:Row>
                <ds:Row>
                    <ct:String>POP_V</ct:String><ct:String>95\%</ct:String><ct:Real>0.2</ct:Real><ct:Real>0.6</ct:Real>
                </ds:Row>
                <!-- omitted other parameters -->
            </ds:Table>
        </AsymptoticCI>
```

4.4.3 Individual estimates

The `<IndividualEstimates>` element consists of three child elements. The first one allows storing measures of central tendency describing the estimated conditional/posterior distribution of individual parameters. Similarly, the inter-individual/inter-occasion (IIV/IOV) and other variance components are placed in the `<RandomEffects>` child-element. ETA-shrinkage is stored within another child element.

The column `OCC` and related columns (i.e., `OccasionStart` and `OccasionEnd`) are optional and needed only when inter-occasion variability or other levels of variability are considered, see section 3.4 for a detailed description. Below we provide the format definition for two cases, the first one when only IIV is considered, table 4.6, the second when IOV has to be taken into account, table 4.7.

SO Element	Format
◦ Individual parameters estimates	
– Mean conditional/posterior distribution	`Table` with columns: `ID`, `PARAM_1`, ..., `PARAM_N`
– Median conditional/posterior distribution	`Table` with columns: `ID`, `PARAM_1`, ..., `PARAM_N`
– Mode conditional/posterior distribution	`Table` with columns: `ID`, `PARAM_1`, ..., `PARAM_N`
◦ Random effects estimates (ETAs)	
– Mean conditional/posterior distribution	`Table` with columns: `ID`, `ETA_1`, ..., `ETA_J`
– Median conditional/posterior distribution	`Table` with columns: `ID`, `ETA_1`, ..., `ETA_J`
– Mode conditional/posterior distribution	`Table` with columns: `ID`, `ETA_1`, ..., `ETA_J`
◦ ETA-shrinkage	`Table` with columns: `ETA_SHRINKAGE_1`, ..., `ETA_SHRINKAGE_J`

Table 4.6: Individual estimates. Table format is specified here for examples characterised by only IIV; for an example with IOV level see Table 4.7.

SO Element	Format
◦ Individual parameters estimates	
– Mean conditional/posterior distribution	`Table` with columns: `ID`, `OccasionStart`, `OccasionEnd`, `PARAM_1`, ..., `PARAM_N`, `OCC`

Table 4.7: Table extension to cover additional levels of variability. In particular, additional column are required for individual estimates with IOV. The columns `OCC`, `OccasionStart`, `OccasionEnd` etc., see section 3.4.2, are optional but needed only when IOV is part of the model.

Complete template

```
<IndividualEstimates>
    <Estimates>
        <Mean>...</Mean>
        <Median>...</Median>
        <Mode>...</Mode>
    </Estimates>
    <RandomEffects>
        <EffectMean>...</EffectMean>
        <EffectMedian>...</EffectMedian>
        <EffectMode>...</EffectMode>
    </RandomEffects>
    <EtaShrinkage>...</EtaShrinkage>
</IndividualEstimates>
```

Selected examples

The first example shows how the mean of the individual estimates is stored

```
<IndividualEstimates>
    <Estimates>
        <Mean>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="CL" columnType="indivParameter" valueType="real" columnNum="2"/>
                <Column columnId="V" columnType="indivParameter" valueType="real" columnNum="3"/>
                <Column columnId="KA" columnType="indivParameter" valueType="real" columnNum="4"/>
                <Column columnId="TLAG" columnType="indivParameter" valueType="real" columnNum="5"/>
                <!-- omitted other parameters -->
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:Id>i1</ct:Id><ct:Real>0.28649</ct:Real><ct:Real>7.892</ct:Real>
                    <ct:Real>0.3025</ct:Real><ct:Real>0.5588</ct:Real>
                </Row>
                <Row>
                    <ct:Id>i2</ct:Id><ct:Real>0.11184</ct:Real><ct:Real>7.670</ct:Real>
                    <ct:Real>1.3134</ct:Real><ct:Real>0.8290</ct:Real>
                </Row>
                <!-- omitted other subjects -->
            </Table>
        </Mean>
```

The second example shows how the mean of random effects are stored

```
<IndividualEstimates>
    <!-- ... -->
    <RandomEffects>
        <EffectMean>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="id" columnNum="1"/>
                <Column columnId="ETA_CL" columnType="randEffect" valueType="real" columnNum="2"/>
                <Column columnId="ETA_V" columnType="randEffect" valueType="real" columnNum="3"/>
                <Column columnId="ETA_KA" columnType="randEffect" valueType="real" columnNum="4"/>
                <Column columnId="ETA_TLAG" columnType="randEffect" valueType="real" columnNum="5"/>
                <!-- omitted other random effects -->
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
```

```
                <ct:Id>i1</ct:Id><ct:Real>0.79602</ct:Real><ct:Real>0.02201</ct:Real>
                <ct:Real>-1.4809</ct:Real><ct:Real>-3.913e-01</ct:Real>
            </Row>
            <Row>
                <ct:Id>i2</ct:Id><ct:Real>-0.1446</ct:Real><ct:Real>-0.00652</ct:Real>
                <ct:Real>-0.012555</ct:Real><ct:Real>2.961e-03</ct:Real>
            </Row>
            <!-- omitted other subjects -->
        </Table>
    </EffectMean>
```

The third example shows how the η-shrinkage is stored

