Citation: CPT Pharmacometrics Syst. Pharmacol. (2017) 6, 285–292; doi:10.1002/psp4.12171

WHITE PAPER

Thoughtflow: Standards and Tools for Provenance Capture and Workflow Definition to Support Model-Informed Drug Discovery and Development

JJ Wilkins[1], PLS Chan[2], J Chard[3], G Smith[4], MK Smith[2], M Beer[5], A Dunn[3], C Flandorfer[5], C Franklin[6], R Gomeni[7], L Harnisch[2], R Kaye[3], S Moodie[8], ML Sardu[9], E Wang[10], E Watson[11], K Wolstencroft[12], SYA Cheung[13]* on behalf of the DDMoRe Consortium

Pharmacometric analyses are complex and multifactorial. It is essential to check, track, and document the vast amounts of data and metadata that are generated during these analyses (and the relationships between them) in order to comply with regulations, support quality control, auditing, and reporting. It is, however, challenging, tedious, error-prone, and time-consuming, and diverts pharmacometricians from the more useful business of doing science. Automating this process would save time, reduce transcriptional errors, support the retention and transfer of knowledge, encourage good practice, and help ensure that pharmacometric analyses appropriately impact decisions. The ability to document, communicate, and reconstruct a complete pharmacometric analysis using an open standard would have considerable benefits. In this article, the Innovative Medicines Initiative (IMI) Drug Disease Model Resources (DDMoRe) consortium proposes a set of standards to facilitate the capture, storage, and reporting of knowledge (including assumptions and decisions) in the context of model-informed drug discovery and development (MID3), as well as to support reproducibility: "Thoughtflow." A prototype software implementation is provided.

CPT Pharmacometrics Syst. Pharmacol. (2017) **6**, 285–292; doi:10.1002/psp4.12171; published online 20 April 2017.

MOTIVATION

Pharmacometrics (PMx) is a quantitative discipline integrating pharmacokinetics (PK), pharmacodynamics (PD), pharmacology, physiology, and statistics to describe and predict drug disposition and effect in individuals and populations. PMx models are used to support drug development and usage by evaluating and simulating drug exposure and response, by explaining variability in drug response and disposition in a study populations, selecting drug doses, optimizing study designs, and describing disease progression. Model building typically spans multiple phases of development and multiple cycles of implementation.[1] The typical PMx analysis process can be represented by a simple workflow tracking the logical sequence of activities and decisions necessary for implementing a model-informed approach.

The underlying structure of and relationships between each stage in a PMx analysis, the assumptions associated with these, and the transitions between stages (supported by decisions) can be highly complex. **Figure 1** provides a detailed illustration of a typical PMx workflow.[2] The PMx analysis consists of a sequence of tasks requiring the integration of different tools and analysis methods—for example, preparing/cleaning/merging datasets from various sources (which might have different formats, and originate from different databases), performing exploratory analyses, making assumptions based on those, preparing for estimation (further data manipulation, setting initial estimates, defining task execution information), estimating parameters in the model, examining model diagnostics, performing model validation, testing assumptions made, and collating all information to make inferences and inform decisions. Each of these tasks has inputs (data, models, parameter values, task properties) which produce outputs (estimation output, graphs, tables, text summary files, etc.) and a description of the sequence of events (such as run logs) and dependencies between tasks: the outputs of one task often provide the inputs for another. There is also an overarching workflow that describes the path from initial model to final model, capturing which "branches" of the development "tree" are fruitful in describing the observed data and which are not. The term "workflow" here speaks as much to knowledge management as it does to a sequence of tasks and dependencies.

The challenge any analyst faces is to be able to track and report these tasks and their interrelationships, as well as the cognitive process behind them, coherently—a

[1]Occams, Amstelveen, The Netherlands; [2]Pharmacometrics, Global Clinical Pharmacology, Pfizer, Sandwich, UK; [3]Mango Solutions, Chippenham, Wiltshire, UK; [4]Scientific Computing Group, Cyprotex Discovery Limited, Macclesfield, Crewe, UK; [5]scinteco, Vienna, Austria; [6]GSK, Clinical Pharmacology Modelling & Simulation, Stockley Park, UK; [7]PharmacoMetrica, La Fouillade, France; [8]Eight Pillars Ltd, Edinburgh, UK; [9]Merck Institute for Pharmacometrics, Merck Serono S.A., Switzerland; [10]Global PK/PD and Pharmacometrics, Eli Lilly and Company, Indianapolis, Indiana, USA; [11]Predictive Compound Safety & ADME, Drug Safety & Metabolism, Innovative Medicines, AstraZeneca, Gothenburg, Sweden; [12]Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands; [13]Quantitative Clinical Pharmacology, Early Clinical Development, Innovative Medicine, AstraZeneca, Cambridge, UK. *Correspondence to: SYA Cheung (Amy.Cheung@astrazeneca.com)

Received 20 September 2016; accepted 4 January 2017; published online on 20 April 2017. doi:10.1002/psp4.12171

