PharmML – An Exchange Standard for Models in Pharmacometrics

Maciej Swat (1), Stuart Moodie (1), Niels Rode Kristensen (2), Nicolas Le Novère (1,3)
on behalf of DDMoRe WP4 contributors
(1) EMBL-EBI, Hinxton UK, (2) Novo Nordisk, Denmark, (3) Babraham Institute, UK

Introduction

A long-standing problem in Pharmacometrics is the lack of a common standard allowing for exchangeability of models between existing software tools, such as Bugs, Monolix, NONMEM and others. PharmML, as part of the DDMoRe interoperability platform, presented here tries to fill this gap.

PharmML stands for 'Pharmacometrics Markup Language'

The modelling framework is that of Nonlinear Mixed Effects Models, NLME, which allows for nonlinear models with random and fixed effects. This new standard provides encoding platform for approaches currently in use but also attempts to create support for novel elements.

Results

The current specification supports the exchange of continuous models. Models encoded in this way can be used not only for the standard tasks, such as simulation and estimation but also modelling and exploration.
The novel clinical trial model provides the modeller with the tools to construct almost arbitrary study designs using only few basic building blocks, such as Treatment, Epoch and Group.
Moreover, PharmML is providing a means to annotate an arbitrary element of the model, making effective searching and reasoning on models in a repository possible

Schematic representation of the interoperability platform

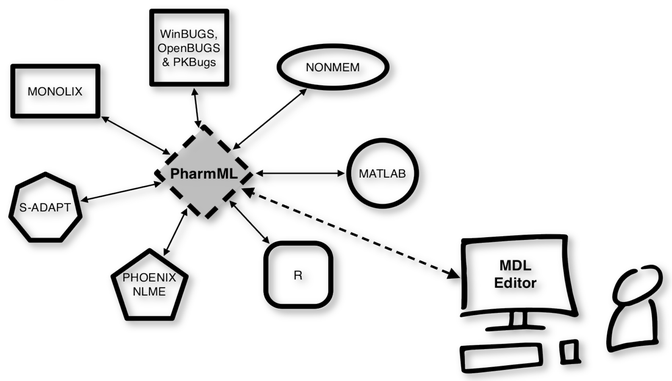

Figure 1: PharmML is one of the key elements of the interoperability platform developed within DDMoRe. Its main objectives are: encoding of models, trial design and basic tasks. It provides also the means for model annotation.

XML (Extensible Markup Language) – the backbone of PharmML

PharmML is a XML based markup language for representing, reading and writing pharmacometrics models. XML has following desirable features:

- Different software tools can directly communicate and store the same computable representation of those models,
- It is an *extensible* language because it allows the users to define their own tags,
- It is both human-readable/legible and machine-readable,
- XML, and therefore PharmML, has the advantages of hierarchical structure, simplicity, arbitrary extensibility, interoperability,
- It is used in many other domains, e.g. HTML (webpages), SVG (graphics), XLSX (spreadsheets),
- There exist a number of software tools to generate, interpret or process XML

Some established XML based standard are SBML – for representing biochemical networks or UncertML – for encapsulating probabilistic uncertainties [1,2].

PharmML organization

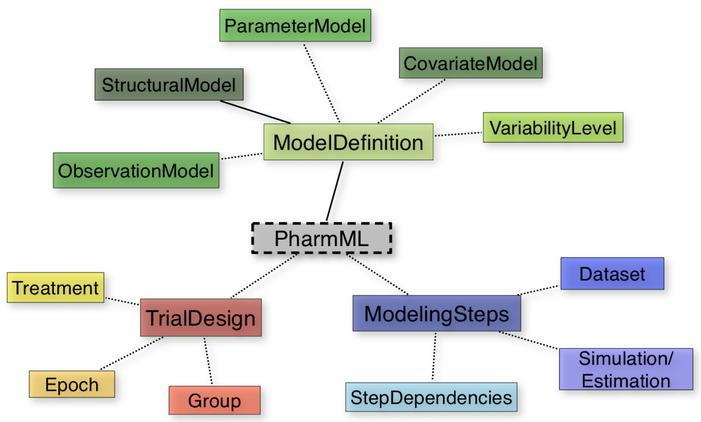

Figure 2: There are three basic blocks in PharmML: 'ModelDefinition', 'TrialDesign' And 'ModelingSteps'