```
<IndividualEstimates>
    <!-- ... -->
    <EtaShrinkage>
        <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Column columnId="ETA_SHRINKAGE_CL" columnType="statPrecision" valueType="real" columnNum="1"/>
            <Column columnId="ETA_SHRINKAGE_V" columnType="statPrecision" valueType="real" columnNum="2"/>
            <Column columnId="ETA_SHRINKAGE_KA" columnType="statPrecision" valueType="real" columnNum="3"/>
            <Column columnId="ETA_SHRINKAGE_TLAG" columnType="statPrecision" valueType="real" columnNum="4"/>
            <!-- omitted other random effects shrinkage -->
        </Definition>
        <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Row>
                <ct:Real>-0.04671</ct:Real><ct:Real>11.073</ct:Real><ct:Real>51.076</ct:Real><ct:Real>59.033</ct:Real>
            </Row>
        </Table>
    </EtaShrinkage>
```

4.4.4 Precision of individual estimates

Any summary statistics and samples of the conditional/posterior distribution of individual parameters can be stored here.

SO Element	Format
– Standard deviation	`Table` with columns: `ID`, `Parameter`, `SD`
– Distribution of individual parameters	`Sample` or `Table` with columns: `ID`, `PARAM_1`, ..., `PARAM_N`
– Percentiles CI	`Table` with columns: `ID`, `Percentile`, `PARAM_1`, ..., `PARAM_K`

Table 4.8: Precision of individual estimates. The occasions level related columns, `OCC`, `OccasionStart` and `OccasionEnd`, are optional and should be used only if such variability level is present in the study. See section 4.4.3 for more details.

Complete template

```
    <PrecisionIndividualEstimates>
        <StandardDeviation>...</StandardDeviation>
        <EstimatesDistribution>...</EstimatesDistribution>
        <PercentilesCI>...</PercentilesCI>
    </PrecisionIndividualEstimates>
```

Selected examples

The first example shows how to store the StandardDeviation

```
<PrecisionIndividualEstimates>
    <StandardDeviation>
        <ds:Definition>
            <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
            <ds:Column columnId="Parameter" columnType="indivParameter" valueType="string" columnNum="2"/>
            <ds:Column columnId="SD" columnType="statPrecision" valueType="real" columnNum="3"/>
        </ds:Definition>
        <ds:Table>
            <ds:Row>
                <ct:String>i1</ct:String><ct:String>CL</ct:String><ct:Real>0.1</ct:Real>
            </ds:Row>
```

```
            <ds:Row>
                <ct:String>i1</ct:String><ct:String>V</ct:String><ct:Real>0.2</ct:Real>
            </ds:Row>
            <!-- omitted other subjects and parameters -->
        </ds:Table>
    </StandardDeviation>
```

The second example shows how the estimates distribution is captured using an empirical distribution *RandomSample* from ProbOnto.

```
<PrecisionIndividualEstimates>
    <EstimatesDistribution>
        <Distribution>
            <po:ProbOnto name="RandomSample">
                <ds:DataSet>
                    <ds:Definition>
                        <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                        <ds:Column columnId="K" columnType="indivParameter" valueType="real" columnNum="2"/>
                        <ds:Column columnId="V" columnType="indivParameter" valueType="real" columnNum="3"/>
                        <ds:Column columnId="CL" columnType="indivParameter" valueType="real" columnNum="4"/>
                    </ds:Definition>
                    <ds:Table>
                        <ds:Row>
                            <ct:String>i1</ct:String><ct:Real>2.06</ct:Real><ct:Real>5.05</ct:Real><ct:Real>0.03</ct:Real>
                        </ds:Row>
                        <ds:Row>
                            <ct:String>i2</ct:String><ct:Real>2.04</ct:Real><ct:Real>5.05</ct:Real><ct:Real>0.10</ct:Real>
                        </ds:Row>
                        <!-- other subjects skipped -->
                    </ds:Table>
                </ds:DataSet>
            </po:ProbOnto>
        </Distribution>
    </EstimatesDistribution>
```

4.4.5 Residuals

Calculated residuals are here stored in a unique table format. The type of residuals is identified by the column name (e.g., RES, IWRES, PWRES) and the type of observation is captured in the DVID column. Within this element the EPS-shrinkage is also stored.

SO Element	Format
○ Table of residuals (includes NPDE, PD, population residuals, population weighted residuals, individual weighted residuals, individual residuals, conditional weighted residuals)♠	`Table` with columns: `ID`, `DVID`, `TIME`, `RESIDUAL_NAME_1`, ..., `RESIDUAL_NAME_W`
○ EPS-Shrinkage	`Table` with columns: `EPS_SHRINKAGE_1`, ..., `EPS_SHRINKAGE_R`

Table 4.9: Residuals. ♠Note that dosing records should not be included in this table.

Example

```
<Residuals>
    <ResidualTable>
        <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
            <Column columnId="DVID" columnType="dvid" valueType="string" columnNum="2"/>
            <Column columnId="TIME" columnType="idv" valueType="real" columnNum="3"/>
            <Column columnId="RES" columnType="residual" valueType="real" columnNum="4"/>
            <Column columnId="IRES" columnType="residual" valueType="real" columnNum="5"/>
            <Column columnId="WRES" columnType="residual" valueType="real" columnNum="6"/>
            <Column columnId="CWRES" columnType="residual" valueType="real" columnNum="7"/>
            <Column columnId="IWRES" columnType="residual" valueType="real" columnNum="8"/>
            <Column columnId="PD" columnType="residual" valueType="real" columnNum="9"/>
            <Column columnId="NPDE" columnType="residual" valueType="real" columnNum="10"/>
```

