Original scientific paper

UDK 66.018.9:669.295.5`71`292.018.9
DOI 10.7251/COMEN2202140S

SGABU PLATFORM – INTEGRATED PLATFORM FOR BIOMEDICAL DATASETS AND MULTISCALE MODELS

Tijana Šušteršič*[1,2], Jelena Marković[1], Aleksandar Atanasijević[3], Andreja Živić[4], Miloš Ivanović[2,4], Nenad Filipović[1,2]

[1] University of Kragujevac, Faculty of Engineering, Kragujevac, Serbia
[2] Bioengineering Research and Development Center (BioIRC), Kragujevac, Serbia
[3] University of Kragujevac Computing Center, Kragujevac, Serbia
[4] University of Kragujevac, Faculty of Science, Kragujevac, Serbia

* Corresponding author: tijanas@kg.ac.rs

Abstract: The purpose of the SGABU platform is to include various models and datasets in the area of multiscale modelling. The main aspect of SGABU platform are variousdatasets and multiscale models related to cancer, cardiovascular, bone and tissue disorders. From the point of view of the dataset integration, a task requires implementation of the userinterface that includes manipulation with either tabular data, or most of the datasets requiredfurther tuning carried out by front-end developers employing technologies such as Angular,Plotly.js, Paraview Glance, etc. From the point of view of integration of the multiscale models, most of the simulation modules provided by SGABU platform are implemented as Common Workflow Language (CWL) workflows. This method is an obvious choice since itmakes use of Docker containerization and a standardized way of representing inputs, outputs,and intermediate results, giving findability, accessibility, inter-operability and reusability (FAIR principles). The effort of providing CWL type workflows consists of two distinct actions: (1) developing CWL implementation on FES (Functional Engine Service) backend and (2) developing an appropriate UI. Such integrated platform demonstrates the use of different modelling examples and illustrates the learning process from idea to implementation.

Keywords: multiscale modelling, datasets, integration, platform.

1. INTRODUCTION

Multiscale modeling is a rapidly expanding scientific field that may be used to solve a variety of bioengineering challenges. As a result, education in this field must stay up with the current developments. Cardiovascular, cancer, tissue and bone disorders impose a major social and economic cost on Europe, causing great human suffering and, in many cases, death. Early detection and monitoring of these disorders can lead to successful patient management and therapy [1]. The introduction of high-resolution imaging, computational, and analytical methods into cancer, arteries, tissue, and bone research, as well as the rapid growth of multiscale modeling techniques, have helped to better understand the living systems, extended understanding of complicated phenomena that cannot be studied in clinical environments, and revealed common underlying principles in cancer, cardiovascular, tissue and bone physiologies [1]. Powerful computers and effective numerical tools make it possible to solve complex biological issues in actual geometries and in record speed. In biomedical engineering, multiscale modeling presents a new paradigm for future clinical diagnostics systems. Furthermore, integrative informatics, which covers bioinformatics, imaging informatics, sensor informatics, clinical informatics, and public health informatics, provides a whole new perspective on informatics and computers in healthcare and life sciences. Globally, education and research in this field are of considerable interest [1].

As a result, there have emerged some effective biomedical research platforms. PANBioRA, for example, is a modular platform that standardizes biomaterial assessment and opens the door to pre-implantation, individualized diagnostics for biomaterial-based applications [2]. The SILICOFCM platform is a novel in silico clinical trials solution for the design and functional optimization of whole cardiac performance as well as monitoring the success of pharmacological therapy that is cloud-based [3]. The Bioengineering and Technology (BET) platform facilitates innovation, supports research, and federates the larger multidisciplinary community involved in translational bioengineering [4]. It is largely focused on the requirements of the cancer research community.

However, to the best of our knowledge, there is not a single platform that includes several fields and combines models from different modelling fields. In addition, there are no platforms that are dedicated both to research and education at the same time. Under the project "Increasing scientific, technological and innovation capacity of Serbia as a Widening country in the domain of multiscale modelling and medical informatics in biomedical engineering (SGABU)" [5], the focus is to develop a platform that would include examples from four different fields: cardiovascular, cancer, tissue and bone modeling. The SGABU platform is a platform for the adaptive knowledge discovery of multiscale models and datasets. The primary benefit is the use of fresh, current, and distinct technologies for different levels of architecture while developing a platform. These technologies are fully independent of the hardware, networking, and the operating system, allowing for the platform to be easily migrated from the virtual machine to the public cloud if necessary. There are several benefits for individuals that utilize the platform. Since the platform is executable within a web browser, users may utilize it without installing any new software on their local devices, including mobile devices thanks to its responsive UI design. Although still under development, the platform's ultimate release will have a significant influence not just in research, but also in teaching. The platform will allow universities and faculties to illustrate the usage of various modeling instances, exhibiting the learning process from idea to execution and analysis of the results. The full platform interactivity implies that modifying the input parameters will reveal how these parameters impact the result, allowing the user to test the deduction process as well as perform research in the SGABU project areas.

