Proceedings Paper:

Tartarini, D., Viceconti, M., Gruel, N. et al. (2 more authors) (2014) The VPH Hypermodelling framework for cancer multiscale models in the clinical practice. In: In Silico Oncology and Cancer Investigation (IARWISOCI), 2014 6th International Advanced Research Workshop on. Proceedings of the 2014 6th International Advanced Research Workshop on In Silico Oncology and Cancer Investigation - The CHIC Project Workshop (IARWISOCI) , 03-04 Nov 2014, Athens. Institute of Electrical and Electronics Engineers .

https://doi.org/10.1109/IARWISOCI.2014.7034642

The VPH Hypermodelling Framework for cancer multiscale models in the clinical practice*

D. Tartarini, K. Duan, N. Gruel, D. Testi, D. Walker, M. Viceconti

Abstract— **The VPH Hypermodelling framework is a collaborative computational platform providing a complete Problem Solving Environment to execute, on distributed computational architectures, sophisticated predictive models involving patient medical data or specialized repositories. In the CHIC[1] project, it will be enhanced to support clinicians in providing prompt personalised cancer treatments. It supports several computational architectures with strict security policies.**

I. INTRODUCTION

Cancer is a complex disease that behaves differently from patient to patient. It can affect several organs of the human body requiring specific treatments. It challenges clinicians in making reliable diagnosis, prognosis and choosing the tailored treatment. The oncological research is further complicated since cancer manifestation spans several spatio-temporal scales, from molecular and cellular level to organ level, from nanosecond molecular reactions to tumour evolution in years [1]. Researchers have proposed numerous mathematical models describing cancer progression and treatments from a biochemical and biophysical perspective [1],[2],[3]. Generally, due to computational resource limitations and the complexity of bridging scales, these models are focused on a single phenomenon at a particular space-time scale. It is evident that a synergic cross-discipline collaboration among researchers (clinicians, biomedical scientists/engineers and computer scientists) would improve the chance of success. In fact, merging knowledge and expertise of researchers would allow the development of more accurate integrative cancer models that, with the help of computer simulations, can support the clinicians in their decisions [2]. This is the philosophy underpinning the Virtual Physiological Human (VPH)[4] agenda. In alignment with this, we aim to provide a hypermodelling environment where experts can contribute with their knowledge to develop sophisticated integrative models.

A. The VPH Hypermodelling framework

The VPH Hypermodelling Framework (VPH-HF) is a collaborative computational platform providing a complete Problem Solving Environment (PSE)[5] to execute, on distributed computational architectures, sophisticated predictive models involving patient medical data or specialized repositories. It is based on a fully-fledged prototype developed in a previous VPH project, the Osteoporotic VPH (VPH-OP)[2], which addressed the estimation of bone fracture risk due to osteoporosis. The aim of VPH-HF is to improve the effectiveness of diagnosis, prognosis and treatment of specific diseases in clinical practice with the ultimate objective to foster the personalized medicine paradigm and perform *in silico* clinical trials [6].

In particular, in the frame of the EC project Computational Horizons in Cancer (CHIC)[1] the VPH-HF is customized for the oncological needs targeting two primary users: the clinician and the researcher. The former needs easy out-of-the-box software tools to analyse patient medical data and simulate cancer behaviour to address specific clinical questions. The latter has a broader profile that includes the creation and validation of complex integrative models [7]. Researchers can populate the CHIC data and model repositories with their experimental data and provide the integrative/predictive models implemented in the computational format of their convenience. Furthermore, in CHIC a semantic annotation system is provided to support the development of *ad hoc* ontologies and folksonomies.

The VPH-HF framework is very flexible: almost any model can be integrated and run in the PSE, whether it is an executable binary file, an interpreted script (e.g. written in Python, Perl, Matlab, Octave) or a more complex case involving external licensed software like Ansys® or Abaqus®. The underlying assumption is that an hypermodel (i.e. an integrative model or a composition/orchestration of models) can be described as a workflow where its composing hypomodels (i.e. models) are connected to produce an output result from a given input and data from repositories and/or patient specific data. Therefore a workflow can be represented as a graph where the nodes are models or data repositories while connections are data or control flows. Two models are connected when an output of the first is an input of the second, while data repositories can be connected to any of the models. In order to build a workflow, hypermodels and hypomodels can be considered as black boxes with a standardized abstract interface exposing input and output ports and control data flow (Figure 1). This

---

*Research supported by the “Computational Horizons in Cancer” project, funded by the European Union, EC Seventh Programme for research, technological development and demonstration under grant agreement No [600841].