```
        </Definition>
        <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Row>
                <ct:String>i1</ct:String><ct:Real>1</ct:Real><ct:Real>2</ct:Real><ct:Real>3</ct:Real>
                <ct:Real>4</ct:Real><ct:Real>5</ct:Real><ct:Real>6</ct:Real><ct:Real>7</ct:Real>
                <ct:Real>8</ct:Real><ct:Real>9</ct:Real><ct:Real>10</ct:Real>
            </Row>
            <Row>
                <ct:String>i2</ct:String><ct:Real>1</ct:Real><ct:Real>2</ct:Real><ct:Real>3</ct:Real>
                <ct:Real>4</ct:Real><ct:Real>5</ct:Real><ct:Real>6</ct:Real><ct:Real>7</ct:Real>
                <ct:Real>8</ct:Real><ct:Real>9</ct:Real><ct:Real>10</ct:Real>
            </Row>
            <!-- omitted other subjects -->
        </Table>
    </ResidualTable>
```

4.4.6 Predictions

The `<Predictions>` element contains the population and individual predictions over time. This element can be populated with any function of time (e.g., hazard, Poisson) as not being restricted to prediction for continuous data.

SO Element	Format
○ Prediction table: includes individual predictions, population predictions♠	`Table` with columns: `ID`, `DVID`, `TIME`, `PRED`, `IPRED`

Table 4.10: Predictions. ♠Note that dosing records should not be included in this table.

Example

```
<Predictions>
    <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
        <Column columnId="ID" columnType="id" valueType="id" columnNum="1"/>
        <Column columnId="TIME" columnType="idv" valueType="real" columnNum="2"/>
        <Column columnId="PRED" columnType="undefined" valueType="real" columnNum="3"/>
        <Column columnId="IPRED" columnType="undefined" valueType="real" columnNum="4"/>
        <!-- omitted other residuals -->
    </Definition>
    <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
        <Row>
            <ct:Id>i1</ct:Id><ct:Real>0</ct:Real><ct:Real>7.892</ct:Real><ct:Real>1.892</ct:Real>
        </Row>
        <Row>
            <ct:Id>i2</ct:Id><ct:Real>1</ct:Real><ct:Real>7.670</ct:Real><ct:Real>6.802</ct:Real>
        </Row>
        <!-- omitted other subjects -->
    </Table>
</Predictions>
```

4.4.7 OFMeasures

The child-elements of `<OFMeasures>` contain the values of various estimated objective function measures, such as the likelihood, log-likelihood, tool-specific objective function, deviance and individual contributions to likelihood as well as the calculated values of the most common information criteria used for model selection purpose (i.e., AIC, BIC, DIC).

In addition to the the tool-specific objective function, it is highly recommended to store values of derived measures in the provided elements.

SO Element	Format
○ Likelihood	`Real`
○ Log-likelihood	`Real`
○ Deviance (e.g. -2LL)	`Real`
○ Tool-specific objective function♠	`Real`

◦ Individual contributions to the log-likelihood	`Table` with columns `ID`, `ICtoLL`
◦ Information criterion	
– AIC	`Real`
– BIC	`Real`
– DIC	`Real`

Table 4.11: Likelihood. ♠The tool-specific objective function, implemented with `<ToolObjfunction>` element, can be annotated using the optional `type` attribute which accepts any string, see example below.

Example

```
<OFMeasures>
    <Likelihood>1.234</Likelihood>
    <LogLikelihood>-2.234</LogLikelihood>
    <Deviance>3.345</Deviance>
    <ToolObjFunction type="MySpecialOFtype">4.345E-01</ToolObjFunction>
    <IndividualContribToLL>
        <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Column columnId="ID" columnType="id" valueType="id" columnNum="1"/>
            <Column columnId="ICtoLL" columnType="undefined" valueType="real" columnNum="2"/>
        </Definition>
        <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Row>
                <ct:Id>i1</ct:Id><ct:Real>7.892</ct:Real>
            </Row>
            <Row>
                <ct:Id>i2</ct:Id><ct:Real>7.670</ct:Real>
            </Row>
            <!-- omitted other subjects -->
        </Table>
    </IndividualContribToLL>
    <InformationCriteria>
        <AIC>3.456</AIC>
        <BIC>4.567</BIC>
        <DIC>5.678</DIC>
    </InformationCriteria>
</OFMeasures>
```

4.5 Model Diagnostic

This SO section is designed for storing information resulting from typical model diagnostic plots; then, this section would allow the user to reproduce the graphics. In particular, the two defined SO elements distinguish between diagnostic plots for assessing the structural model and residual error specification, and diagnostic plots for individual parameters.

SO Element	Format
◦ Diagnostic for structural model and residual errors	
– Individual observations and predictions♠	`Table` with columns `ID`, `TIME`, `DVID`, `OBSERVATION`, `PREDICTIONcont`, `Residual_Name_1`, ..., `Residual_Name_R`, `STRATA_Covariate_1`, ..., `STRATA_Covariate_S`
– VPC	`Table` with columns `TIME`, `BinLow`, `BinHigh`, `Percentile_1`, `OBSERVED_PERCENTILE_1`, `LowerBound`, `UpperBound`, ..., `Percentile_X`, `OBSERVED_PERCENTILE_X`, `LowerBound`, `UpperBound`, `STRATA_Covariate_1`,..., `STRATA_Covariate_S`
◦ Diagnostic for individual parameters♢	
– Random effects	`Table` with columns `ID`, `ETA_1`, ..., `ETA_J`
– Individual parameters and covariates♡	`Table` with columns `ID`, `TIME`, `PARAM_1`, ..., `PARAM_N`, `Covariate_1`, ..., `Covariate_D`, `STRATA_Covariate_1`, ..., `STRATA_Covariate_S`
– Distribution of individual parameters (theoretical and empirical)	`Sample` or `Table` with columns: `ID`, `TIME`, `PARAM_1`, ..., `PARAM_N`, `Covariate_1`, ..., `Covariate_D`, `STRATA_Covariate_1`, ..., `STRATA_Covariate_S`

Table 4.12: Model Diagnostic. For each individual predictions values are stored at the available time instants of correspondent observation only.
♠The PREDICTIONcont element stores the predicted time course of the modelled variables. Note that dosing records should not be included in this table.
♢The occasions level related columns, `OCC`, `OccasionStart` and `OccasionEnd`, are optional and should be used only if such variability level is present in the study. See section 4.4.3 for more details.
♡`TIME` column is required only for time-varying covariates (regressors); for occasion dependent covariates the occasion level related columns can be used.

Complete template

