# Clinical Decision Support Framework for Validation of Multiscale Models and Personalization of Treatment in Oncology

Anca Bucur, Jasper van Leeuwen, Traian Cristian Cirstea, and Norbert Graf

*Abstract*—**The implementation of Clinical Decision Support (CDS) solutions is an important prerequisite for reducing the knowledge gap between clinical research and practice, especially in a complex genetic disease such as cancer. However, current CDS solutions are unable to support all the complex decisions required for personalized treatment of cancer patients and become quickly obsolete due to the high rate of change in therapeutic options and knowledge. Our CDS framework enables the development of decision support tools that flexibly integrate a large variety of multiscale models and can leverage the efforts of a large community of modellers. In our implementation, we combine community-developed models described in the literature (e.g. the St. Gallen stratification for early breast cancer) and models derived by mining the comprehensive datasets from clinical trials and care brought together in the p_Medicine collaborative research project. This framework and its underlying solution for models storage, management and execution will also constitute a platform for continuous validation of existing models on new data. Our goal is to enable the reuse of existing models for CDS and for the development of new models, and to support collaboration among modellers, CDS implementers, biomedical researchers and clinicians. We initially develop and deploy our solution in the context of the p-Medicine project in the oncology domain, but we aim to expand our scope and to reach out to a wide community of users in the biomedical area.**

## I. INTRODUCTION

P-MEDICINE (www.p-medicine.eu) [1,2] brings together leaders in their fields to create an infrastructure that supports the transition from current practice towards personalized medicine. The project aims to help bridge the current knowledge gap between clinical research and practice [3,4]. p-Medicine will enable efficient secure sharing of comprehensive data sets used for the development of demanding Virtual Physiological Human (VPH) multiscale tools and models. In this context, the p-Medicine clinical decision support framework aims to leverage the large coherent datasets from clinical trials available in p-Medicine and the VPH models developed in the project to efficiently bring this new knowledge to the bedside.

This project/study is partially funded by the European Commission under the 7th Framework Programme (FP7-ICT-2009-6- 270089).

Anca Bucur, PhD, Cristian Cirstea and Jasper van Leeuwen are with Phillips Research, Healthcare Information Management, High Tech Campus 34, 5656 AE Eindhoven, The Netherlands (e-mail: anca.bucur@phillips.com, cristian.cirstea@philips.com & jasper.van.leeuwen@phillips.com). Norbert Graf, Prof. Dr. Med., is with the Department of Pediatric Oncology and Hematology, Saarland University, Germany (e-mail: Norbert.Graf@uniklinikum-saarland.de)

A CDS system assists clinical users in making clinical decisions relevant for a patient case. Current CDS solutions are faced with significant challenges [3].

Knowledge and information access and maintenance are key for the effectiveness and success of CDS: Medical knowledge, especially in a complex genetic disease such as oncology, is changing and growing at an unprecedented rate. New evidence is found and gradually brought into clinical care, new medication and treatments are introduced, the evidence-based guidelines are evolving and expanding. In this context keeping the CDS tools and their recommendations up to date is a major task. In addition to the maintenance of external knowledge and access to the latest evidence, the ability to efficiently update the CDS tools to take this new evidence into account is of utmost importance. A flexible and scalable CDS solution that is easy to update with the inclusion of new clinical evidence, for instance in the form of clinical models supporting specific decisions can effectively address these challenges.

In this paper we introduce our models-based CDS framework and the underlying solution for storage, management and execution of multiscale models. Our approach to CDS that relies on clinical knowledge encapsulated in (VPH) clinical models has in our view many advantages compared to building monolithic and closed CDS solutions: It is modular, scalable, can be efficiently customized and updated, and can benefit of the modelling efforts of a large community.

Two important sources of models are considered in order to support future CDS applications. One is based on the clinical knowledge provided in the literature (evidence generated by the wide community), while another source is based on data mining the large repository of clinical research and care data built within the p_Medicine context. We have currently implemented both types of models: In [6] we have reported on the development and integration into the CDS framework of predictive models developed by mining a large clinical trial dataset available in the p-Medicine data environment. We have demonstrated that the solution enables efficient collaboration among clinical researchers, knowledge modellers, data miners and CDS implementers. An example of literature-based models implemented in our framework is the St. Gallen stratification for early breast cancer [7].