PharmML model in editor view

```
<?xml version="1.0" encoding="UTF-8"?>
<PharmML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.pharmml.org/2013/03/PharmML"
    xsi:schemaLocation="http://www.pharmml.org/2013/03/PharmML http://www.pharmml.org/2013/03/PharmML"
    xmlns:math="http://www.pharmml.org/2013/03/Maths"
    xmlns:ct="http://www.pharmml.org/2013/03/CommonTypes"
    xmlns:mdef="http://www.pharmml.org/2013/03/ModelDefinition"
    xmlns:mstep="http://www.pharmml.org/2013/03/ModellingSteps"
    name="Example 1 - continuous PK/PD"
    independentVar="t" writtenVersion="0.1">
    <!-- SYMBOL DEFINITION - RESUIDUAL ERROR MODEL -->
    <SymbolDefinition symbId="constantErrorModel" symbolType="scalar"> [9 lines]
    <!-- BLOCK I: MODEL DEFINITION -->
    <ModelDefinition  xmlns="http://www.pharmml.org/2013/03/ModelDefinition">
        <!-- VARIABILITY MODEL -->
        <VariabilityLevel id="indiv"/>
        <!-- COVARIATE MODEL -->
        <CovariateModel id="cm1"> [37 lines]
        <!-- PARAMETER MODEL -->
        <ParameterModel id="pm1"> [254 lines]
        <!-- STRUCTURAL MODEL - ODEs -->
        <StructuralModel id="sm1"> [136 lines]
        <!-- OBSERVATION MODEL - PSTAR -->
        <ObservationModel id="om1"> [26 lines]
    </ModelDefinition>
    <!-- BLOCK II: MODELING STEPS -->
    <ModellingSteps xmlns="http://www.pharmml.org/2013/03/ModellingSteps"> [178 lines]
</PharmML>
```

Figure 3: Three first layers of the PharmML hierarchical structure are shown: (a) the root level 'PharmML', (b) second level with 'SymbolDefinition', 'ModelDefinition' and 'ModellingSteps', (c) third level within 'ModelDefinition'

Use Case: Tumour growth model

The model shown in Figure 4 was proposed by Ribba et al. 2012 [3]; it describes tumour growth inhibition for low-grade glioma upon chemotherapeutic treatment. It can be formulated in PharmML by following elements:

- '*StructuralModel*' - the kinetic of the tumour tissue growth and K-PK model are defined using an ODE system and one algebraic equation.
- '*ParameterModel*' - individual parameters which follow log-normal distribution,
- '*ObservationModel*' - an additive residual error model is assumed.

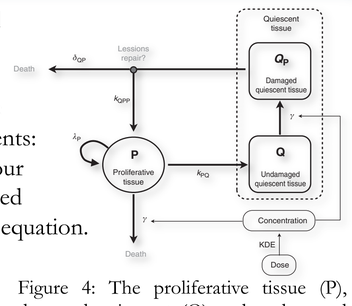

Figure 4: The proliferative tissue (P), damaged quiescent (Q) and undamaged quiescent tissue (Q_P) are visualized beside the K-PK model for the drug cocktail.

Structural Model

$$f(x_{ij}, \psi_i)$$

This structural model can be formulated as a simple algebraic equation (e.g. Hill equation) or complex physiology-based PK model implemented as system of ODEs. When defined in such framework, this deterministic model for an individual will later be embedded in a statistical model.

```
<!-- dQ/dt -->
<Variable symbId="Q" independentVar="t" symbolType="derivative">
    <Equation writtenVersion="0.1" xmlns="http://www.pharmml.org/2013/03/Maths">
        <Binop op="minus">
            <Binop op="times">
                <Var block="pm1" symbId="KPQ"/>
                <Var symbId="PT"/>
            </Binop>
            <Binop op="times">
                <Var block="pm1" symbId="GAMA"/>
                <Binop op="times">
                    <Var symbId="C"/>
                    <Binop op="times">
                        <Var block="pm1" symbId="KDE"/>
                        <Var symbId="Q"/>
                    </Binop>
                </Binop>
            </Binop>
        </Binop>
    </Equation>
</Variable>
```

$$\frac{dQ}{dt} = k_{QP} - \gamma \times C \times KDE \times Q$$

```
<!-- PSTAR - algebraic equation -->
<Variable symbId="PSTAR" symbolType="scalar">
    <Equation xmlns="http://www.pharmml.org/2013/03/Maths" writtenVersion="0.1">
        <Binop op="plus">
            <Var symbId="PT"/>
            <Binop op="plus">
                <Var symbId="Q"/>
                <Var symbId="QP"/>
            </Binop>
        </Binop>
    </Equation>
</Variable>
```

$$P^\star = P + Q + Q_P$$

Parameter Model

The parameter model offers a rich and flexible structure allowing for the implementation of continuous and discrete covariates, correlation structure of the random effects and virtually any level of variability as nested hierarchy.
E.g. the parameter model for *KDE* with log-normal distribution and *Weight* as covariate reads