D. Tartarini, D. Walker, K. Duan, N. Gruel, M. Viceconti are with the INSIGNEO Institute for in silico Medicine, Sheffield, S13JD UK (corresponding author to provide phone: +44 (0) 114 222 6173; e-mail: d.tartarini@sheffield.ac.uk).

D. Testi is with Consorzio inter-universitario CINECA, Bologna, Italy

[1] http:// http://chic-vph.eu/

[2] http://www.vphop.eu

interface is well defined within the CHIC project and is called *Component Model Generic Stub*. It ensures the interoperability between all the provided hyper- and hypo-models.

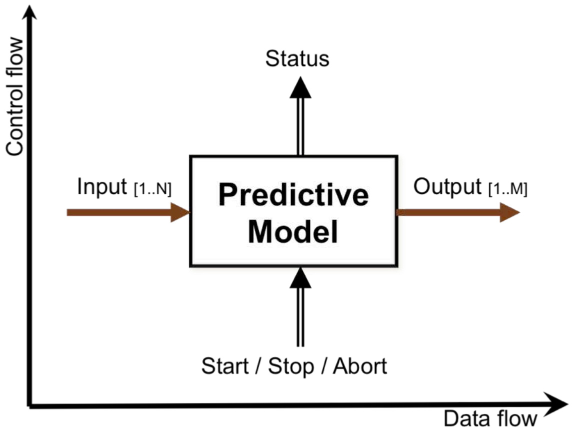

Figure 1 Predictive model as black box

The VPH-HF is compliant with this interface and provides a software implementation that follows the Wrapper pattern [8]. It allows the actual integration of any of the computational instances of the models in a workflow including data and control flow in an appropriate standard format. The *model wrapper* (Figure 2) provides the following functionalities: adapts the parameters from the format used in the actual model to the standard one of the Component Model Generic Stub interface, retrieves input from the storage services, produces execution logs, runs the model and stores the results in the Storage service. In order to ensure modularity the wrapper exposes an XML-RPC/SOAP interface that is used by the Taverna Server [9] to run the model.

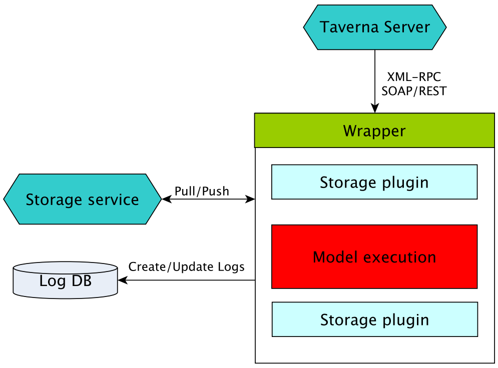

Figure 2 The model wrapper for the Generic Stub interface

B. *The VPH-HF architecture*

The architecture of VPH-HF is inspired by the concept of modularity: each component can be used in isolation or ensemble with others to offer more sophisticated functionalities. This approach ensured an effective extension of the VPH-HF prototype developed in the VPH-OP project to the new requirements and scenarios of the CHIC project. The whole back-end VPH-HF architecture is hidden to the users since the target goal is to allow the adoption in the clinical practice through an easy and secure interface. In particular, as shown in (Figure 3), the clinician interacts only with the *CHIC portal* that embeds the key functionalities: uploading and (pseudo)anonymisation of patient data, visualization of processed patient data and images (CT, MRI etc), semantic searching for data and models using metadata information, configuring and running existing workflows using the *Hypermodelling Editor*, retrieval and validation of results. On the other side, VPH-HF also provides applications with an interface conceived for an expert user to fully control the execution of a workflow in the framework (i.e. the Hypermonitor).

