Citation: CPT Pharmacometrics Syst. Pharmacol. (2018) 7, 543–546; doi: 10.1002/psp4.12339

PERSPECTIVES

The Standard Output: A Tool-Agnostic Modeling Storage Format

Nadia Terranova[1]*, Mike K. Smith[2], Rikard Nordgren[3], Emmanuelle Comets[4,5], Marc Lavielle[6], Kajsa Harling[3], Andrew C. Hooker[3], Celine Sarr[7], France Mentré[4,5], Florent Yvon[8,9] and Maciej J. Swat[8,10]

New standards have been recently defined and implemented enabling a reliable exchange of pharmacometric models across software tools, and facilitating collaborative drug and disease modeling and simulation (M&S) activities. Among these, the Standard Output (SO) has been proposed as the tool-independent exchange and storage format for typical M&S results. The SO integration within the Drug Disease Model Resource (DDMoRe) interoperability framework (IOF) has already shown its potential for enabling effective data flow across modeling tasks and facilitating information retrieval.

DEVELOPING A STANDARD OUTPUT FORMAT TO EFFECTIVELY SUPPORT PHARMACOMETRIC WORKFLOW INTERACTIONS

Pharmacometric M&S input to study design and decision making frequently requires a variety of methods, potentially using disparate software tools.[1–5] There is generally a lack of consistency across these tools in model definition, data format, numerical methods, and output of results. This limits the potential reuse of models and methods across tools by the analyst, who would then need to understand varying model and tool-specific requirements to correctly translate and recode models and data. This lack of standard languages and interoperability across tools also significantly limits the integration of tools into seamless pharmacometric workflow as well as the sharing of existing knowledge through shared code.

The DDMoRe project has addressed these shortcomings through the definition, implementation and integration of a set of standards into the so-called IOF.[6–9] Among these, the SO has been proposed as the tool-independent format for storing output typically produced from pharmacometric M&S tasks and workflows to guide the model building through assessment of the goodness-of-fit between the model and the dataset and the appropriateness of the underlying model assumptions, as well as to allow the reproducibility of the results. Developing a standardized output format across M&S tools allows better sharing of information, reusable code, and better integration of a wide variety of tools within a single pharmacometric workflow. Although some initiatives aim at improving the standardization of input M&S data (e.g., CDISC, COMBINE, and ISoP Data Standards Working Group), to our knowledge, the SO represents the first standard format proposed for M&S output data.

SO AS A FLEXIBLE STORAGE STRUCTURE FOR TYPICAL RESULTS OF M&S ANALYSIS

The SO is an XML-based exchange format intended for storage of results in a standardized form. It is complementary to the Pharmacometrics Markup Language (PharmML),[8] representing the exchange medium for mathematical and statistical models across various target tools. The SO is based on a hierarchical structure defined to separate independent output information or results obtained from different modeling tasks (e.g., estimation, simulation, and optimal design), whereas standardizing and providing consistent definition of output structure from different estimation methods and tools (e.g., for maximum likelihood and stochastic or sample-based estimation). The standardization is intended to be method specific, rather than software specific. As shown in **Figure 1**, it consists of seven main sections that comprise specialized SO elements, with varying number of levels. A tree plot of the SO structure is also provided as **Appendices S2**.

The *Tool Settings* and *Raw Results* sections are designed to store references to related files. Specifically, the first one keeps the location to files containing tool settings (e.g., task related and graphical) of the performed task. Original data and graphical output files produced by target tools (e.g., WINBUGS, Monolix, NONMEM, PsN, PopED, and PFIM) can be retrieved within the respective elements of the *Raw Results* section.