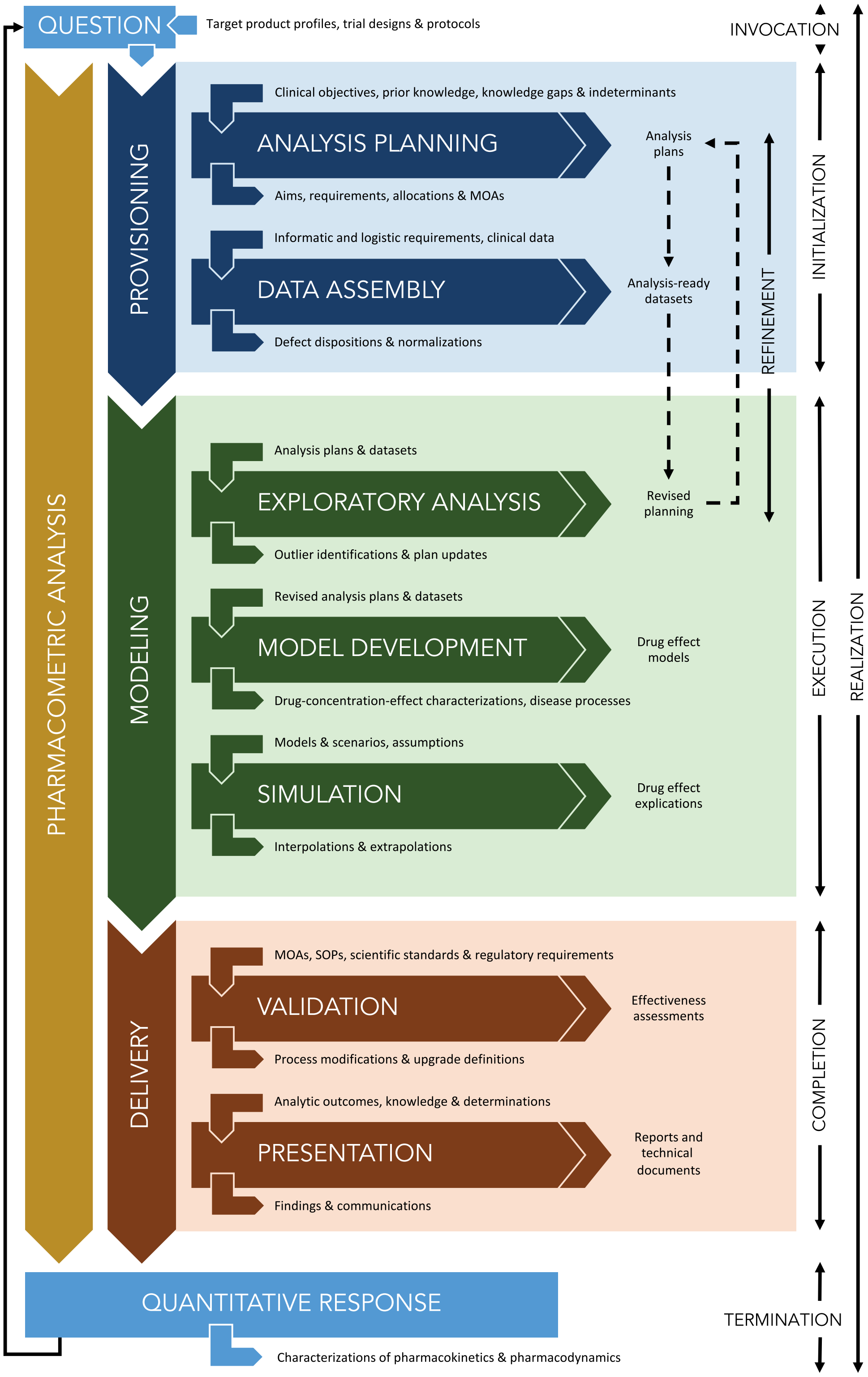

QUESTION
Target product profiles, trial designs & protocols
INVOCATION
PHARMACOMETRIC ANALYSIS
PROVISIONING
Clinical objectives, prior knowledge, knowledge gaps & indeterminants
ANALYSIS PLANNING
Analysis plans
Aims, requirements, allocations & MOAs
Informatic and logistic requirements, clinical data
DATA ASSEMBLY
Analysis-ready datasets
Defect dispositions & normalizations
INITIALIZATION
REFINEMENT
MODELING
Analysis plans & datasets
EXPLORATORY ANALYSIS
Revised planning
Outlier identifications & plan updates
Revised analysis plans & datasets
MODEL DEVELOPMENT
Drug effect models
Drug-concentration-effect characterizations, disease processes
Models & scenarios, assumptions
SIMULATION
Drug effect explications
Interpolations & extrapolations
EXECUTION
REALIZATION
DELIVERY
MOAs, SOPs, scientific standards & regulatory requirements
VALIDATION
Effectiveness assessments
Process modifications & upgrade definitions
Analytic outcomes, knowledge & determinations
PRESENTATION
Reports and technical documents
Findings & communications
COMPLETION
QUANTITATIVE RESPONSE
Characterizations of pharmacokinetics & pharmacodynamics
TERMINATION

Figure 1 Elements of the process of pharmacometric analysis. Based on figure 36.3 from Grasela *et al.*[2] MOA, measures of acceptability; SOP, standard operating procedure.

process for which we have coined the term "Thoughtflow." Capturing and summarizing these workflow components is often performed manually postanalysis; a difficult, tedious, time-consuming, task fraught with opportunities to introduce errors. This is made even more complex by the variation in processes and support infrastructure across different institutions.