In Section II.A we describe and provide a motivation for this approach to CDS implementation, in Section II.B we present the general architecture of our solution, while in Sections II.C and II.D we describe the implementation of services for two execution engines: Jess [8] and an engine enabling the composition of individual models into complex models of models.

## II. The Clinical Decision Support Framework

### A. *Benefits of a flexible and scalable approach to CDS implementation*

To deliver flexibility and scalability, we propose a solution in which the CDS recommendation applications are loosely-coupled through well-defined interfaces with the models management infrastructure. This facilitates the easy integration of new clinical models and of updates and extensions of existing models. As a result, our approach to CDS can deal well with the high rate with which medical knowledge changes and can also lead to higher acceptance of both the CDS tool and of the underlying clinical models by the clinical users. Another advantage of this solution relates to the ability to independently validate the clinical models and gradually add them to the CDS tool and apply them in patient care. Standard interfaces to the Model Repository enable the integration of the models by a variety of CDS applications enabling customization and the selection of suitable models to closely address the needs and the context of each healthcare organization.

We also strive for flexibility in the definition of the models management infrastructure by aiming to support a range of execution engines and of corresponding languages widely used by model developers. Our preliminary selection includes Jess and Drools [9] (both rule based), R [10] and Java/Groovy [11]. In this paper we describe the integration of the Jess engine and our initial solution to building composite models, i.e. models of models (hyper-models).

Workflow integration is another key success factor for the CDS systems. To facilitate adoption of the proposed models, we make sure that the CDS application encapsulating the clinical models fits in the current clinical workflow and can be efficiently used in the care setting. This ensures that all extensions and updates with new models can be easily integrated in the current way of working of the clinicians.

### B. *Architecture overview*

Figure 1 depicts a simplified architecture of our CDS framework. The models management approach is at the core of this solution: Next to the Model Repository we provide services enabling users to inspect the available models and execution engines, to run models on their datasets, and to add, remove and update models via an administration service. Models can be represented in a variety of ways, as modules in one of the supported languages (currently Jess or Java). A CDS client can query the available models and execute them making use of the applicable engine. Deployed on the p-Medicine platform, this solution will also enable the validation of models on new datasets and their remote update, without being disruptive for the CDS clinical end-user. Models description using semantic annotations based on widely-adopted ontologies will facilitate the use of our framework by model developers and tool builders outside the p-Medicine user community.

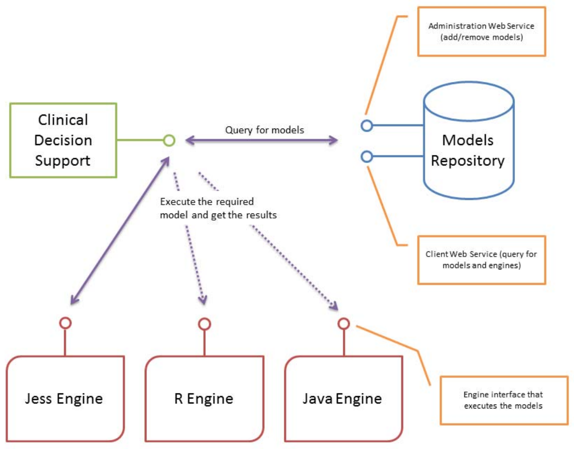
Administration Web Service (add/remove models)
Clinical Decision Support
Query for models
Models Repository
Execute the required model and get the results
Client Web Service (query for models and engines)
Jess Engine
R Engine
Java Engine
Engine interface that executes the models

**Figure 1** Architecture overview of the models-based CDS framework

The CDS framework consists of three main sub-systems:

- A front-end CDS application including the user interface and services focused on the user experience and on the visualization of data and on providing CDS recommendations,
- The CDS models framework - the backend application consisting of one or more databases to store the clinical models and their annotations, and a range of services focused on the management of the models.
- A set of services (one or more workers) that execute the models on the input data and generate the output that will be used by the front-end layer to generate the recommendations. These make use of the execution engines integrated into the environment.

This architecture makes the system easily scalable with respect to models execution and representation. Each model in our system is described by a set of input elements and types, a set of output elements and their types, a model file that can be executed by one of the available engines, and annotation that sufficiently describes the model. If required, more engines can be easily added to support the execution of models represented in different languages. The decision support framework allows therefore scientists to implement their clinical models in various programming languages that suit their needs.