In the *Task Information* section, five elements provide the user with information about the modeling step execution: tool message and diagnostic information, execution time, the path pointing to the original output file, number of chains (in case of Markov Chain Monte Carlo (MCMC) execution), and iterations. Warning contents, errors, and

[1]Merck Institute for Pharmacometrics, Merck Serono S.A., Zurich, Switzerland “a Subsidiary of Merck KGaA, Darmstadt, Germany”; [2]Global Clinical Pharmacology, Pfizer, Sandwich, UK; [3]Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden; [4]INSERM, IAME, UMR 1137, Paris, France; [5]University Paris Diderot, IAME, UMR 1137, Sorbonne Paris Cité, Paris, France; [6]Inria Ile-de-France and Ecole Polytechnique, Paris, France; [7]Pharmetheus, Uppsala, Sweden; [8]EMBL - European Bioinformatics Institute, Cambridge, UK; [9]Barcelona Supercomputing Center, Barcelona, Spain; [10]Simcyp (a Certara company), Sheffield, UK. *Correspondence Nadia Terranova (nadia.terranova@merckgroup.com)

Received on 17 April, 2018; Accepted 5 July, 2018; published online on 12 Aug 2018. doi:10.1002/psp4.12339

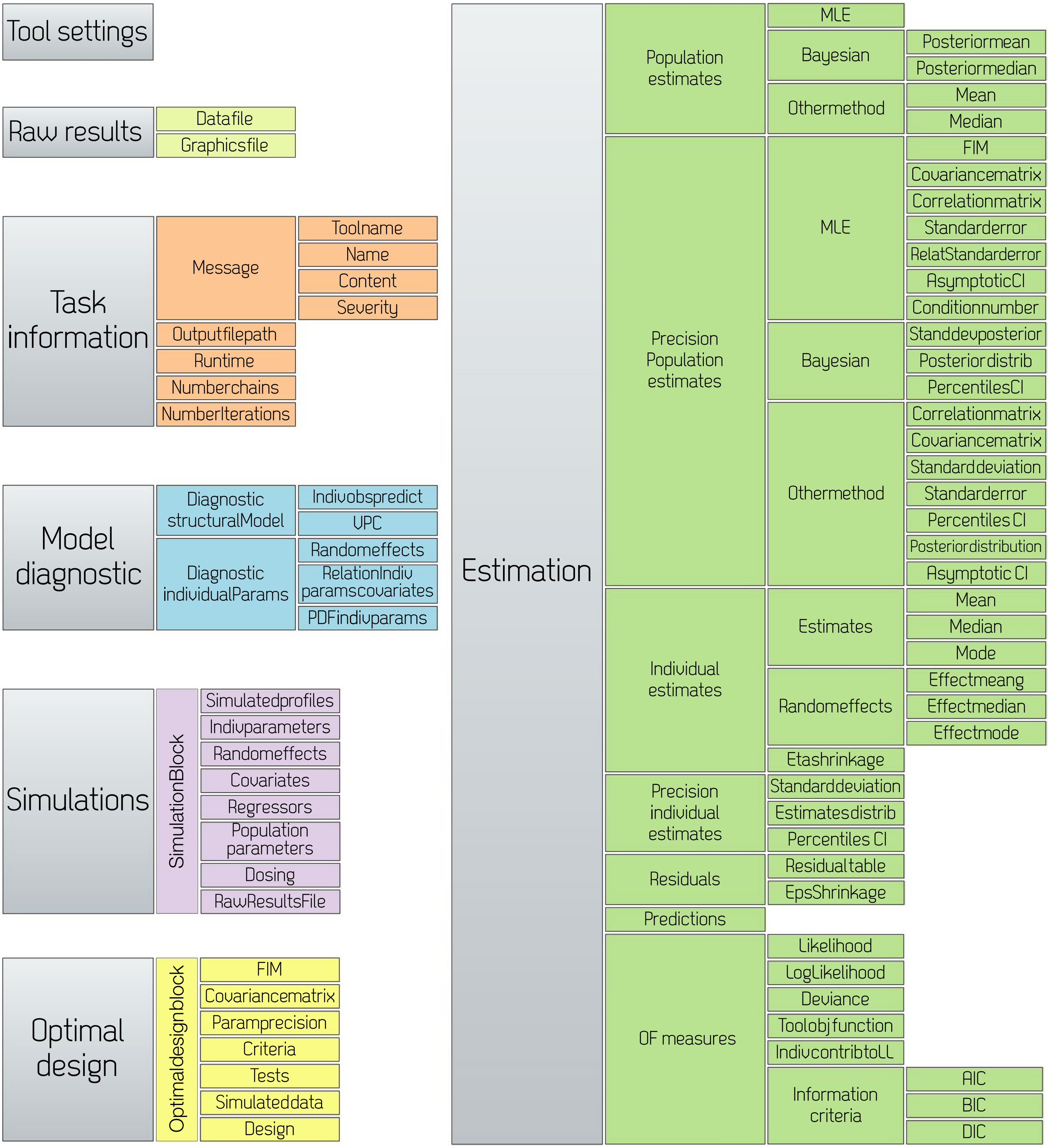

Figure 1 Overview of Standard Output (SO) hierarchical structure consisting of seven main sections (in gray) in turn composed of elements and child elements (from left columns to the right one). A detailed description of SO elements and format specification is reported in the user guide provided as **Appendices S1**. with authors permission. Figure reproduced from the SO user guide version 0.3.1. AIC, Akaike information criterion; BIC, Bayesian information criteria; CI, confidence interval; DIC, Deviance Information Criterion; FIM, Fisher Information Matrix; MLE, maximum likelihood estimation; OF, Objective Function; PDF, probability density function VPC, Visual Predictive Check.

program termination are captured here within a dedicated element (<*Message*>).

The *Estimation* section is designed to capture typical results from an estimation step and it is composed of seven main elements. Four elements allow storage of the population parameter estimates and individual parameter estimates along with their uncertainty measures (e.g., covariance, correlations, standard errors, and confidence intervals). Different child-elements are used to account for different techniques (e.g., maximum likelihood estimateion (MLE) and Bayesian), for different measures of central tendency in sampled-based estimation methods and for different variance components (e.g., interindividual/interoccasion variability). In addition, multiple types of residuals (e.g., population residuals, individual weighted residuals, and conditional weighted residuals), and individual predictions can be stored within two separate elements. Finally, the <*OFMeasures*> element contains 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., Akaike information criterion, Bayesian information criteria, and Deviance Information Criterion).