Capturing the provenance of task outputs ("how was this output created?") as well as providing knowledge management for the PMx workflow ("how did we get to this model?") facilitates reproducibility, "housekeeping," sharing, and communicating results with others. It also increases transparency and assists the analyst in constructing a more robust and objective modeling "story." The establishment of firm links between source data, data transformations and manipulations, final model and simulation code, and conclusions in documentation is crucial in order to facilitate traceability, enabling a third party (such as an independent or regulatory agency reviewer) to reproduce a complete analysis. It is essential to ensure data integrity, maximize quality assurance and reproducibility in order to enhance the contribution of model-based methods within the regulatory submission and review cycle.[3]

In addition, capturing and reporting provenance information is important in supporting compliance with regulatory expectations of a computer system producing material that may form part of a regulatory interaction (FDA regulation 21 CFR Part 11).[4] A system that can perform this function will support these regulatory requirements, and will encourage transparency of the rationales and details of the work for communication both internally and externally with regulators and other collaborators.

The Drug Disease Model Resources (DDMoRe) consortium was created to improve the quality, efficiency, and cost-effectiveness of model-informed drug discovery and development, a set of concepts, principles, and recommendations that have evolved into what is now termed MID3.[3] Among its other activities,[5–7] it has also developed a framework for capturing, organizing, and reporting this kind of provenance information (M.K. Smith, unpublished data). The concepts and standards we introduce in this article provide a solid basis for developing standardized software implementations that can address this unmet need.

State of the art for workflow management tools

There are a range of tools available in the marketplace, both commercial and open source, for supporting workflows of executable, computational tasks and tracking the provenance of workflow executions. Taverna,[8] KNIME,[9] Kepler,[10] Activiti (http://activiti.org), Navigator (Mango Solutions, Chippenham, UK), Improve (scinteco, Vienna, Austria), and Pipeline Pilot (BIOVIA, San Diego, CA) are some examples. While these tools are all competent at handling workflows, noncomputational tasks, such as making inferences and decisions based on analytical results, are equally important when tracking and reporting PMx analyses—direct comparisons to Thoughtflow are, perhaps, unfair, since the applications and domains are quite different. It was clear after our assessment of the available tools that a platform able to capture computational tasks, business processes, and the interactions between them was required.

BASIC PRINCIPLES

A provenance-based approach

The Thoughtflow approach is centered on the recording and exploration of provenance: information about entities, activities, and agents involved in producing a piece of data—or any other kind of entity—and the relationships between them.

We use the PROV-O ontology (http://www.w3.org/TR/prov-o/) as the basis of our approach to defining and capturing provenance information. PROV-O is a World Wide Web Consortium (W3C) standard that captures the provenance and relationships between items, as well as an audit trail information (who did what, when, and how) required of the Thoughtflow environment. The use of an existing, well-established standard means that a set of tools for visualizing relationships are already available, and tools developed by DDMoRe and its successors are likely to be easily supportable. Building on an established standard also brings with it a large body of experience in other applications besides PMx.

Briefly, the PROV-O ontology defines a series of terms relating to *entities*, *activities,* and *agents* (**Figure 2**). An entity is a physical, digital, conceptual, or other kind of thing with some fixed aspects, which may be real or imaginary (examples include a model, a dataset, an output file, a script, a decision, or an assumption). An activity is something that occurs over a period of time and acts upon or with entities, and may encompass consuming, processing, transforming, modifying, relocating, using, or generating entities (such as estimating the parameters of a model, or performing a visual predictive check). Finally, an agent is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity. An agent can be an organization, a person, or a software application.

PROV-O defines a set of relationships that describe the interactions between entities, activities, and agents. Entities may be derived from other entities (a model based on another model), may be generated by activities (a model output may be generated by an estimation step), and are attributed to agents (such as NONMEM). Activities may use entities (a model fit may use a data file), may be informed by other activities (a previous model development step), and are associated with agents (NONMEM, or the analyst, or both). Agents may act on behalf of other agents (NONMEM may be used by the analyst, and the analyst may be working on behalf of an organization). PROV-O also supplies the necessary framework to invalidate entities and/or activities, allowing the effects of making a change in an analysis to be assessed based on the relationships of the affected entity or activity with downstream activities and entities: everything dependent on the change could, for example, be flagged as invalid, or as requiring reassessment. This has obvious benefits during scientific and quality review.

In line with the principles of good practice (GxP), descriptions, rationales, as well as comments for each activity should be provided in order to identify the key analysis steps (base and final models, for example), to highlight model assumptions, and to document rationales and justifications at decision points. Recording, storing, and querying this information has distinct benefits for accurately capturing and describing how a model was developed, evaluated, and ultimately applied to address a drug development question.