2. METHODOLOGY

The purpose of the SGABU platform is to interconnect a number of various tools into repeatable data- intensive pipelines of workflows. The integration of the platform is performed according to the standardized software engineering procedures. The architecture of the SGABU platform is presented in Figure 1, with the modules and their corresponding engines and tools. The SGABU framework can be described as a hierarchical multilayer schema comprising of five layers.

At the bottom lies the hardware layer where virtual machines are deployed. Just above lays the security layer which provides mechanisms for user access management, encrypted communication and user authentication within the platform. Next layer corresponds to the workflow tools which providing core engines to be utilized by the upper layers. The workflow layer includes the following engines: *(i)* workflow engine, *(ii)* the Docker container engine, *(iii)* the data quality control engine and *(iv)* 3D visualizer.

Each one of these engines can be invoked directly by SGABU tools and modules that lie in the upper, back-end layer. The last is the front-end layer which provides the user interface. In further few paragraphs, we explain a few structural and implementation details for each of these layers.

Hardware layer. The hypervisor cluster consists of 3 physical servers with a total of 96 CPU cores, 376 GB RAM and 1.89 TB SSD, interconnected with 10 Gbps internal network, separate management network and the gigabit uplink. It runs KVM (Kernel-based Virtual Machine) under ProxmoxVE 7 environment.

Security layer. In SGABU, all outside HTTP traffic is routed by a central NGINX reverse proxy and is SSL encrypted. As mentioned, a separate VPN secured private network was established for developers and accessing the platform's internal infrastructure.

Workflow layer. The most resource-demanding backend component is *FunctionalEngineServer* (FES), responsible for the execution of the CWL (Common Workflow Language) [6] compatible scien-