To support result's reproducibility, a dedicated section, which we called *Model Diagnostic*, is designed for storing information for typical model diagnostic plots used to assess the structural model and residual error specification (e.g., individual observations vs. predictions and visual predictive check), and diagnostic plots for individual parameters. This allows the user to use these values in creation of diagnostic plots without further data manipulation.

The *Estimation* and *Model Diagnostic* sections can store information on parameters at arbitrary variability levels, and are not limited to interindividual variability. In fact, any level below (e.g., occasion) or above (e.g., country or center) the subject level can be considered. Values for relevant random effects are captured using the “columnType” and “level” attributes.

In the *Simulation* section, results from simulations can be captured for each replicate by using separate blocks of child elements. These include the simulated time course of each subject, population and individual parameters, random effects, covariates, dosing records, and references to raw results. This allows prediction of outcomes for a single set of model inputs as well as clinical trial simulations that could incorporate variability in model inputs and parameters.

Results from both evaluation and optimization design steps can be stored within the *Optimal Design* section. Similar to the *Simulation* section, this was structured to capture results from single as well as multiple evaluation or optimization steps in separate blocks. The Fisher Information Matrix, variance/covariance matrix, parameter values along with their precision, information about the adopted criteria, and performed tests, are some of the output information stored within the defined child elements.

POPULATING SO

The SO has been designed to allow flexibility in populating its elements. Sections or elements can be omitted if these are not available or not appropriate for a given target software. In addition, elements can be populated through the defined XML structure (e.g., as inline data or by referring to external files). Unique data types were defined for each SO element reusing some of the structures in the PharmML schema (e.g., matrices and datasets) and distribution specifications from ProbOnto (e.g., posterior distributions).[8,9] For data stored in a tabular form, the *columnType* attribute was defined to support column descriptors (e.g., individual parameter, population parameter, and random effect) that improve usability and interpretation of stored values. As an example, the attribute of a column relative to a variability parameter would be specified to clarify whether this reports a variance, standard deviation, covariance, or correlation measure.