$$\log(KDE_i) = \log(KDE_{pop}) + \beta_{1,W} \log(W_i/70) + \eta_{KDE}$$

or $$KDE_i = KDE_{pop}(W_i/70)^{\beta_{1,W}} e^{\eta_{KDE}} \quad \text{with} \quad \eta_{KDE} \sim \mathcal{N}(0, \omega_{KDE})$$

```
<!-- KDE log-normal distributed -->
<Parameter symbId="KDE" transformation="log">
    <Equation writtenVersion="" xmlns="http://www.pharmml.org/2013/03/Maths">
        <Uniop op="ln">
            <Var symbId="pop_KDE"/>
        </Uniop>
    </Equation>
    <RandomEffect symbId="eta_KDE" levelId="indiv">
        <Distribution xmlns="http://www.pharmml.org/2013/03/Uncertainty" writtenVersion="0.1">
            <Normal>
                <Mean>
                    <Equation xmlns="http://www.pharmml.org/2013/03/Maths" writtenVersion="0.1">
                        <math:Scalar value="0"/>
                    </Equation>
                </Mean>
                <StdDev>
                    <Equation xmlns="http://www.pharmml.org/2013/03/Maths" writtenVersion="0.1">
                        <Var symbId="omega_KDE"/>
                    </Equation>
                </StdDev>
            </Normal>
        </Distribution>
    </RandomEffect>
    <Covariate>
        <math:Var block="cm1" symbId="W"/>
        <FixedEffect>
            <math:Var symbId="beta1_W"/>
        </FixedEffect>
    </Covariate>
</Parameter>
```

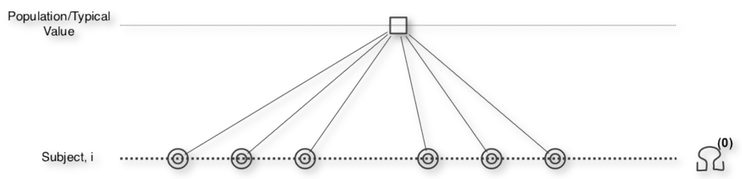

Figure 5: In the presented use case there is only one level of variability – the inter-individual variability (IIV), see the right panel for the 'hierarchical structure of variability '.

Residual error model

$$g(x_{ij}, \psi_i, \xi)\epsilon_{ij}, \quad \epsilon_{ij} \sim \mathcal{N}(0,1)$$

The observation model for continuous models provides the structure to implement the residual error model. In this example the additive error model, $g = a$, is applied, i.e.

$$y_{ij} = f(x_{ij}, \psi_i) + a\ \epsilon_{ij}$$

The PharmML implementation reads

```
<!-- OBSERVATION MODEL - PSTAR -->
<ObservationModel id="om1">
    <Parameter symbId="a"/>
    <Continuous symbId="PSTAR">
        <math:Var block="sm1" symbId="PSTAR"/>
        <ErrorModel>
            <math:Var symbId="constantErrorModel"/>
            <math:FunctionArgument symbId="a">
                <math:Scalar value="1"/>
            </math:FunctionArgument>
            <math:FunctionArgument symbId="f">
                <math:Var symbId="PSTAR"/>
            </math:FunctionArgument>
        </ErrorModel>
        <RandomEffect>
            <Distribution xmlns="http://www.pharmml.org/2013/03/Uncertainty" writtenVersion="0.1">
                <Normal>
                    <Mean>
                        <math:Scalar value="0"/>
                    </Mean>
                    <StdDev>
                        <math:Scalar value="1"/>
                    </StdDev>
                </Normal>
            </Distribution>
        </RandomEffect>
    </Continuous>
</ObservationModel>
```

Non-linear mixed effects model for continuous data

The general NLME model for N subjects and n_i measurements per subject i reads as follows

$$\underbrace{y_{ij}}_{\text{Experimental data}} = \underbrace{f(x_{ij}, \psi_i)}_{\text{Model prediction}} + \underbrace{g(x_{ij}, \psi_i, \xi)\ \epsilon_{ij}}_{\text{Residual error}}, \quad 1 \le i \le N,\ 1 \le j \le n_i$$

- y_{ij} - j^{th} observation for subject i
- f - structural model prediction
- x_{ij} - regression variables
- ψ_i - individual parameters
- g - standard deviation of the residual error
- ξ - parameters of the residual error
- ϵ_{ij} - residual error

Variability – as nested hierarchical structure

Nested hierarchy (also known as inclusion hierarchy) provides the means to define the variability structure of random effects. It can be visualized as a tree, see Figure 7, or alternatively using a Venn diagram.