```
<ModelDiagnostic>
    <DiagnosticStructuralModel>
        <IndivObservationPrediction>...</IndivObservationPrediction>
        <VPC>...</VPC>
    </DiagnosticStructuralModel>
    <DiagnosticIndividualParams>
        <RandomEffects>...</RandomEffects>
        <RelationIndivParamsCovariates>...</RelationIndivParamsCovariates>
        <PDFIndivParams>...</PDFIndivParams>
    </DiagnosticIndividualParams>
</ModelDiagnostic>
```

Example

The following snippet shows the entire diagnostic section as it is one of the more complex parts of SO.

```
<ModelDiagnostic>
    <DiagnosticStructuralModel>
        <IndivObservationPrediction>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="TIME" columnType="time" valueType="real" columnNum="2"/>
                <Column columnId="DVID" columnType="dvid" valueType="real" columnNum="3"/>
                <Column columnId="OBSERVATION" columnType="dv" valueType="real" columnNum="4"/>
                <Column columnId="PREDICTIONcont" columnType="undefined" valueType="real" columnNum="5"/>
                <Column columnId="WRES" columnType="residual" valueType="real" columnNum="6"/>
                <!-- omitted other residuals -->
                <Column columnId="STRATA_DOSE" columnType="strataVariable" valueType="real" columnNum="7"/>
                <!-- omitted STRATA -->
                <Column columnId="STRATA_STUDY" columnType="strataVariable" valueType="real" columnNum="8"/>
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:String>i1</ct:String><ct:Real>1</ct:Real>...<ct:Real>5</ct:Real>
                </Row>
                <!-- omitted other subjects -->
            </Table>
        </IndivObservationPrediction>
        <VPC>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="TIME" columnType="time" valueType="real" columnNum="1"/>
                <Column columnId="BinLow" columnType="statPrecision" valueType="real" columnNum="2"/>
                <Column columnId="BinHigh" columnType="statPrecision" valueType="real" columnNum="3"/>
                <Column columnId="Percentile_1" columnType="statPrecision" valueType="real" columnNum="4"/>
                <Column columnId="ObservedPercentile_1" columnType="statPrecision" valueType="real" columnNum="5"/>
                <Column columnId="LowerBound" columnType="statPrecision" valueType="real" columnNum="6"/>
                <Column columnId="UpperBound" columnType="statPrecision" valueType="real" columnNum="7"/>
                <!-- omitted columns -->
                <Column columnId="Percentile_X" columnType="statPrecision" valueType="real" columnNum="10"/>
                <Column columnId="ObservedPercentile_X" columnType="statPrecision" valueType="real" columnNum="11"/>
                <Column columnId="LowerBound" columnType="statPrecision" valueType="real" columnNum="12"/>
                <Column columnId="UpperBound" columnType="statPrecision" valueType="real" columnNum="13"/>
                <Column columnId="STRATA_DOSE" columnType="strataVariable" valueType="real" columnNum="14"/>
                <!-- omitted STRATA -->
                <Column columnId="STRATA_STUDY" columnType="strataVariable" valueType="real" columnNum="15"/>
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
```

```
                    <ct:Real>1</ct:Real><ct:Real>2</ct:Real>...<ct:Real>15</ct:Real>
                </Row>
                <!-- omitted rows -->
            </Table>
        </VPC>
    </DiagnosticStructuralModel>