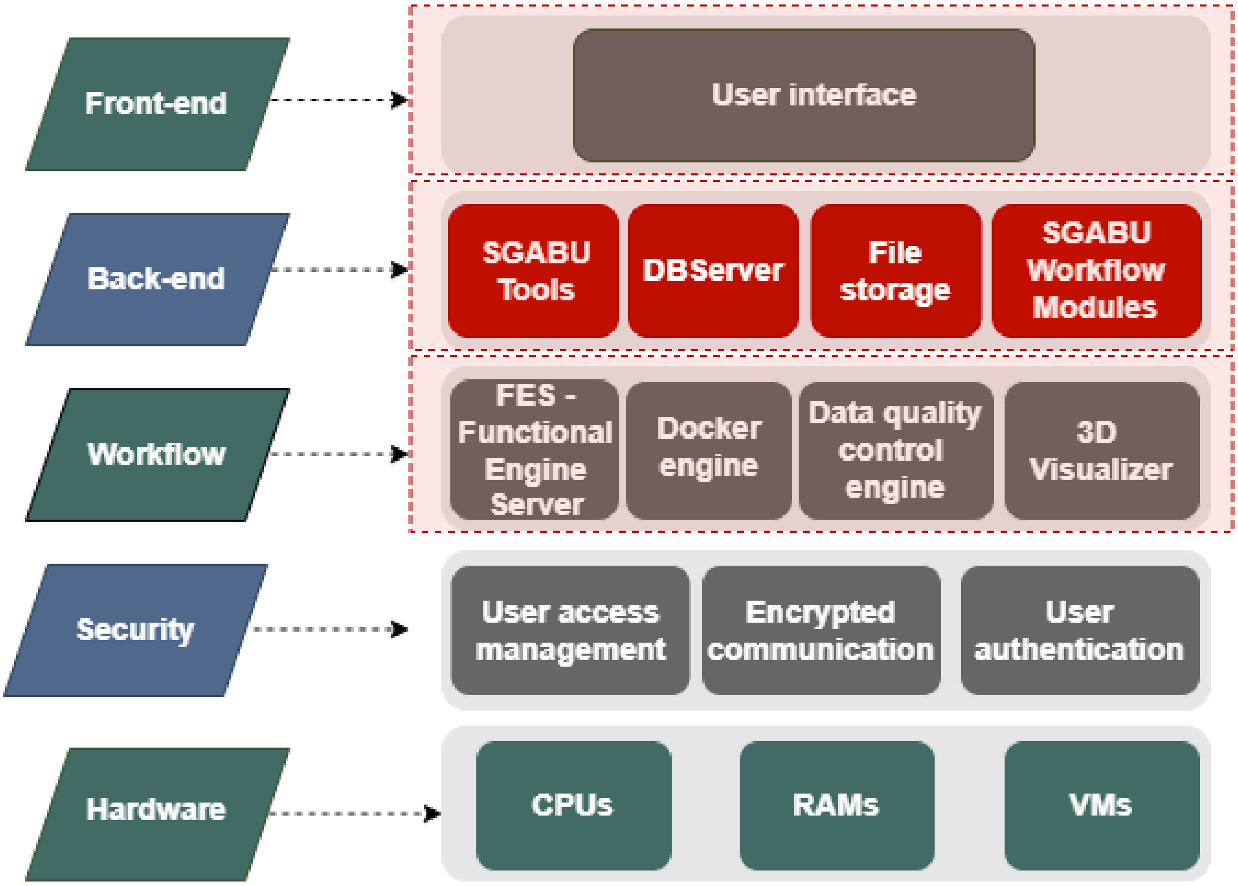


Figure 1. *The architecture of the SGABU platform*

tific workflows. The FES (Functional Engine Server) API is an API developed in Python whose main purpose is to run and manipulate various workflows. FES-API implements the required executor for running workflows. The FES in the background employs Common Workflow Language (CWL) which provides portability of the workflows. It is possible to manage the entire lifecycle of a single workflow using the functional server interface, including creation, handling inputs and outputs. FES is in charge of handling input files, as well as storage where the workflows store their outputs. The significant advantage of this system is an asynchronous execution of workflows as background processes, allowing multiple workflows started by multiple users to run simultaneously.

The customized Paraview Glance [7] plays the role of the 3D visualizer. It is tightly coupled with FES. Its purpose is to perform postprocessing and transformation tasks on 2D/3D simulation data. Thanks to this tight coupling, the user is released from necessity of transporting large workflow output files. The integrated 3D viewer obtains output files directly from FES, avoiding large data transfers, unless explicitly required, i.e., for offline analysis.

Back-end layer. Laravel [8] is a free and open-source PHP framework that provides a set of tools and resources to build modern PHP applications. It is used as a backend framework for SGABU platform. It follows a model-view-controller design pattern, which generally makes it significantly easier to start creating and after that, maintaining the functionality of platform. On SGABU platform it also provides important built-in features like authentication, sessions, routing, migration system etc. *DBServer* provides a standard relational MySQL (MariaDB) database.

Frontend layer. As one of the most popular and supported software development tools available, Angular [9] is used for building single-page client applications using HTML and TypeScript. This framework implements core and optional functionality as a set of TypeScript libraries that are imported into applications. The core value proposition of

Angular is to make it possible to build applications that work for nearly any platform - whether mobile, web, or desktop. SGABU frontend is developed in the responsive manner. Some models or datasets in the SGABU platform will produce outputs intended for visualization with a goal of better understanding.

2.1. Integration of the datasets

From the point of view of the SGABU integration, a single dataset integration task usually starts with interviewing the dataset provider about the way he/she wants the specific dataset to be presented in the platform. The subsequent activity is the implementation of the agreed UI. Then the team returns to the data provider with the requirement for testing and iterative UI tuning until reaching the provider's criteria. For some datasets, the activity is trivial due to tabular data, while most of the datasets required further tuning carried out by front-end developers employing technologies such as Angular, Plotly.js, Paraview Glance, etc. Angular is classified between the most popular software development tools and programmers use this framework for designing dynamic applications. By using different components, programmers are shortening code and in such a way, they reduce the time for changing it. The SGABU platform can be used on any device, by changing the size of device panels of platform and forms. Visualization some data is available by using Plotly.js which is created on base JavaScript. Advantage to the other tools for visualization is many types of graphs for example maps, pie charts, bar charts, bubble charts etc. Many of types are used on the platform SGABU. Plotly.js provides many possibilities like downloading plot as a PNG, zoom, pan etc. [10].

2..2. Integration of the CWL workflows

For practical reasons, but also to promote an open-science approach, all data, analytical tools and methods developed in SGABU project framework should be findable, accessible, interoperable and reusable (FAIR principles). The FAIR principles serve as a guideline for data producers and researchers to be interoperable as much as possible. The individual tools should be organized and interconnected in a standardized way. This automatically implies packaging software using Linux container technologies, such as Docker or Singularity, and then orchestrating workflows and pipelines using domain-specific workflow language such WDL (Workflow Description Language) and CWL (Common Workflow Language).