For instance, for the St. Gallen stratification model each of the four input parameters has a type and is linked to a query to the patient database to retrieve the corresponding elements (the query can be modified to match the individual source), the model is represented as a Jess script and the output is the patient stratification yielded by the execution of the script by the Jess engine.

As data is stored by different hospitals by making use of different standards (or proprietary solutions), modifications and data schemas, the access to data is not standardized. Therefore, feeding the right data to the model is not a trivial process. The decision support system addresses this difficult situation by separating the clinical model from the actual location of the data. It acts as an abstraction layer between the two, and the actual binding between the input variable of the model and the actual database fields of the patient data source is performed automatically by making use of an external mapping solution.

### *C. The Jess Engine*

The Jess engine is integrated into the platform because it is powerful and flexible enough to cope with most of the requirements of clinical models. Jess is a rule based engine and scripting environment that uses an enhanced version of the Rete algorithm to process rules. Jess comes with its own scripting language which is an extension of CLIPS (C Language Integrated Production System) language.

The Rete algorithm [12] is an efficient pattern matching algorithm. It uses a rooted acyclic directed graph, where the nodes, except the root, represent patterns, and paths from the root to the leaves represent left-hand sides of rules. Each node stores information about the facts satisfied by the patterns of the nodes in the paths from the root up to and including this node. This information is a relation representing the possible values of the variables occurring in the patterns in the path.

Jess applies a set of if-then statements (rules) to a dataset (facts). Figure 2 describes how the Jess engine works. The *rules* represent the knowledge, whereas the *facts* are the evidences on which Jess reasons, based on defined knowledge. When a fact (or a set of facts) matches the one or more rules, then the rules are executed, therefore the working memory is updated (possibly new facts are added, removed or changed). A notifying message stating that changes occurred in the Working Memory (facts) is broadcasted to the Production Memory which automatically triggers all the rules to be re-evaluated (including the one that triggered the evaluation).

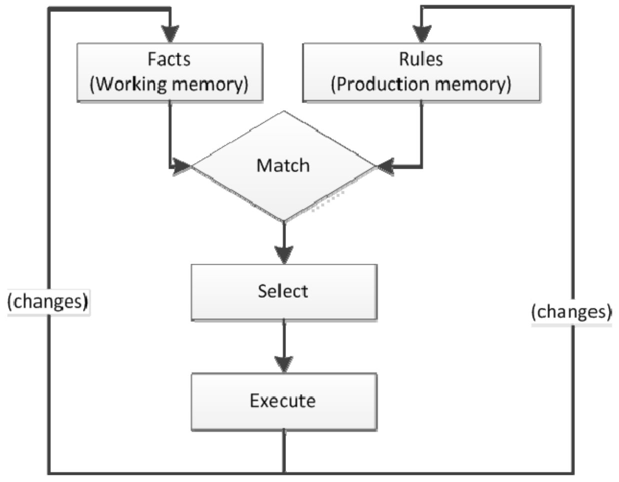

**Figure 2** Jess engine: How it works

### *D. Engine for building composite models*

A Composite Models Engine is also provided by the platform. This is a generic engine allowing for the execution of a model subsuming other models residing in the Model Repository. It allows for models to be connected together in order to create a composite model (i.e. a model of models). The purpose of providing this engine is to support re-use and collaboration by enabling users (modellers) to build complex models in a modular and scalable way by combining and extending the available models.

The Composite Models Engine takes as one of the inputs a specification containing both an enumeration of the models to be used and a specification of how the models are interconnected. The Composite Model Engine executes a subsumed model when all required inputs are available for that subsumed model. Note that subsumed models can be composite models themselves. For convenience, the Composite Models Engine also allows to set a value as input for an input channel of a subsumed model. The output of a model can be connected to multiple input channels.