    <DiagnosticIndividualParams>
        <RandomEffects>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="ETA_V" columnType="randEffect" valueType="real" columnNum="2"/>
                <Column columnId="ETA_CL" columnType="randEffect" valueType="real" columnNum="3"/>
                <!-- omitted other random effects -->
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:String>i1</ct:String><ct:Real>.1</ct:Real><ct:Real>.2</ct:Real><!-- ... -->
                </Row>
                <!-- omitted other subjects -->
            </Table>
        </RandomEffects>
        <IndivParamsCovariates>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="V" columnType="popParameter" valueType="real" columnNum="2"/>
                <Column columnId="CL" columnType="popParameter" valueType="real" columnNum="3"/>
                <!-- other parameters skipped -->
                <Column columnId="WT" columnType="covariate" valueType="real" columnNum="7"/>
                <Column columnId="SEX" columnType="covariate" valueType="real" columnNum="8"/>
                <Column columnId="AGE" columnType="covariate" valueType="real" columnNum="9"/>
                <!-- other covariates skipped -->
                <Column columnId="STRATA_DOSE" columnType="strataVariable" valueType="real" columnNum="12"/>
                <Column columnId="STRATA_STUDY" columnType="strataVariable" valueType="real" columnNum="13"/>
                <!-- other strata skipped -->
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:String>i1</ct:String><ct:Real>1</ct:Real>...<ct:Real>12</ct:Real>
                </Row>
                <!-- omitted other subjects -->
            </Table>
        </IndivParamsCovariates>
        <DistributionIndivParams>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <Column columnId="V" columnType="indivParameter" valueType="real" columnNum="2"/>
                <Column columnId="CL" columnType="indivParameter" valueType="real" columnNum="3"/>
                <!-- other parameters skipped -->
                <Column columnId="KA" columnType="indivParameter" valueType="real" columnNum="6"/>
                <Column columnId="WT" columnType="covariate" valueType="real" columnNum="7"/>
                <Column columnId="SEX" columnType="covariate" valueType="real" columnNum="8"/>
                <!-- other covariates skipped -->
                <Column columnId="AGE" columnType="covariate" valueType="real" columnNum="11"/>
                <Column columnId="STRATA_DOSE" columnType="strataVariable" valueType="real" columnNum="12"/>
                <!-- other strata skipped -->
                <Column columnId="STRATA_STUDY" columnType="strataVariable" valueType="real" columnNum="13"/>
            </Definition>
            <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <Row>
                    <ct:String>i1</ct:String><ct:Real>1</ct:Real>...<ct:Real>12</ct:Real>
                </Row>
                <Row>
                    <ct:String>i1</ct:String><ct:Real>1.1</ct:Real>...<ct:Real>5.1</ct:Real>
                </Row>
                <!-- omitted other samples and subjects -->
            </Table>
        </DistributionIndivParams>
    </DiagnosticIndividualParams>
</ModelDiagnostic>
```

Notes

- Metadata are required for annotating the methods computing the random effects;
- `STRATA_Covariate_S` will be populated with the value of the covariate used for stratification (more than one stratifying variable is allowed).

4.6 Simulation

In the `<Simulation>` section typical results produced in a simulation task can be stored. In particular, within the `<SimulationBlock>` for each replicate, whose number is kept by the `replicate` attribute, the simulated time course of each subject, population and individual parameters, random effects, covariates, dosing records and reference to raw results are stored. Multiple instances of each element can be stored within this section by specifying the `name` and `extFileNo` attributes.

SO Element	Format
○ SimulationBlock	any number of replicates allowed
– Simulated profiles/time courses♠	`Table` with columns `ID`, `DVID`, `TIME`, `Observation_1`, ..., `Observation_I`
– Individual parameters	`Table` with columns `ID`, `PARAM_1`, ..., `PARAM_N`
– Individual contribution to variabilitycomponents (ETAs)	`Table` with columns `ID`, `ETA_1`, ..., `ETA_J`
– Covariates♡	`Table` with columns `ID`, `Covariate_1`, ..., `Covariate_D`
– Regressors♡	`Table` with columns `ID`, `TIME`, `RegressorName`, `RegressorValue`
– Population parameters	`Table` with columns `POP_PARAM_1`, ..., `POP_PARAM_K`
– Dosing records	`Table` with columns `ID`, `TIME`, `AMT`, `TINF`, `TYPE/ADM`, `SS`, `II`, etc. ...
– Raw result file per replicate	`File Reference`

Table 4.13: Simulation. The occasions level related columns, `OCC`, `OccasionStart` and `OccasionEnd`, are optional and should be used only if such variability level is present in the study. See section 4.4.3 for more details. ♠Note that dosing records should not be included in this table.
♡`TIME` column is required only for regressors (time-varying covariates); for occasion dependent covariates the occasion level related columns can be used as explained above.

Complete template

```
<Simulation>
    <SimulationBlock replicate="1">
        <SimulatedProfiles>...</SimulatedProfiles>
        <IndivParameters>...</IndivParameters>
        <RandomEffects>...</RandomEffects>
        <Covariates>...</Covariates>
        <Regressors>...</Regressors>
        <PopulationParameters>...</PopulationParameters>
        <Dosing>...</Dosing>
        <RawResultsFile></RawResultsFile>
    </SimulationBlock>
    <!-- <SimulationBlock replicate="2">
        ... if required
    </SimulationBlock>-->
</Simulation>
```

Selected examples

The first example shows how to store simulation profiles coming from different files. Moreover, sometimes only selected columns are of interest. Here data from two files, *SimulatedProfiles_file1.txt* and *SimulatedProfiles_file2.txt*, is extracted.

```
<Simulation>
    <SimulationBlock replicate="1">
        <SimulatedProfiles name="SimProfile1" extFileNo="1">
            <ds:Definition>
                <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <ds:Column columnId="TIME" columnType="time" valueType="real" columnNum="2"/>
                <ds:Column columnId="Observation1" columnType="dv" valueType="real" columnNum="5"/>
                <ds:Column columnId="Observation3" columnType="dv" valueType="real" columnNum="29"/>
            </ds:Definition>
            <ds:ExternalFile oid="oid1">
                <ds:path>SimulatedProfiles_file1.txt</ds:path>
            </ds:ExternalFile>
        </SimulatedProfiles>
        <SimulatedProfiles name="SimProfile2" extFileNo="2">
            <ds:Definition>
                <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <ds:Column columnId="TIME" columnType="time" valueType="real" columnNum="2"/>
                <ds:Column columnId="Observation2" columnType="dv" valueType="real" columnNum="4"/>
                <ds:Column columnId="Observation13" columnType="dv" valueType="real" columnNum="9"/>
            </ds:Definition>
            <ds:ExternalFile oid="oid1">
                <ds:path>SimulatedProfiles_file2.txt</ds:path>
            </ds:ExternalFile>
        </SimulatedProfiles>
```

The second example shows how the dosing information is captured