For the purpose of building SGABU platform backend, we opted for Common Workflow Language [6] as a specification pathway for all our workflows. It employs Docker containers as primary building blocks to provide a straightforward definition of any scientific workflow. In the following sections, we will describe how to build a typical CWL workflow out of the already available software components.

1. Containerization,
2. Creating a CWL tool out of Docker image,
3. Merging the tools into a CWL workflow, and
4. Creating an appropriate frontend UI.

Thanks to established streamlined integration guidelines, the platform can be easily extended with new workflows, making it a sustainable, future proof product. However, one has to take care of the UI, the task which has to include a certain degree of manual development.

3. USE CASE EXAMPLES

The first encounter with the platform is login. As mentioned, besides regular SGABU accounts, Google accounts are fully supported, significantly lowering the barrier for new users. The access to each module on the SGABU platform is provided through the main dashboard as presented in Figure 2. For a user with administrative rights, there are additional options directed towards managing access rights to the regular users. Upon login, a regular user is presented a list of datasets and workflows he/she is allowed to interact with. The sections of biomedical fields covered by SGABU (Bone modeling, Cancer modeling, Cardiovascular disease modeling, Tissue engineering) are marked with different color. Each section contains datasets (upper line) and real simulation workflows (bottom line). SGABU acts as a stateful platform, saving all user activities and providing read and write access to all previous experiments (workflow executions). Thanks to this feature, it is possible to use it for certain simplistic research tasks, besides its primary educational purpose.

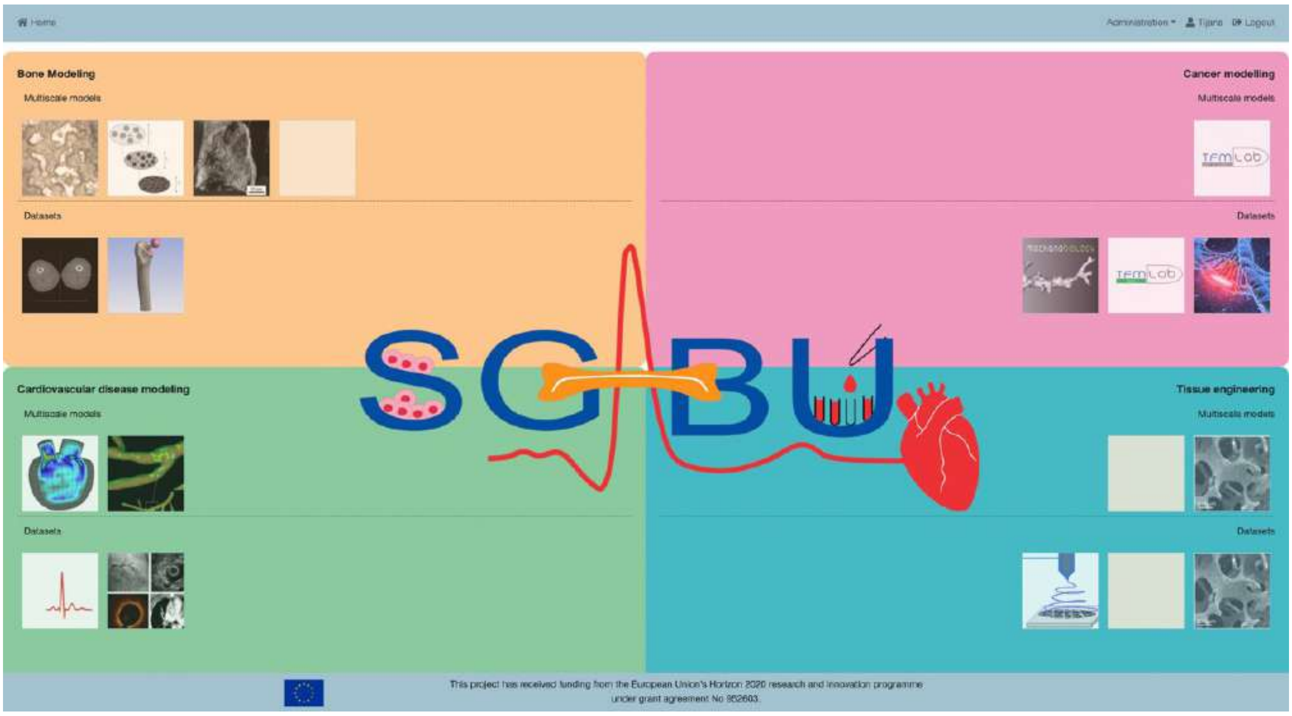


Figure 2. *Main platform dashboard*

3.1. Use case example of a dataset

The example of the dataset is collection of images of Hip joint with femoral implant dataset, which belongs to the Bone Modeling module on the SGABU platform. This dataset contains 301 DICOM images, obtained from a CT scanner [11]. The scans include one hip joint with artificial femoral prosthesis (left) and one healthy hip joint (right). User interface for this dataset is shown in Figure 3.