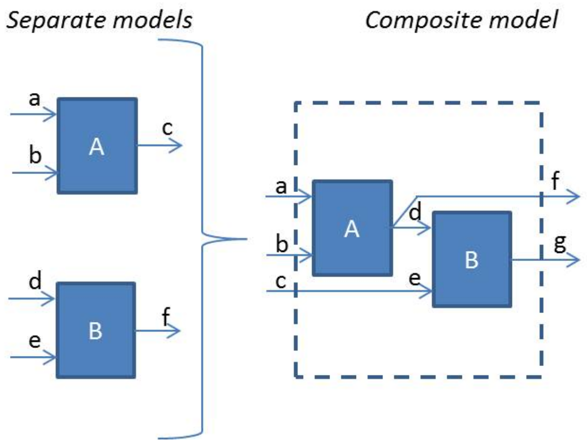

**Figure 3** Building composite models

The input to the Composite Models Engine is an XML file that has the following format:

```
<composedmodel>
    <model id="1" name="…" version="…"/>
    <model id="2" name="…" version=”…”/>
    <dataflow from="input" to="1"/>
            <dataflow from="1" to="output"/>
    <dataflow from="input" to="2">
        <map-parameter from-parameter="param1" to-
parameter="param2"/>
    </dataflow>
    <dataflow from="2" to="output">
        <map-parameter from-parameter="output" to-
parameter="return1"/>
        <map-parameter value="1" to-parameter="return2"/>
    </dataflow>
</composedmodel>
```

In the above template, **<composedmodel>** is the root element. **<model id="" name="" version="">** is the declaration of the model that is going to be used in the composed model. In the declaration, name is the name of the model in the Models Repository and version contains the version of the model according to the Models Repository. The model will be further referred by its identifier.

The declaration **<dataflow from="" to=""/>** depicts the binding of two models by their identifiers. If not present, the **from** attribute is considered input and the **to** attribute is considered output. If no **<map-parameter>** tag is included in the **<dataflow>** tag, all the output parameters of the **from** model are passed as input parameters to the **to** model.

The declaration **<dataflow from="input" …>** enables binding of input parameters of the composed model, while **<dataflow … to="output">** expresses binding to the output parameters of the composed model.

Finally, the statement **<map-parameter from-parameter/value="" to-parameter=""/>** maps a parameter from the output parameters of the **from** model to an input parameter of the **to** model.

## III. Conclusion

We propose a flexible, models-based CDS framework that enables decision support applications to cope with the high rate of growth of medical knowledge. The framework facilitates easy integration of new models and their continuous validation with new datasets by making use of the comprehensive clinical trials datasets available on the p-Medicine platform. This approach may also lead to higher acceptance of the CDS tool and of the underlying clinical models by the clinical users.

The uniform interfaces that we provide towards the front-end CDS applications will enable the use of our models framework by a wide range of CDS tools, making the knowledge incorporated in the models available to a wide community of CDS developers and clinical users. We believe that his open approach will encourage models developers to contribute new models to our repository which in turn will enable tools to support complex decisions and keep up with the updates in available knowledge, and will also facilitate the adoption of the solution by clinicians.

Important areas of extension of our solution refer to (1) enabling the use of a wider range of execution engines such as R, Groovy and Drools, (2) defining comprehensive annotations of models to enable their use by researchers outside the p-Medicine user community, and (3) developing a suitable semantic solution mapping the metadata (e.g. annotations) describing the models to the patient datasets (in the care setting) to which the models need to be applied.

## References

[1] Graf et al., "p-medicine from data sharing and integration via vph tools to personalized medicine," in Proc. VPH2012, London, 2012.

[2] S. Rossi et al., "p-Medicine: From data sharing and integration via VPH models to personalized medicine". Ecancermedicalscience. 2011; 5: 218. Published online 2011 August 17. doi: 10.3332/ecancer.2011.

[3] D.F. Sittig et al., "Grand challenges in clinical decision support". J Biomed Inform 2008: 41 (2): 387-92.

[4] M.T. Scheuner et al., "Delivery of Genomic Medicine for Common Chronic Adult Diseases". JAMA, 2008 299(11):1320-1334.

[5] S. Straus et al. (Eds.), "Knowledge Translation in Health Care: Moving from Evidence to Practice". John Wiley & Sons, 28 mei 2013.

[6] Stefan Rüping et al., "Improving the Implementation of Clinical Decision Support Systems". Proc. of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'13), July 3-7, Osaka, Japan.

[7] A. Goldhirsch et al., "Strategies for subtypes—dealing with the diversity of breast cancer: highlights of the St Gallen International Expert Consensus on the Primary Therapy of Early Breast Cancer 2011", Annals of Oncology 22: 1736–1747, 2011, Published online 27 June 2011.

[8] Jess, http://www.jessrules.com.

[9] Drools, http://www.jboss.org/drools/.

[10] R, http://www.r-project.org/.

[11] Groovy, http://groovy.codehaus.org/.

[12] Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," Art. Intelligence, vol. 19, pp. 17–37, 1982.