The application of the PROV-O standard to support the PMx use case necessitated some focus in how entities, activities, and agents from the PROV-O model were

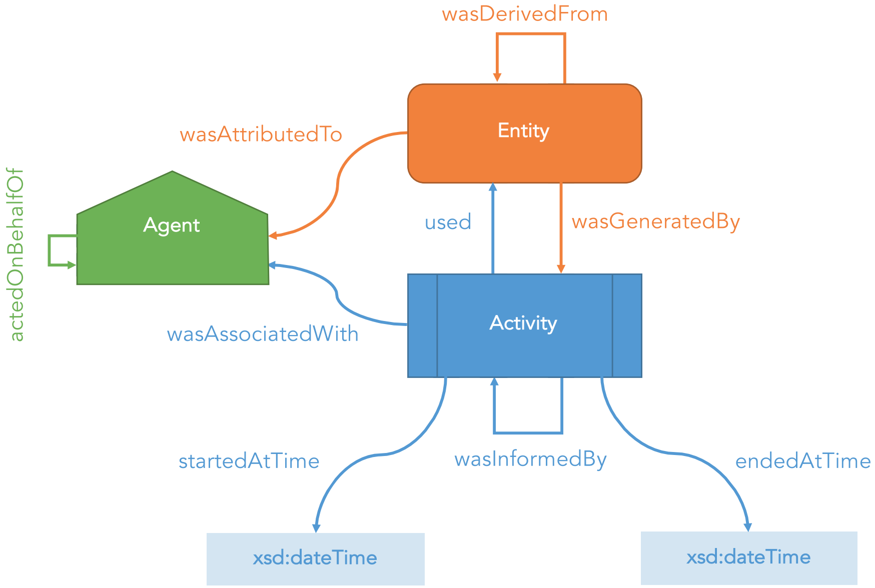


Figure 2 Entities, activities, agents, and relationships in PROV-O at a high level. See **Supplemental Material S1** for definitions of PROV-O terms *endedAtTime*, *startedAtTime*, *used*, *wasAssociatedWith*, *wasAttributedTo*, *wasDerivedFrom*, *wasGeneratedBy*, *wasInformedBy, actedOnBehalfOf.* xsd, XML Schema Definition.

described and annotated. There are a number of metadata items, such as flags or states, that are important to be able to be attached to entities, such as the importance of an output ("pivotal" or not), its quality control (QC) review status ("pass" or "fail," or null if not assessed), and its significance in the analysis (such as flagging base and final models as such). Other pieces of metadata related to entities might include file names and sizes, dates and times of file creation or most recent modification, details related to the analyst who created or most recently modified an entity, as well as version and location control. Metadata for an activity might include the name of the analysis tool(s) and their versions, compiler information (if relevant), the command line used for execution, the hardware and platform on which it was executed, etc. Metadata for an agent might include their role in the analysis and within the PMx team.

Transparency in the setting and evaluation of assumptions that may impact model application is of great importance in the planning and documentation of any model-informed activity.[3] It was necessary to create a structured definition for assumptions to enable them to be stored, associated with other entities or activities, to be peer-reviewed, and to be reported. Should an assumption change, an analyst could easily track the components of the model development process that depend on that assumption, and reevaluate the relevant steps. In addition, decisions (model selection, based on outputs from assumption testing, etc.) are crucial to document. Decisions were defined as entities that physically take the form of structured text files containing 1–2 sentences describing the decision made.

The Thoughtflow specification provides a framework for querying provenance data to extract, report, and visualize provenance data associated with an analysis, allowing the creation of run records, QC and management summaries, audit trails, and analysis flow diagrams.