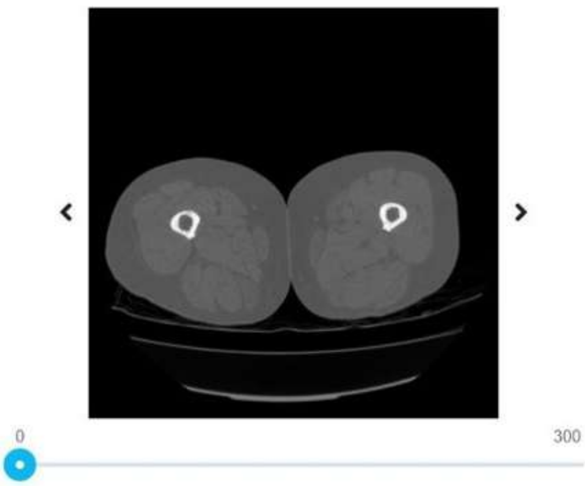


Figure 3. *User Interface for Hip joint with femoral Implant dataset*

A user can move through the dataset by clicking on left or right arrows next to the image, or by using the slider function below the image. The image 0 (default image after dataset is opened) represents the shaft ofthe femoral bone. As the number of the image increases the user will be able to see upper segments of the femoral bone and hip implants as well as the pelvic region.

3.2. Use case example of a model

One example of cardiovascular field model integrated into SGABU platform is a parametric heart modelof the left ventricle is used to simulate the cardiac cycle with patient-specific dimensions, which are providedby the user. The model is developed by [12]. Based on user-provided parameters, the left ventricle model is generated and finite element simulation is run with prescribed boundary conditions. Boundary conditions consist of inlet and outlet velocities and/or pressures, which are prescribed to the valves of the model. Users are also allowed to change stiffness of left ventricle wall, and calcium concentration which controls the muscle contraction.

User Interface for *ParametricHeart* example is shown in Figure 4. The window is divided into 2 sections – *Workflows* and *Add new workflow*. The basic execution unit in SGABU is a workflow. In the uppersection of the window, users are able to see names and status of the workflows. Possible statuses of the workflow are:

- Not yet executed
- Terminated
- Running
- Finished OK
- Finished Error

In the below section of the window, users can create new workflows for this submodule. Each of the forms needs to be filled out in order for simulation to run. Users are expected to fill out the following forms:

- Workflow name
- Left section:
  - Base division
  - Connection division
  - Aortic division
  - Wall division
  - Valves division
  - Mitral division
- Right section:
  - IVS-diastolic [cm]
  - LVID-diastolic [cm]
  - LVPW-diastolic [cm]
  - IVS-systolic [cm]
  - LVID-systolic [cm]
  - Inlet Velocity time function
  - Outlet velocity time function

The divisions control the number of finite elements within the model. Base division controls number of finite elements along the height of left ventricle base. Connection division controls the number of finite elements along the height between base and the valves. Valves division controls the number of finite elements along the width of valves. Aortic/mitral division controls the number of finite elements along the height of the aortic/mitral valve. Wall division controls the number of finite elements along the width of heart wall. Regarding the dimensions that need to be filled in, IVS-diastolic/systolic [cm] represents the interventricular septum (IVS) in diastole/systole, LVID-diastolic/systolic [cm] represents left ventricular internal diameter (LVID) in diastole/systole and LVPW-diastolic/systolic [cm] represents left ventricular posterior wall (LVPW) in diastole/systole.

All forms except for Workflow name are numerical and value ranges are provided for the users.

Figure 4 shows the velocity functions. Inlet velocity function is prescribed to the mitral valve of the left ventricle, while outlet velocity function is prescribed to the aortic valve. Interactive graphs with inlet andoutlet velocity time functions are also shown in Figure 5.