In summary, the user interacts with the whole system through the following tools:

1. *The CHIC Portal*: a Liferay[3] based portal offering tools to anonymise patient data, to upload/download data, search model and data through their metadata, execute workflows.
2. *The CHIC Hypermodelling Editor*: is a Java portlet allowing the user to compose hypomodels into hypermodels and execute new or pre-built workflows on patients data or data repositories.
3. *The Hypermonitor*: a VPH-HF stand-alone application enabled to launch existing workflows on specific data, monitoring the execution status, and visualize the execution logs.
4. *The VPH-HF portlet*: provides the Hypermonitor functionalities into a web portal.

In particular the first two tools are developed by partners of the CHIC project and integrated in VPH-HF, while the last two are applications developed within the VPH-HF software stack. Nevertheless the VPH-HF adopted the security and authentication mechanisms to fully integrate with the CHIC requirements and services. In fact the whole computational platform has to obey to strict security policies given the involvement of patient data.

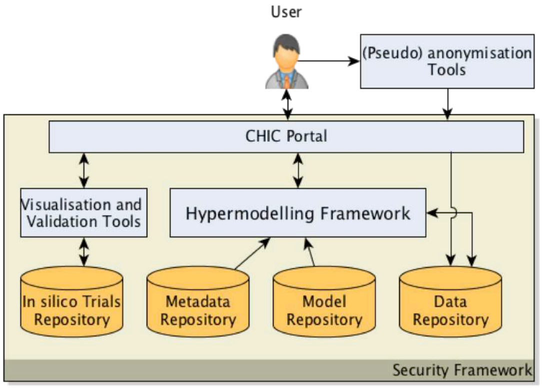

Figure 3 User perspective of the CHIC architecture

The VPH-HF architecture is designed to be flexible, modular, easy to maintain and be customized to mutable needs of the users, as well as sophisticated computational

---

[3] http://www.liferay.com/

architectures. All the software components (Figure 3) expose a standard interface (e.g. XML-RPC, SOAP, REST) that potentially allows them to be used in isolation. In the most common scenario the authenticated user prepares a workflow (using the available hypermodels or the CHIC workflows), submits it and waits for results. The end-user applications interact with the *Workflow Manager* that interprets their instructions and, through the *Communication bus (i.e. MAF3 Event Bus),* initialises the necessary services, transfers data, sets up the execution environment, runs the workflow on the computational infrastructure and retrieves the results. A key role is played by the *Workflow Orchestrator* (an instance of the *Taverna Server* [9]*),* which actually orchestrates the execution of all the models composing the workflow and the data flow between them. Each model exposes a standardized interface through the *model wrapper* that takes care of adapting the input/output to the particular model interface. The framework can address more sophisticated scenarios, thanks to the functionalities offered by its components (Figure 4):

- *Workflow Manager or Director*: orchestrates the entire process from the authentication of the user, the setup of the execution environment, the control of execution of the workflow, storage and retrieval of the results.
- *Communication bus*: provides the communication services between the software components of the framework even in a distributed computational architecture.
- *Authentication service*: manages the user authentication, permissions, role and a single sign on service provided by partners in the CHIC consortium.
- *Workflow Orchestrator*: it is based on the Taverna server that actually interprets the workflow and coordinates the model execution and data transfer.
- *Registry service*: provides a registry for the services available on the framework and the models that can be executed with their respective status.
- *Storage services*: provide the storage for the workflow inputs and results through a REST interface.
- *Log management*: stores the execution log of all the models composing a workflow.
- *Probabilistic engine*: allows users to create probabilistic variables when required in the workflow execution.
- *Wrapper*: implements a standardised interface to interact with every model instance, executes models, performs the push/pull operations from the data storage services and creates execution logs
- *Mechanical Turk*: an application programming interface, implemented in the *Director*, that integrates functionalities that require human intervention or manual operations during the execution of hypermodels (e.g. request for the user to validate data from partial workflow execution).
- *Transformation services:* allow the data from one format to be transformed to another according to the model specifications.

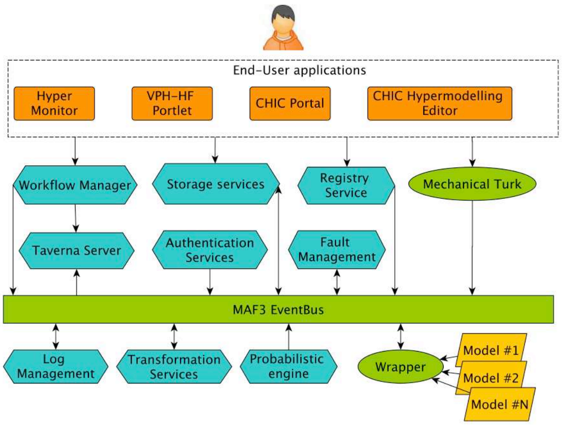

Figure 4 The VPH-HF architecture has a modular design where components (in hexagonal shapes) communicate each other through the MAF3 Event Bus. User can design, execute and monitor workflow execution through Web or stand alone applications (shown in rectangular boxes).