```
<Simulation>
    <SimulationBlock replicate="1">
        <!-- ... -->
        <Dosing>
            <ds:Definition>
                <ds:Column columnId="ID" columnType="id" valueType="string" columnNum="1"/>
                <ds:Column columnId="TIME" columnType="time" valueType="real" columnNum="2"/>
                <ds:Column columnId="AMT" columnType="dose" valueType="real" columnNum="3"/>
                <ds:Column columnId="TINF" columnType="duration" valueType="real" columnNum="4"/>
                <ds:Column columnId="TYPE" columnType="adm" valueType="real" columnNum="5"/>
                <ds:Column columnId="II" columnType="ii" valueType="real" columnNum="6"/>
                <!-- other columns possible/omitted -->
            </ds:Definition>
            <ds:Table>
                <ds:Row>
                    <ct:String>i1</ct:String><ct:Real>1</ct:Real><ct:Real>1</ct:Real><ct:Real>2</ct:Real>
                    <ct:Real>3</ct:Real><ct:Real>4</ct:Real><ct:Real>5</ct:Real><ct:Real>6</ct:Real>
                </ds:Row>
                <ds:Row>
                    <ct:String>i1</ct:String><ct:Real>2</ct:Real><ct:Real>1</ct:Real><ct:Real>2</ct:Real>
                    <ct:Real>3</ct:Real><ct:Real>4</ct:Real><ct:Real>5</ct:Real><ct:Real>6</ct:Real>
                </ds:Row>
                <!-- omitted other TIMES for ID=1 and other ID's -->
                <ds:Row>
                    <ct:String>i10</ct:String><ct:Real>1</ct:Real><ct:Real>1</ct:Real>
                    <ct:Real>2</ct:Real><ct:Real>3</ct:Real><ct:Real>4</ct:Real>
                    <ct:Real>5</ct:Real><ct:Real>6</ct:Real>
                </ds:Row>
                <!-- omitted other TIMES for ID=10 -->
            </ds:Table>
        </Dosing>
```

Notes

- Annotation required for capturing simulation information (e.g., observation level with residual variability).
- Annotation required for pointing to the target tool.
- The occasions level related columns, `OCC`, `OccasionStart` and `OccasionEnd`, are optional and should be used only if such variability level is present in the study. See section 4.4.3 for more details.
- Examples for additional column: level of randomisation – 2 replicates per subject.

- Every element can be composed of multiple Tables/Datasets using the attribute `name` and `extFileNo`, e.g. there might be multiple `<SimulatedProfiles>` with values $1, 2, 3, \ldots$ for the `extFileNo` attribute.

4.7 Optimal Design

Typical results produced by Optimal Design (OD) can be stored here. The `type` attribute of this section allows to distinguish between an evaluation or optimization step. Results coming from more than one OD step can be separately stored in the `<OptimalDesign>` block by specifying the `blockNumber` number.

SO elements in this section allow to capture and store the Fisher Information Matrix, variance/covariance matrix, parameter values and their precision, information about the adopted criteria and performed tests. Moreover, references to the simulated dataset and the final design can be stored within the two defined elements/child-elements `<SimulatedData>` and `<Design>`, respectively.

SO Element	Format
○ Optimal Design Block (any block number allowed)	
– Fisher information matrix (FIM) or expectation of the FIM	`Matrix` with `RowNames`, `ColumnNames`
– Variance/covariance matrix	`Matrix` with `RowNames`, `ColumnNames`
– Parameters values and precision with expected SE, RSE and expected shrinkage	`Table` with columns `Parameter`, `Value`, `SE`, `RSE`, `Shrinkage`
– Criteria	`Table` with columns `Criterium`♠, `Value`, `Comments`
– Tests	`Table` with columns `Parameter`, `Test`♡, `Type`♢, `Value`, `Delta`, `LowerLimit`, `UpperLimit`, `Type1Error`, `PredictedPower`, `TargetPower`, `NumberSubject`
– Simulated datasets (if evaluation done through simulation)	`File Reference`
– Design	`Reference` to `Design` stored in a PharmML file *finalModel1.xml*

Table 4.14: Optimal Design. ♠The `CriteriumName` column can be populated with terms specified in the following controlled list: D-optimality, ED-optimality, A-optimality, lnD-optimality, ElnD-optimality, EID-optimality, HCD-optimality, Ds-optimality, Compound Optimality, InverseRSE, Other. ♡ Only 'Wald' for now. ♢The `TestName` column can be populated with terms specified in the following controlled list: comparison, equivalence, non-inferiority, non-superiority.

Complete template

```
<OptimalDesign type="...">
    <OptimalDesignBlock blockNumber="1">
        <FIM>...</FIM>
        <CovarianceMatrix>...</CovarianceMatrix>
        <ParameterPrecision>...</ParameterPrecision>
        <Criteria>...</Criteria>
        <Tests>...</Tests>
        <SimulatedData>...</SimulatedData>
        <Design>...</Design>
    </OptimalDesignBlock>
</OptimalDesign>
```

Selected examples

Shown are the criteria and tests in SO, other elements are very similar to those described before.