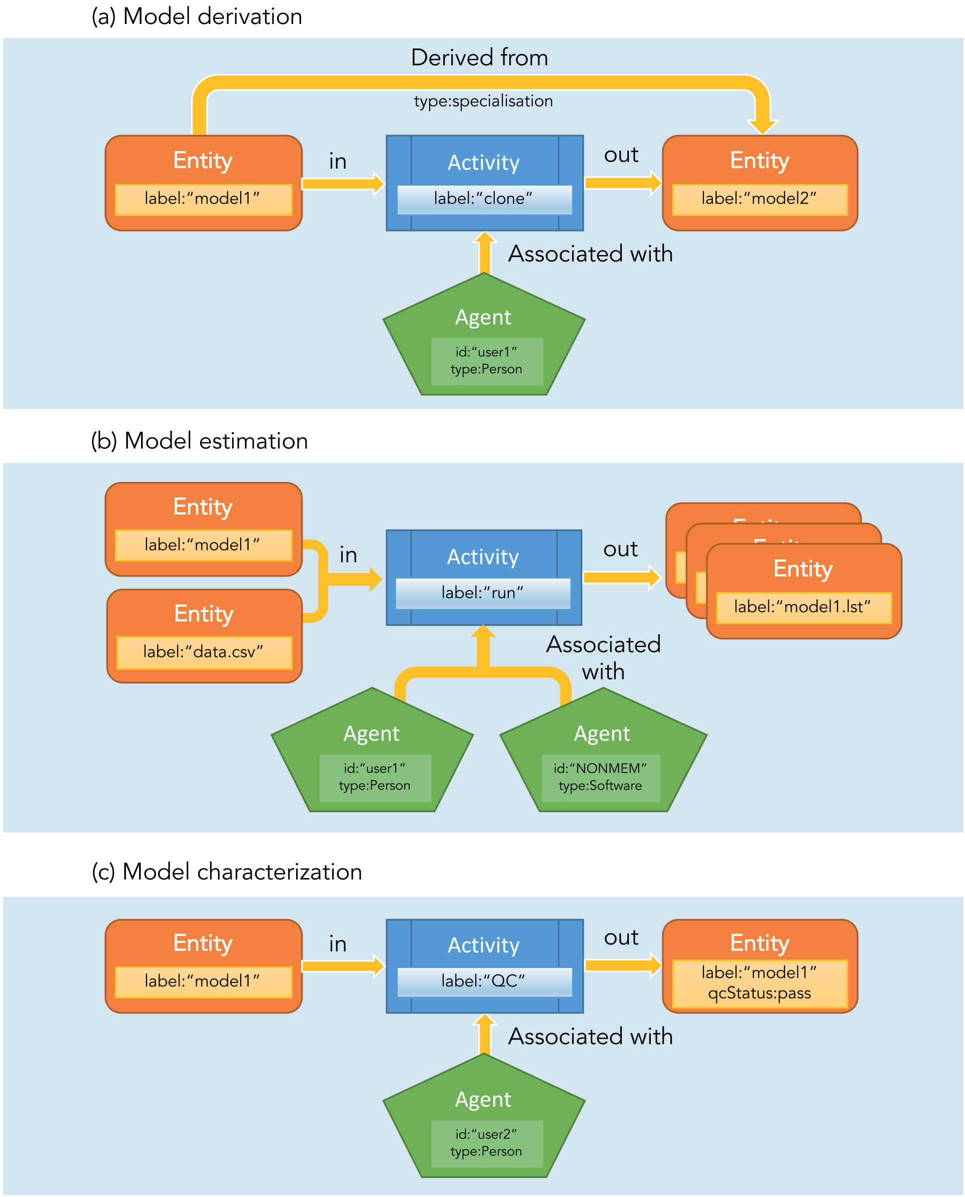
Figure 3 shows how these relationships fit together for different tasks and activities.

IMPACT

The benefits of the proposed Thoughtflow system are manifold. Filtered, interactive views and queries of the data according to user role (analyst, manager, reviewer) and appropriate export functionality, in combination with tagging of entities and activities by analysts to identify key model development steps, would enable the large amounts of information typically associated with a PMx analysis to be reduced to a manageable level, and support the streamlining of tracking and reporting activities.

Analysts would be able to apply provenance information to rapidly generate documentation for analyses, from run records to complete reports, in a reproducible and structured manner. A well-structured "Thoughtflow" capture and reporting tool would enhance knowledge sharing and communication among team members: an individual analyst would be able to rapidly familiarize themselves with an analysis started by a colleague, or on projects where several analysts are collaborating, for example. This avoids the repetition or duplication of prior work and helps assure quality and reproducibility by ensuring the use of the same versions of analysis datasets and software.

Using a Thoughtflow-based tool, it would be more straightforward for analysts to repeat an analysis either completely or in part, if required (if source data changed mid-analysis, for example). Using tagging and appropriate queries, a Thoughtflow tool could also facilitate the rapid generation of complete reports (through LaTeX (https://www.latex-project.org) and support tools such as knitr[11]) and summary outputs, such as run logs, model development summaries that include



(a) Model derivation

(b) Model estimation

(c) Model characterization

Figure 3 Examples of a provenance approach to pharmacometrics workflow. In (**a**), entity "model2" is derived from entity "model1" via activity "clone." Agent "user1" was associated with the activity. In (**b**), the parameters of entity "model1" are estimated using entity "data.csv" during activity "run," generating a range of new entities including "model1.lst." Two agents were associated with this activity: the analyst "user1" and the software "NONMEM." In (**c**), entity "model1" is assessed by quality control reviewer "user2" during activity "QC" to generate a new version of entity "model1," with the flag qcStatus set to "pass."

the source (provenance), version and location of key input and output files for QC and review purposes, high-level project summaries for management, and complete audit trails, as well as complete reports. This has the added benefit of avoiding error-prone hand-generated tables.

Objective and detailed summaries of the model development process are possible using this approach, including descriptions of the assumptions being made as well as the objectives and rationales for individual tasks. This allows reviewers and regulators to assess precisely what activities have been completed and evaluate the appropriateness of the results and conclusions, potentially reducing delays associated with missing or incomplete information. Additionally, the analysis could be reproduced either completely or in part if necessary, using a minimal number of steps (keystrokes or mouse clicks). Standardization of the way the provenance data are stored, imported, and exported between applications would facilitate straightforward sharing of databases between analysts and organizations.