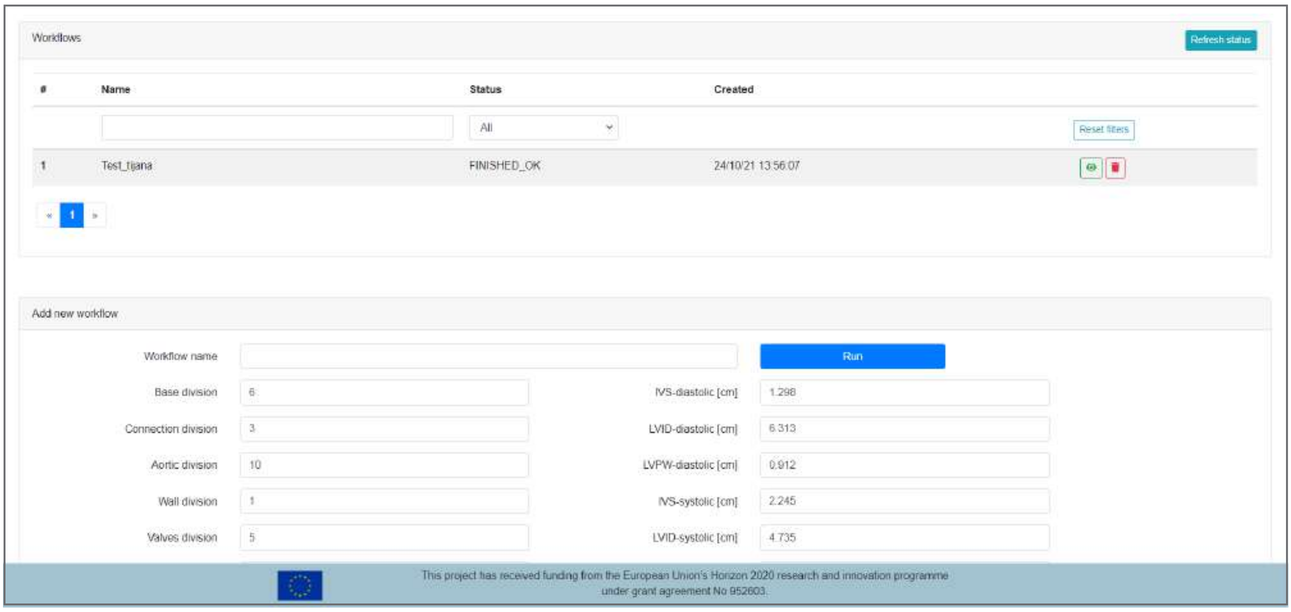


Figure 4. *User Interface for ParametricHeart module*

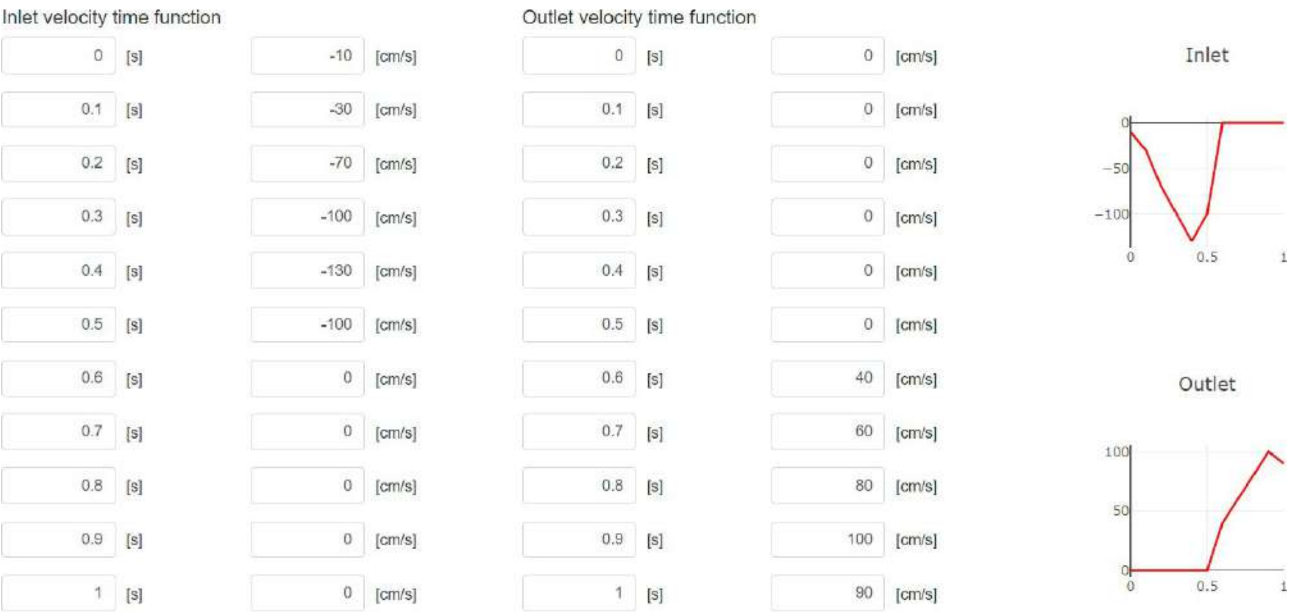
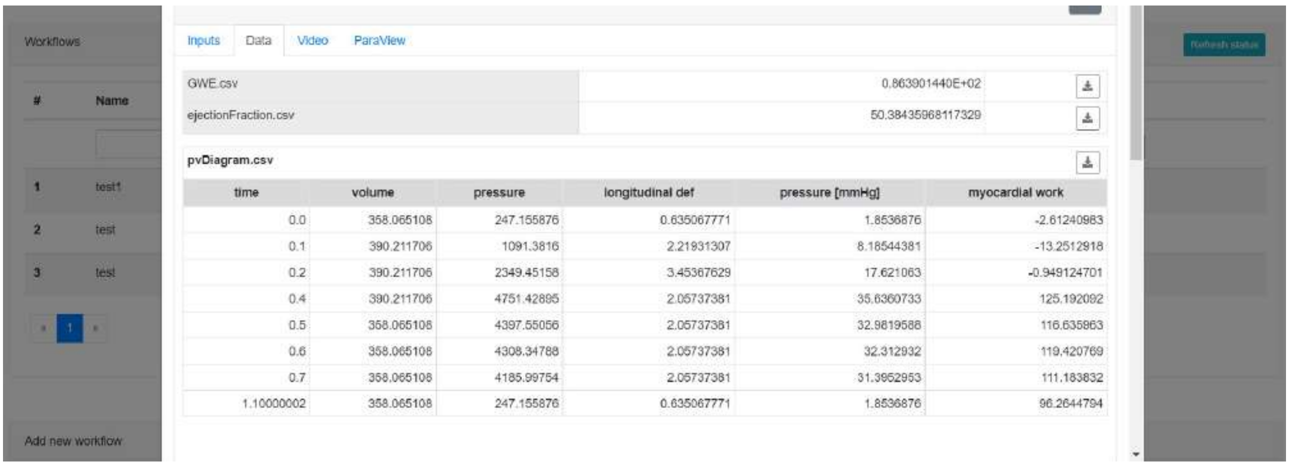