```
<OptimalDesign type="evaluation">
    <OptimalDesignBlock blockNumber="1">
        <Criteria>
            <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
                <ds:Column columnId="Criterium" columnType="undefined" valueType="string" columnNum="1"/>
```

```
            <ds:Column columnId="Value" columnType="undefined" valueType="real" columnNum="2"/>
            <ds:Column columnId="Comments" columnType="undefined" valueType="string" columnNum="3"/>
        </Definition>
        <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Row>
                <ct:String>D-optimality</ct:String><ct:Real>371.3125</ct:Real><ct:String>determinant_of_FIM</ct:String>
            </Row>
        </Table>
    </Criteria>
    <Tests>
        <Definition xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <ds:Column columnId="Parameter" columnType="undefined" valueType="string" columnNum="1"/>
            <ds:Column columnId="Test" columnType="undefined" valueType="string" columnNum="2"/>
            <ds:Column columnId="Type" columnType="undefined" valueType="string" columnNum="3"/>
            <ds:Column columnId="Value" columnType="undefined" valueType="real" columnNum="4"/>
            <ds:Column columnId="Delta" columnType="statPrecision" valueType="real" columnNum="5"/>
            <ds:Column columnId="LowerLimit" columnType="statPrecision" valueType="real" columnNum="5"/>
            <ds:Column columnId="UpperLimit" columnType="statPrecision" valueType="real" columnNum="6"/>
            <ds:Column columnId="Type1Error" columnType="statPrecision" valueType="real" columnNum="7"/>
            <ds:Column columnId="PredictedPower" columnType="statPrecision" valueType="real" columnNum="8"/>
            <ds:Column columnId="TargetPower" columnType="statPrecision" valueType="real" columnNum="9"/>
            <ds:Column columnId="NumberSubject" columnType="undefined" valueType="real" columnNum="10"/>
        </Definition>
        <Table xmlns="http://www.pharmml.org/pharmml/0.8/Dataset">
            <Row>
                <ct:String>beta_V_Sex_F</ct:String><ct:String>Wald</ct:String><ct:String>equivalence</ct:String>
                <ct:Real>0.1823216</ct:Real><ct:Real>-0.2231436</ct:Real><ct:Real>0.2231436</ct:Real>
                <ct:Real>0.05</ct:Real><ct:Real>0.1042397</ct:Real><ct:Real>0.9</ct:Real><ct:Real>2286.081</ct:Real>
            </Row>
            <Row>
                <ct:String>beta_Cl_Treat_B</ct:String><ct:String>Wald</ct:String><ct:String>equivalence</ct:String>
                <ct:Real>0.09531018</ct:Real><ct:Real>-0.2231436</ct:Real><ct:Real>0.2231436</ct:Real>
                <ct:Real>0.05</ct:Real><ct:Real>0.9818745</ct:Real><ct:Real>0.9</ct:Real><ct:Real>24.50351</ct:Real>
            </Row>
            <Row>
                <ct:String>beta_V_Sex_F</ct:String><ct:String>Wald</ct:String><ct:String>comparison</ct:String>
                <ct:Real>0.4054651</ct:Real><ct:Real>0.050000</ct:Real><ct:Real>0.978421</ct:Real><ct:Real>0.9</ct:Real>
                <ct:Real>26.50458</ct:Real><ct:Real>0.9</ct:Real><ct:Real>24.50351</ct:Real>
            </Row>
        </Table>
    </Tests>
```

Appendix A

Document Revision History

Version	Authors	Date	Comments
0.0.1	CS & MS	30 Apr 2014	First document combining Marc's and Mike's proposals.
0.0.2	CS & MS	26 Jun 2014	Draft proposal on SO send out to the SO group.
0.0.3	CS & MS	8 Jul 2014	Updated version after TC and incorporation of feedback.
0.0.4	NT	30 Jul 2014	Comments after testing SO with provided examples.
0.0.5	MS & NT	5 Sept 2014	Update after TC.
0.0.6	NT & MS	12 Sept 2014	Final corrections after feedback from Emma.
0.0.7	NT & MS	17 Oct 2015	New section *Tool settings* and changes in simulation and optimal design sections.
0.0.8	NT & MS	8 Dec 2015	New section *Task Information*. Suggestions from Kajsa and Rikard incorporated.
0.0.9	NT & MS	6 Jan 2015	Changes in model diagnostic and simulation sections after feedback from Marc and Mike.
0.0.10	NT & MS	15 Jan 2015	More changes in model diagnostic section, e.g. strata added. Dosing records in simulation section.
0.0.11	NT & MS	12 Feb 2015	More minor corrections/changes e.g. relocated shrinkage elements.
0.1	NT & MS	16 Feb 2015	Removed redundant OriginalDataset element in Simulation section, renamed Residual to Residuals.
0.1.1	NT & MS	11 Mar 2015	Optimal Design extensions implemented. SO schema updated and minor edits.
0.1.2	NT & MS	14 Apr 2015	New posterior distribution elements in bootstrap element and related, new attribute in simulation child-elements.
0.2	NT & MS	22 Apr 2015	Multiple tool-setting files allowed, adding footnotes on dosing records inclusion, minor corrections.