Finally, managers would be able follow the process of a data analysis project at whatever level of detail wished. Since progress would be updated in real time with respect to the analysis, this could potentially reduce the time spent supervising an analyst. It would also be easier to track the

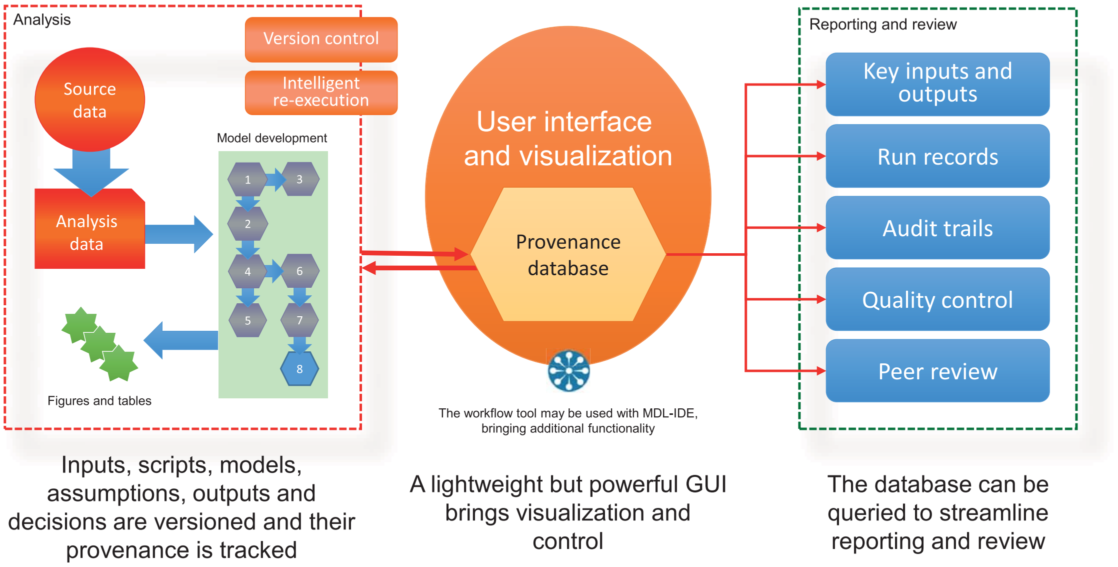

Analysis
Version control
Intelligent re-execution
Source data
Model development
Analysis data
1
3
2
4
6
5
7
8
Figures and tables
User interface and visualization
Provenance database
The workflow tool may be used with MDL-IDE, bringing additional functionality
Reporting and review
Key inputs and outputs
Run records
Audit trails
Quality control
Peer review
Inputs, scripts, models, assumptions, outputs and decisions are versioned and their provenance is tracked
A lightweight but powerful GUI brings visualization and control
The database can be queried to streamline reporting and review

Figure 4 Scope of the proposed Thoughtflow approach. GUI, graphical user interface.

progress of an analysis with respect to project timelines, and could thus allow for better resource allocation to meet timelines.

SPECIFICATIONS AND PROTOTYPE

The proposed DDMoRe Thoughtflow specifications describe cross-platform, robust support for tracking, reporting, and replicating elements of a modeling and simulation pharmacokinetics, pharmacodynamics (PKPD) project (across all phases of development), either in part or as an entire analysis.

A technical specification for provenance capture and suggestions for implementation has been developed (included as electronic **Supplementary Material**) which describes the input-to-output task workflow, including decisions made, as well as the PMx knowledge management suitable for describing a complete, end-to-end modeling and simulation exercise.

Any Thoughtflow solution, in order to be generalizable, must facilitate the flexible integration of varying operational processes and standards in industry, small and medium enterprise, and academia, as well as diverse computational tools for data handling, visualization, modeling, simulation, and reporting. Commonly-used tools for data handling and visualization include R (R Foundation, Vienna, Austria), MatLab (MathWorks, Natick, MA), SAS (SAS Institute, Cary, NC), and a myriad of others. Tools used for modeling and simulation include nonlinear mixed-effects tools such as NONMEM (ICON, Dublin, Ireland), R, SAS, Phoenix NLME (Certara, Princeton, NJ), and Monolix (Lixoft, Antony, France), as well as Bayesian tools such as WinBUGS (MRC Biostatistics Unit, Cambridge Institute of Public Health, Cambridge, UK), OpenBUGS,[12] and Simcyp (Certara). Reports are typically authored using literate programming tools, such as Rmarkdown (http://rmarkdown.rstudio.com) and LaTeX/knitr, or Microsoft Word (Microsoft, Redmond, WA). New tools such as Stan[13] and Julia[14] may come into more common use in the future, hence it will be essential to allow their integration as well. Owing to the sheer number and diversity of options available, it is not possible to explicitly support every tool that can be used for PMx data analyses. The DDMoRe strategy focuses instead on an application programming interface (API) for capturing provenance information, with prebuilt hooks for a limited number of widely used tools, such as NONMEM, R, and Monolix. It is nonetheless possible, in principle, to support any tool via the API.

Ultimately, an implementation of the Thoughtflow standards should facilitate the comprehensive reexecution of analyses, either completely or in part, based on changes to key elements in the analysis (such as source data). A tool should be able to identify dependencies of the changed object(s), and will allow these to be selectively updated without the need to completely rerun the entire analysis.

Figure 4 provides a high-level overview of what is proposed, and **Figure 5** provides a simple example of how the components of an analysis are linked with one another.