Figure 5. *Prescribing inlet and outlet time function for ParametricHeart module*

Figure 6. *Running the workflow*

The exception handling is integrated into the user interface (UI) of the SGABU platform (Empty forms, Non-numerical forms, Out of range values). Once everything is correctly filled, the workflow can be started. The user can monitor the current status of the workflow in the left section of the window (Figure 6).

Results are displayed in the form of tables, data, video and Paraview [7]. The result of the simulation consits of velocity, pressure and displacement fields during full cardiac cycle. Along the physical fields we also provied the pressure-volume diagrams and diagrams of myocardial work during cardiac cycle. Figure 7 shows input divisions, dimensions of the model and inlet velocity prescribed to the mitral valve.



time	volume	pressure	longitudinal def	pressure [mmHg]	myocardial work
0.0	358.065108	247.155876	0.635067771	1.8536876	-2.61240983
0.1	390.211706	1091.3816	2.21931307	8.18544381	-13.2512918
0.2	390.211706	2349.45158	3.45367629	17.621063	-0.949124701
0.4	390.211706	4751.42895	2.05737381	35.6360733	125.192092
0.5	358.065108	4397.55056	2.05737381	32.9819588	116.635863
0.6	358.065108	4308.34788	2.05737381	32.312932	119.420769
0.7	358.065108	4185.99754	2.05737381	31.3952953	111.183832
1.10000002	358.065108	247.155876	0.635067771	1.8536876	96.2644794

Figure 7. *Inputs tab of the results section*

The results can also be downloaded in the form of csv files. Figure 8 shows PV diagram, ejection fraction and global work efficiency obtained after executing our finite element simulation.