0.2.1	NT	10 Jun 2015	New DVID column and strata variable column in residual and prediction section.
0.2.2	NT	10 Jun 2015	Changes in model elements and revision of model diagnostic section.
0.2.3	MS & FY	20 Nov 2015	Rikard's suggestions for simplification of *Task Information* implemented.
0.2.3	MS & FY	26 Nov 2015	Renamed from Standardised Output to Standard Output.
0.2.4	NT & MS	2 Dec 2015	Fixed higher variability notation, removed redundant columns.
0.2.5	NT & MS	10 Dec 2015	Implementing Rikard's suggestions that all blocks are now mandatory.
0.2.6	NT & MS	16 Jan 2016	Diagnostic section changes; Likelihood block renamed to OFMeasures and new elements added (Likelihood and ToolObjFunction).
0.3	NT & MS	18 Jan 2016	Minor corrections and updates.
0.3.1	NT & MS	16 Mar 2016	Support for higher levels of variability and integer definition change.

Table A.1: CS=Celine Sarr, FY=Florent Yvon, NT=Nadia Terranova, MS=Maciej Swat.

Bibliography

[Bazzoli et al., 2010] Bazzoli, C., Retout, S., and Mentré, F. (2010). Design evaluation and optimisation in multiple response nonlinear mixed effect models: Pfim 3.0. *Computer Methods and Programs in Biomedicine*, 98(1):55–65.

[D'Argenio and Wang, 2009] D'Argenio, D.Z., A. S. and Wang, X. (2009). Adapt 5 user's guide: Pharmacokinetic/pharmacodynamic systems analysis software. Technical report, Biomedical Simulations Resource, Los Angeles.

[Harnisch et al., 2013] Harnisch, L., Matthews, I., Chard, J., and Karlsson, M. O. (2013). Drug and disease model resources: a consortium to create standards and tools to enhance model-based drug development. *CPT Pharmacometrics Syst Pharmacol*, 2:e34.

[Lavielle, 2014] Lavielle, M. (2014). Tasks, Methods, Inputs and Output. Technical report, INRIA Saclay & EMBL-EBI.

[Lindbom et al., 2004] Lindbom, L., Ribbing, J., and Jonsson, E. N. (2004). Perl-speaks-nonmem (psn)–a perl module for nonmem related programming. *Comput Methods Programs Biomed*, 75(2):85–94.

[Lixoft Team, 2013] Lixoft Team (2013). MLXTRAN, The model coding language for MLXPLORE. Technical report, INRIA Saclay & Lixoft.

[Lixoft Team, 2014] Lixoft Team (2014). MLXTRAN, The model coding language for Simulator. Technical report, INRIA Saclay & Lixoft.

[Nyberg et al., 2012] Nyberg, J., Ueckert, S., Strömberg, E. A., Hennig, S., Karlsson, M. O., and Hooker, A. C. (2012). Poped: an extended, parallelized, nonlinear mixed effects models optimal design tool. *Comput Methods Programs Biomed*, 108(2):789–805.

[Smith, 2014] Smith, M. K. (2014). Standard Output Object specification. Technical report, Pfizer.

[Swat, 2015] Swat, M. J. (2015). Extentions in PharmML 0.7.3. Technical report, EMBL-EBI.

[Swat et al., 2016a] Swat, M. J., Grenon, P., and Wimalaratne, S. M. (2016a). ProbOnto 1.1 - Ontology and Knowledge Base of Probability Distributions. Technical report, EMBL-EBI, Hinxton, UK; UCL, UK.

[Swat et al., 2015a] Swat, M. J., Grenon, P., Yvon, F., Wimalaratne, S., and Kristensen, N. R. (2015a). Extentions in PharmML 0.7-0.7.2. Technical report, EMBL-EBI.

[Swat et al., 2015b] Swat, M. J., Moodie, S., Wimalaratne, S. M., Kristensen, N. R., Lavielle, M., Mari, A., Magni, P., Smith, M. K., Bizzotto, R., Pasotti, L., Mezzalana, E., Comets, E., Sarr, C., Terranova, N., Blaudez, E., Chan, P., Chard, J., Chatel, K., Chenel, M., Edwards, D., Franklin, C., Giorgino, T., Glont, M., Girard, P., Grenon, P., Harling, K., Hooker, A. C., Kaye, R., Keizer, R., Kloft, C., Kok, J. N., Kokash, N., Laibe, C., Laveille, C., Lestini, G., Mentré, F., Munafo, A., Nordgren, R., Nyberg, H. B., Parra-Guillen, Z. P., Plan, E., Ribba, B., Smith, G., Trocóniz, I. F., Yvon, F., Milligan, P. A., Harnisch, L., Karlsson, M., Hermjakob, H., and Le Novère, N. (2015b). Pharmacometrics Markup Language (PharmML): Opening New Perspectives for Model Exchange in Drug Development. *CPT Pharmacometrics Syst Pharmacol*, 4(6):316–9.

[Swat et al., 2015c] Swat, M. J., Wimalaratne, S. M., Kristensen, N. R., Yvon, F., Moodie, S., and Le Novère, N. (2015c). Pharmacometrics Markup Language (PharmML), Language Specification for Version 0.6.

[Swat et al., 2016b] Swat, M. J., Yvon, F., and Wimalaratne, S. (2016b). Extentions in PharmML 0.8-0.8.1. Technical report, EMBL-EBI.

[Terranova et al., 2016] Terranova, N., Lavielle, M., Smith, M. K., Comets, E., Nordgren, R., Edwards, D., Sarr, C., Harling, K., Hooker, A. C., Mentré, F., Yvon, F., and Swat, M. J. (2016). Standard Output (SO): Format Specification for Version 0.3. Technical report, DDMoRe SO group.

[Terranova et al., 2015] Terranova, N., Swat, M. J., Lavielle, M., Smith, M. K., Comets, E., Nordgren, R., Sarr, C., Harling, K., Hooker, A. C., and Mentré, F. (2015). Standardised Output Specification, Version 0.1. Technical report, DDMoRe SO group.