A software prototype demonstration of how the Thoughtflow API might be implemented has been developed. Although not suitable for production use, and lacking some features such as the ability to reexecute an analysis, it is cross-platform (supporting all commonly used operating systems) and portable: it has been designed to provide a basis for further development, with a view to a future version being able to be deployed as painlessly as possible in a range of likely industry, academic, and regulatory scenarios. It is comprised of several components: the provenance infrastructure, provenance services, a provenance database, a version control system, the graphical user interface (GUI), and satellite tools allowing reporting and visualization based on queries of the database.

Provenance infrastructure

The provenance infrastructure monitors user actions and automates actions such as adding files to the repository. The infrastructure creates suitable messages and sends them to

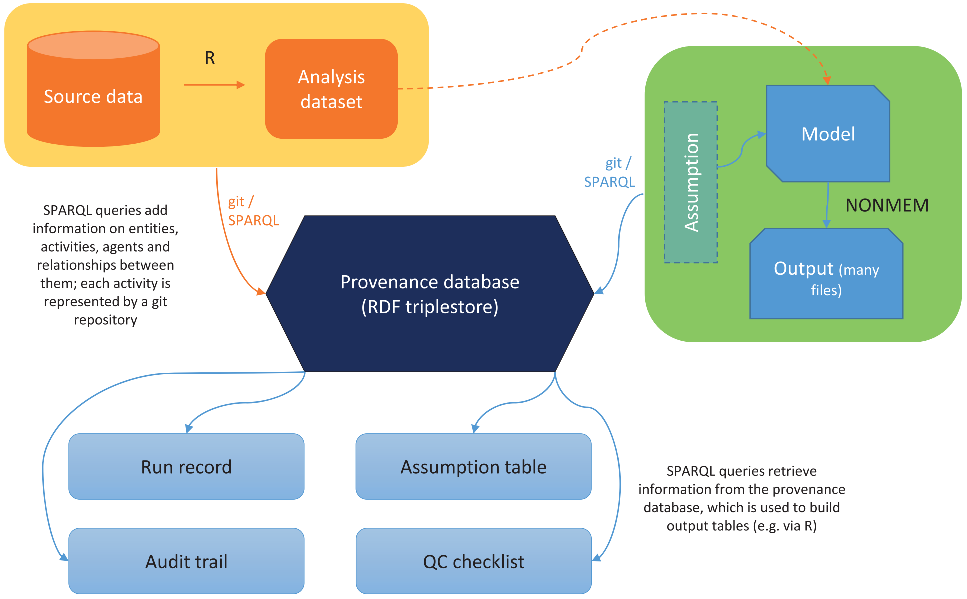

Source data
R
Analysis dataset
git / SPARQL
SPARQL queries add information on entities, activities, agents and relationships between them; each activity is represented by a git repository
Provenance database (RDF triplestore)
git / SPARQL
Assumption
Model
NONMEM
Output (many files)
Run record
Assumption table
Audit trail
QC checklist
SPARQL queries retrieve information from the provenance database, which is used to build output tables (e.g. via R)

Figure 5 A simple example workflow, showing how various components are linked with one another. QC, quality control; RDF, Resource Description Framework; SPARQL, SPARQL Protocol and RDF Query Language.

the provenance service, initiating actions such as storing Thoughtflow information in the provenance database.

Provenance database

In contrast to the traditional relational database approach, a Resource Description Framework (RDF) triplestore, which is a "graph" database (a database that uses graph structures with nodes, edges, and properties to represent and store data) designed for the storage and retrieval of "triples" using semantic queries via the RDF format lies at the heart of the system. A "triple" is a data construct composed of a subject, a predicate, and an object, and can be visualized as two nodes linked by a relationship: for example, "dataset1.csv was created by createData.R," or "run5.mod was associated with assumption1." Thoughtflow information is captured as a series of interconnected nodes, joined together by these relationships, which capture all the decisions, activities, inputs, and outputs in a single graph. The graph can be traversed using semantic queries, matching on nodes across the graph, using the RDF format. Queries can return multiple downstream nodes (for example, finding outputs that are affected by a change to an upstream input), and support the use of ontologies—a relationship or node can have subtypes (for instance, both a graph and a table can be an output of an activity, and can be handled separately, enabling one to "find graphs" or "find outputs").

Provenance information will be encoded in the triplestore using the PROV-O standard. The triplestore is queried via standard SPARQL queries via a SPARQL server (Apache Fuseki in our implementation). The prototype uses Apache Jena (https://jena.apache.org), a free and open source Java framework for building semantic web and linked data applications to support the interface with the triplestore. The prototype was designed with the option to use the commercial RDF server Virtuoso (OpenLink, Burlington, MA) in production if so required.

Provenance services

A suite of services will allow a client to query the content of the provenance database using a simple representational state transfer (REST) web interface, insulating them from the underlying implementation and negating the need to craft their own queries. A request from the client is then implemented using SPARQL (a semantic query language for databases: SPARQL Protocol and RDF Query Language, https://www.w3.org/TR/rdf-sparql-query). The services will support queries to add, update, and retrieve information—for instance, to determine the provenance of an entity, to retrieve the information to create a project run record, or to retrieve a list of invalidated entities and activities following an update to an upstream entity or activity.