A detailed description of the structure, features, and XML format specifications can be found in the SO user guide available at https://www.ddmore.foundation/our-standards/ and http://www.pharmml.org/. The user guide of SO specification format version 0.3.1 used in the official IOF public release is also provided as **Appendices S1** with copyright permissions.

LibSO, a Java library, was developed to provide basic programmatic functionality for creation and validation of SO data XML documents. The library 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, executable via the command line, is also provided (https://sourceforge.net/p/ddmore/libpharmml/wiki/Home/). An alternative library *libsoc*, developed in C, is also available. It facilitates bindings to multiple languages and an R package based on this library is available on CRAN (https://CRAN.R-project.org/package=libsoc). The R package translates all the data types of the SO into native R data types.

INTEROPERABILITY AND FUTURE PLANS

As one of the core standards of the DDMoRe IOF, use of the SO has been integrated in the DDMoRe R package, and it has shown its benefit in effectively supporting pharmacometric workflow interactions and information retrieval.[10] The SO elements could be populated following estimation and model qualification tasks in PsN, NONMEM, Monolix, and BUGS, and simulation tasks in NONMEM and PsN, thanks to the developed converters and connectors. The SO facilitates interoperability across tools in an analogous way to PharmML, because we can easily pass output information between tools by extracting or converting the information in the SO as input for the next tool. A simplified representation of this workflow, also involving additional tools, is shown in **Figure 2**.

Larizza *et al.*[10] have shown how the DDMoRe standards, including SO, facilitate complex workflow incorporating BUGS and NONMEM to compare methodological approaches. Example workflows available with the DDMoRe IOF show how the SO can be used to update model parameters following estimation and to facilitate model diagnostics in a tool independent way. Examples of this are provided as **Appendices S3**.

CONCLUSIONS AND PERSPECTIVES

SO provides a standard for output information produced in a pharmacometric workflow. It has demonstrated effective data flow across pharmacometric tasks, between software tools, and facilitates information retrieval for postprocessing,