C. *The computational scenarios*

The VPH-HF architecture is designed to support several computational architectures since the candidate scenarios are diverse. In fact, some models are computationally intensive, may require high bandwidth, or specific hardware or software with some machine specific license. In particular, we identified and support the following scenarios:

- Parallel systems: multiprocessor systems and clusters for computationally intensive models or models implemented through parallel computing libraries.
- Distributed systems: systems including remote data repositories and storage services, High Performance Computing facilities, architectures with specialised hardware as attached accelerators (GPU, Xeon Phi), systems with specific licensed software.
- Private Cloud infrastructure.

In the CHIC project the target architecture adopted is a private cloud based on OpenStack [10]. The VPH-HF has been deployed on this platform in a configuration where all the VPH-HF services, the CHIC repositories and storage services are installed in the same system. This solution enforces the security requirements and the execution performance especially for those models requiring substantial bandwidth and a high degree of coupling.

D. *The CHIC clinical use case*

In the field of computational cancer research, one typical workflow could, for example, address the growth of a generic tumour embedded in a particular tissue. It can combine models describing the tumour growth, nutrient and oxygen diffusion, vascularization, and actions of any chemotherapeutic agents. Specifically the CHIC project is addressing different cancer types, according to the expertise of the groups involved in the project. These include glioblastoma, nephroblastoma, lung, prostate and colon cancer. Clinicians are offered visualisation tools (e.g. Dr Eye[4] [11]) to analyse patient data and the hypermodelling framework to predict cancer evolution and the impact of different treatments. The aim of the project is to identify a personalised treatment in a short time to effectively help the clinician to minimise the patient suffering while awaiting treatment and surgery. Simulations based on specific patient data (e.g. imaging data) can better capture the cancer evolution in time and the best treatment strategy to adopt to increase the chances of success.

REFERENCES

[1] T. S. Deisboeck, Z. Wang, P. Macklin, and V. Cristini, "Multiscale Cancer Modeling," *Annu. Rev. Biomed. Eng.*, vol. 13, no. 1, pp. 127–155, Aug. 2011.

[2] H. M. Byrne, "Dissecting cancer through mathematics: from the cell to the animal model," 2010.

[3] T. S. Deisboeck and G. S. Stamatakos, *Multiscale Cancer Modeling*. CRC Press, 2010.

[4] P. Kohl and M. Viceconti, "The virtual physiological human: computer simulation for integrative biomedicine II," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1921, pp. 2837–2839, Jun. 2010.

[5] "Computer as thinker/doer: problem-solving environments for computational science," *IEEE Computational Science and Engineering*. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=326669. [Accessed: 29-Oct-2014].

[6] G. Clermont, J. Bartels, R. Kumar, G. Constantine, Y. Vodovotz, and C. Chow, "In silico design of clinical trials: A method coming of age," *Critical Care Medicine*, vol. 32, no. 10, pp. 2061–2070, Oct. 2004.

[7] A. Anderson and V. Quaranta, "Integrative mathematical oncology," 2008.

[8] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. 1994.

[9] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidalga, M. P. Balcazar Vargas, S. Sufi, and C. Goble, "The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud.," *Nucleic Acids Res.*, vol. 41, no. Web Server issue, pp. W557–61, Jul. 2013.

[10] O. Sefraoui, M. Aissaoui, and M. Eleuldj, *OpenStack: toward an open-source solution for Cloud Computing Platform Using OpenStack*. International Journal of Computer …, 2012.

[11] E. Skounakis, V. Sakkalis, K. Marias, K. Banitsas, and N. Graf, "DoctorEye: A multifunctional open platform for fast annotation and visualization of tumors in medical images," presented at the 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3759–3762.

---

[4] http://biomodeling.ics.forth.gr/