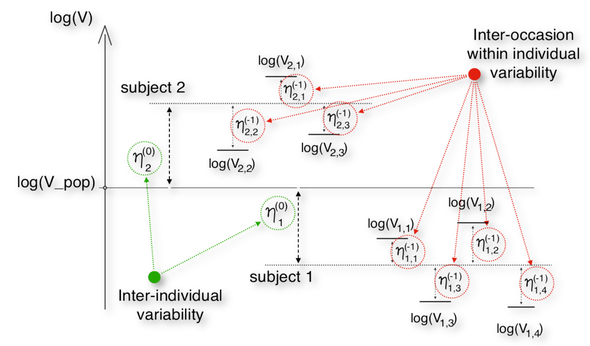

Figure 6: Visualized is subject variability level, (0), and inter-occasion variability level, (-1), typically occurring in an experiment, with index i for subjects and k for occasion. Here data for two subjects only is shown, each of them having four or three occasions, respectively.

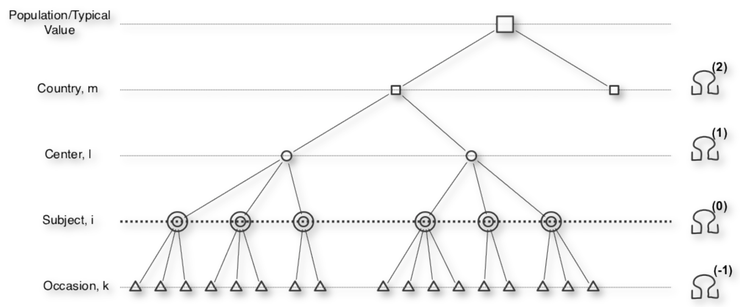

Figure 7: Tree representation of the variability structure for a complex example. Usually, the variability structure consists of only one or two levels, e.g. individual or {individual, occasion}.

The root, i.e. the top node in the tree structure, stands for the population/typical value of a parameter. Every subsequent variability level is either 'positive' or 'negative' dependent on its position relative to the 'subject level', denoted as 0 – the level 'zero'. Each level has a covariance matrix associated with it, i.e.

- $\Omega^{(+n)}$ – for levels above the 'zero' level
- $\Omega^{(0)}$ – also called BSV (between subject variability) or IIV (inter-individual variability)
- $\Omega^{(-n)}$ – for levels below the 'zero' level – called WSV (within-subject variability) or IOV (inter-occasion variability)

Trial Design

Until now the common practice was to encode the trial design in experimental data files. PharmML offers an alternative – a very flexible structure for the setup of clinical trials, especially for simulations. Using only a few basic elements the modeler can compose several types of designs, see Figure 8 for few examples.
The basic building blocks are:

1. *Treatment* – describes dosing related data – supporting definition of (a) Administration/dosing type, (b) Dosing times/amount and (c) Dosing target,
2. *TreatmentEpoch* – basic time interval within a study – with elements such as (a) Epoch name, e.g. 'Treatment A' or 'Washout' or 'Run-in', (b) Start time and end time of an epoch and (c) Occasion,
3. *Group* – basic grouping structure for subjects.

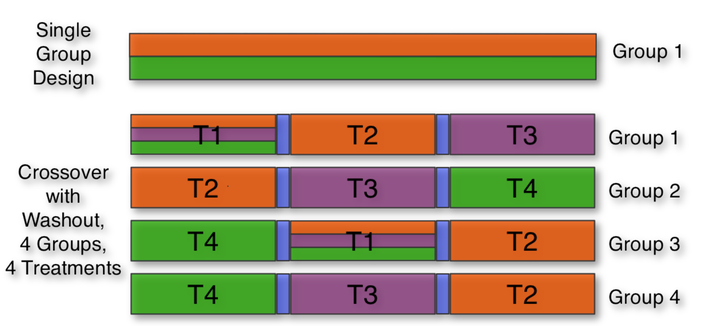

Figure 8: Examples of different trial designs. It is possible to construct wide range of study types by using three basic PharmML elements: 'Treatment'. 'Epoch' and 'Group'. Here two examples are shown, (top) single group design with two types of treatment running in parallel and (bottom) a crossover design with 'washout' (blue) between the treatments for four groups receiving four different treatment types, T1-T4.

References

[1] Hucka, M., Finney, A. et al. (2003). The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. Bioinformatics, vol. 19, no. 4, pp. 524–531.
[2] Williams, M. Cornford, D. et al. (2009) Uncertainty Markup Language (UncertML), OGC Discussion Paper. URL: www.UncertML.org.
[3] Ribba et al. (2012). A tumor growth inhibition model for low-grade glioma treated with chemotherapy or radiotherapy. Clin Cancer Res. 2012 Sep 15;18(18):5071-80.

Acknowledgement: The research leading to these results has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement n° 115156, resources of which are composed of financial contribution from the European Union's Seventh Framework Programme (FP7/2007-2013) and EFPIA companies' in kind contribution. The DDMoRe project is also financially supported by contributions from Academic and SME partners.