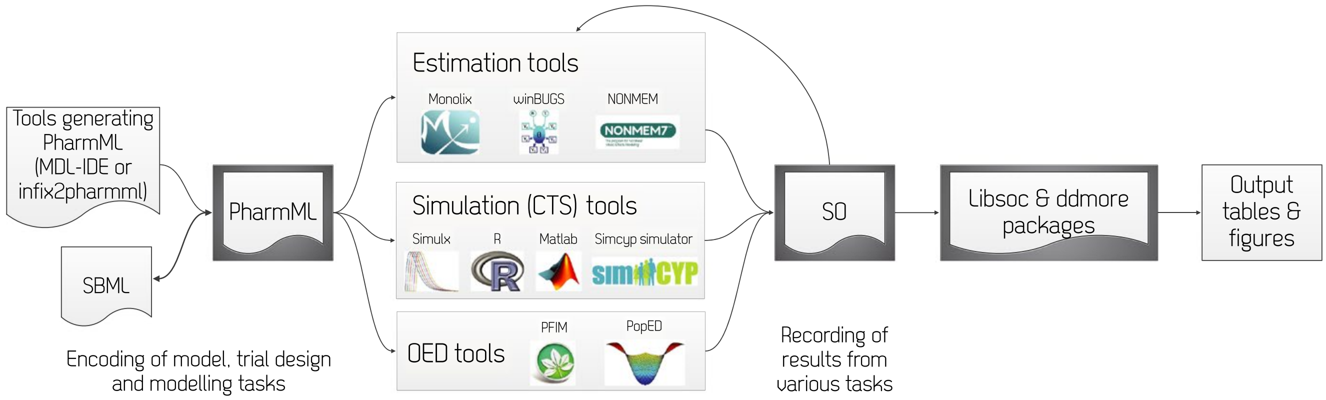

Figure 2 A schematic view of a typical workflow in pharmacometric analyses supported by Drug Disease Model Resource (DDMoRe) standard formats. By encoding or generating (e.g., through the Model Description Language Integrated Development Environment (MDL-IDE)[7,10]) the relevant input information in Pharmacometrics Markup Language (PharmML), the considered modeling task (estimation, clinical trial simulation (CTS), and optimization) can be executed in any of the shown tools thanks to the developed converters. Results are then recorded in Standard Output (SO) files, which can be used in R to generate output tables and figures as well as update modeling information for subsequent tasks by using available libraries. OED, Optimal Experimental Design; SBML, Systems Biology Markup Language.

reviewing, and reporting. As the other DDMoRe standards, the SO definition is open-source and further development and improvements are expected. Specifically, suggestions from the M&S community to the SO definition are encouraged to enhance collaborative drug and disease modeling based on an efficient exchange and reuse of knowledge between all stakeholders (regulators, academics, and pharmaceutical industries) involved in drug discovery and development.

SUPPORTING INFORMATION

Supplementary information accompanies this paper on the *CPT: Pharmacometrics & Systems Pharmacology* website. (www.psp-journal.com)

Appendix S1*. Standard output user guide version 0.3.1.*
Appendix S2*. Tree plot of the Standard Output structure.*
Appendix S3*. Model code of examples.*

Acknowledgments. The authors wish to thank the input of many colleagues across the DDMoRe project who have contributed to discussion of and providing training for SO.

Funding. This study has received support from the Innovative Medicines Initiative Joint Undertaking under grant agreement number 115156, resources of which are composed of financial contributions 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.

Conflict of Interest. As an Associate Editor for CPT: Pharmacometrics & Systems Pharmacology, France Mentré was not involved in the review or decision process for this article.

1. Beal, S.L., Sheiner, L.B., Boeckmann, A.J. & Bauer, R.J. *NONMEM User's Guides, Technical report* (Icon Development Solutions, Ellicott City, MD, 2009).
2. Monolix, Lixoft, Antony, France and Inria, Orsay, France. <http://www.lixoft..com/>.
3. Lunn, D.J., Thomas, A., Best, N. & Spiegelhalter, D. WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Stat. Comput.* **10**, 325–337 (2000).
4. Keizer, R., Karlsson, M. & Hooker, A. Modeling and simulation workbench for NONMEM: tutorial on Pirana, PsN, and Xpose. *CPT Pharmacometrics Syst. Pharmacol.* **2**, 1–9 (2013).
5. EFPIA MID3 Workgroup *et al.* Good practices in model-informed drug discovery and development: practice, application, and documentation. *CPT Pharmacometrics Syst. Pharmacol.* **5**, 93–122 (2016).
6. Harnisch, L., Matthews, I., Chard, J. & Karlsson, M.O. Drug and disease model resources: a consortium to create standards and tools to enhance model-based drug development. *CPT Pharmacometrics Syst. Pharmacol.* **2**, 1–3 (2013).
7. Smith, M.K. *et al.* Model Description Language (MDL): a standard for modeling and simulation. *CPT Pharmacometrics Syst. Pharmacol.* **6**, 647–650 (2017).
8. Swat, M.J., Moodie, S., Wimalaratne, S.M. Pharmacometrics Markup Language (PharmML): opening new perspectives for model exchange in drug development. *CPT Pharmacometrics Syst. Pharmacol.* **4**, 316–319 (2015).
9. Swat, M.J., Grenon, P. & Wimalaratne, S. ProbOnto – ontology and knowledge base of probability distributions. *Bioinformatics* **32**, 2719 (2016).
10. Larizza, C. *et al.* Complex Bayesian modeling workflows encoding and execution made easy with a novel WinBUGS Plugin of the drug disease model resources interoperability framework. *CPT Pharmacometrics Syst. Pharmacol.* **7**, 298–308 (2018).