Version control system

It is essential to unambiguously and reproducibly identify all entities within a project. In order to support this, a dedicated version control system is necessary. This function is performed in the prototype by Git (https://git-scm.com), a free, open source, and ubiquitous distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Graphical user interface

A browser-based GUI has been developed to allow interactive visualization of projects based on Thoughtflow provenance data, and to facilitate simple use of the reporting tools.

Software for reporting and visualization

R can be used to process extracted provenance information to produce output such as comma separated value (CSV) data files, publication-ready tables (run records, QC summaries, audit trails, as rich text or LaTeX), and visualizations (flow charts and trees at varying levels of detail), and scripts for reexecution of the analysis or parts thereof.

RECOMMENDATIONS AND ANTICIPATED OUTCOMES

The concepts and standards we have described here, along with the technical specification and demonstrator tool we have developed, illustrate DDMoRe's proposed Thoughtflow infrastructure. Although the software implementation that has been developed is a technology demonstration rather than a production-ready tool, the source code is open and available under the Affero General Public License, v. 3 (AGPLv3), which allows for its use by third parties provided derivative works are released under a compatible open source license. Implementations are already underway in commercial offerings such as Navigator and Improve. We anticipate that the work already done will provide a solid foundation for further development.

The potential for this technology is considerable. We hope the PROV-O-based standards and general system architecture we have developed will be adopted more widely in the future, as a platform for documenting, reproducing, and seamlessly sharing analyses between analysts, teams, companies, regulators, and other stakeholders in a nonproprietary way.

Acknowledgments. The authors thank Wendy Aartsen, Marylore Chenel, Marc-Antoine Fabre, Mats O. Karlsson, Enrique Larios Vargas, Peter Milligan, Céline Sarr, Nadia Terranova, and Eunice Soek Mun Yuen for their support of and input into this work. Work performed by Justin Wilkins was funded by Pfizer, Eli Lilly & Co., and Uppsala University. Work performed by Michael Beer and Christian Flandorfer was funded by Leiden University. Work performed by Roberto Gomeni was funded by AstraZeneca. The research leading to these results has received support from the Innovative Medicines Initiative (IMI) Joint Undertaking under grant agreement no. 115156, resources of which are composed of financial contributions from the European Union's Seventh Framework Programme (FP7/2007-2013) and European Federation of Pharmaceutical Industries and Association's companies in kind contribution. The DDMoRe project is also financially supported by contributions from academia and small and medium enterprise partners (SME).

Conflict of Interest. All authors are employees or contractors of stated companies and represent these companies as part of their participation in the IMI DDMoRe project.

Author Contributions. All authors were involved in the writing and review of the manuscript.

1. Bergsma, T.T. *et al.* Facilitating pharmacometric workflow with the metrumrg package for R. *Comput. Methods Programs Biomed.* **109**, 77–85 (2013).
2. Grasela, T. & Dement, C.W. Engineering a pharmacometrics enterprise. In *Pharmacometrics: The Science of Quantitative Pharmacology* (eds. Ette, E.I. & Williams, P.J.) 903–924 (John Wiley & Sons, Hoboken, NJ, 2007).
3. Marshall, S.F. *et al.* Good practices in model-informed drug discovery and development: practice, application, and documentation. *CPT Pharmacometrics Syst. Pharmacol.* **5**, 93–122 (2016).
4. U.S. Food and Drug Administration. CFR — Code of Federal Regulations Title 21. <http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=1>. Accessed September 2016.
5. 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**, e34–35 (2013).
6. Ribba, B. *et al.* A review of mixed-effects models of tumor growth and effects of anticancer drug treatment used in population analysis. *CPT Pharmacometrics Syst Pharmacol.* **3**, e113 (2014).
7. Swat, M.J. *et al.* Pharmacometrics Markup Language (PharmML): opening new perspectives for model exchange in drug development. *CPT Pharmacometrics Syst. Pharmacol.* **4**, 316–319 (2015).
8. Wolstencroft, K. *et al.* The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res.* **41**, W557–561 (2013).
9. Berthold, M.R. *et al.* KNIME: The Konstanz Information Miner. In *Data Analysis, Machine Learning and Applications* 319–326 (Springer, Berlin, Heidelberg, 2008). Studies in Classification, Data Analysis, and Knowledge Organization.
10. Ludäscher, B. *et al.* Scientific workflow management and the Kepler system. *Concurr. Comput. Pract. Exper.* **18**, 1039–1065 (2006).
11. Xie, Y. knitr: A comprehensive tool for reproducible research in R. In *Implementing Reproducible Computational Research* (eds. Stodden, V., Leisch, F. & Peng, R.D.) (Chapman and Hall/CRC, New York, 2014).
12. Lunn, D., Spiegelhalter, D., Thomas, A. & Best, N. The BUGS project: evolution, critique and future directions. *Stat. Med.* **28**, 3049–3067 (2009).
13. Carpenter, B. *et al.* Stan: a probabilistic programming language. *J. Stat. Softw*; e-pub ahead of print 2016.
14. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V.B. Julia: a fresh approach to numerical computing. <http://arxiv.org/abs/1411.1607> (2014).

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