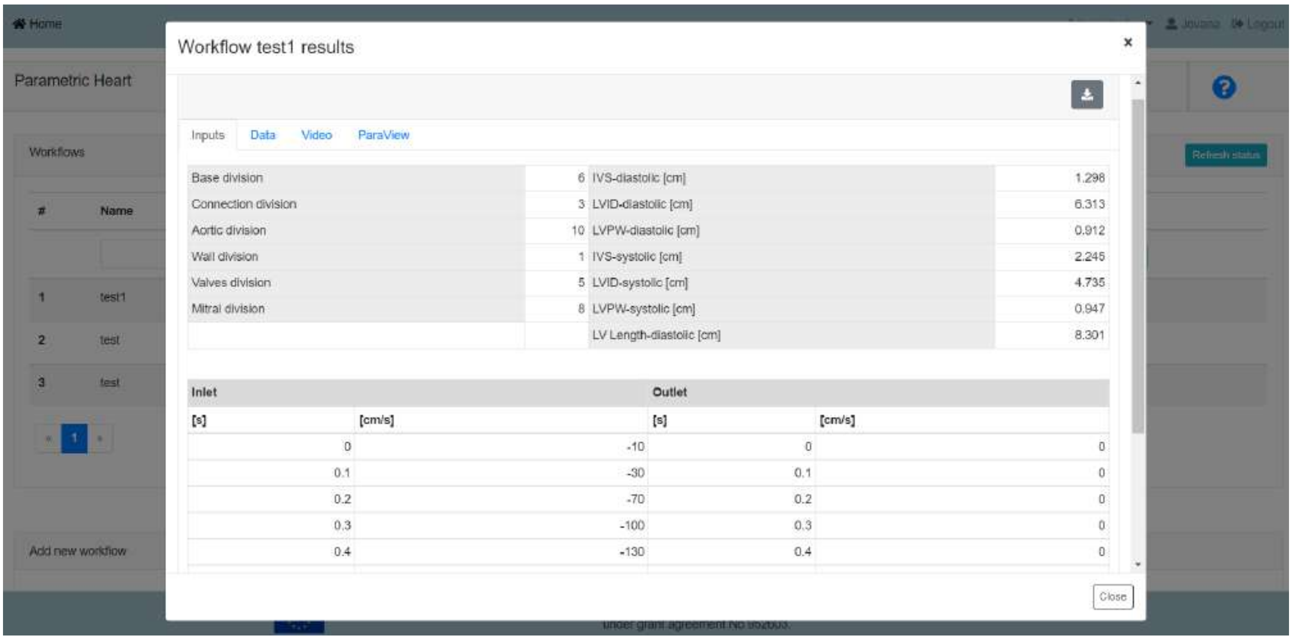


Figure 8. *Data tab of the results section*

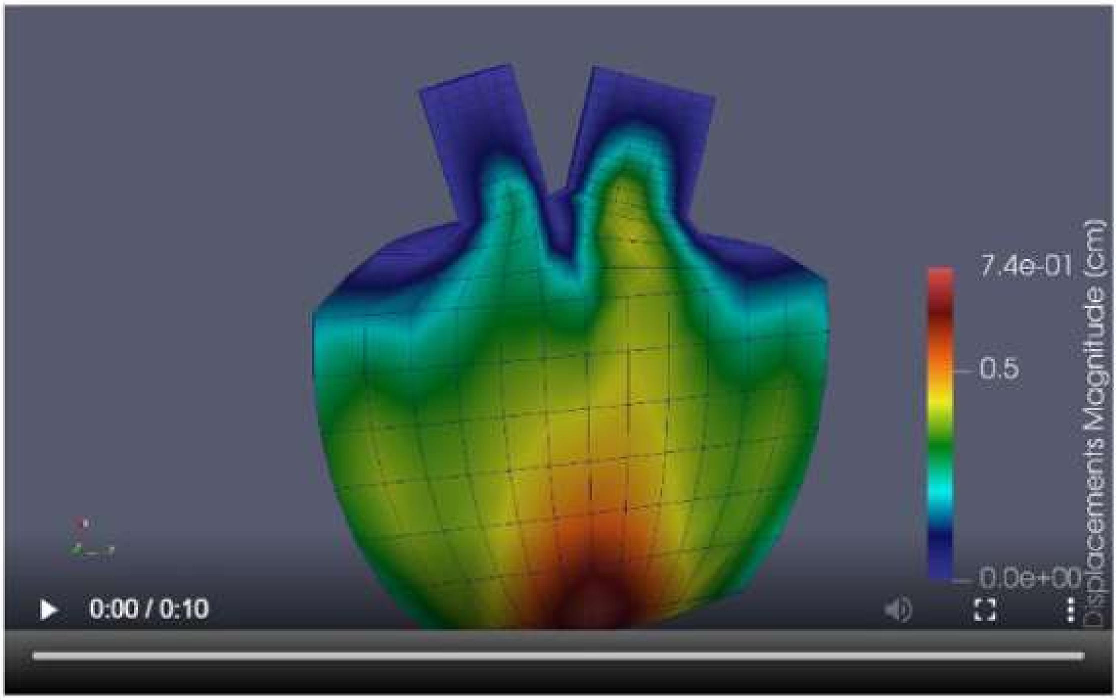


Figure 9. *Video tab of the results section*

Figure 9 presents pressure field within left ventricle model. Pressure is the largest during systolic phase of the cycle.

Clicking on the Paraview tab will open a new browser tab with results. Figure 10 presents geometry of the left ventricle.

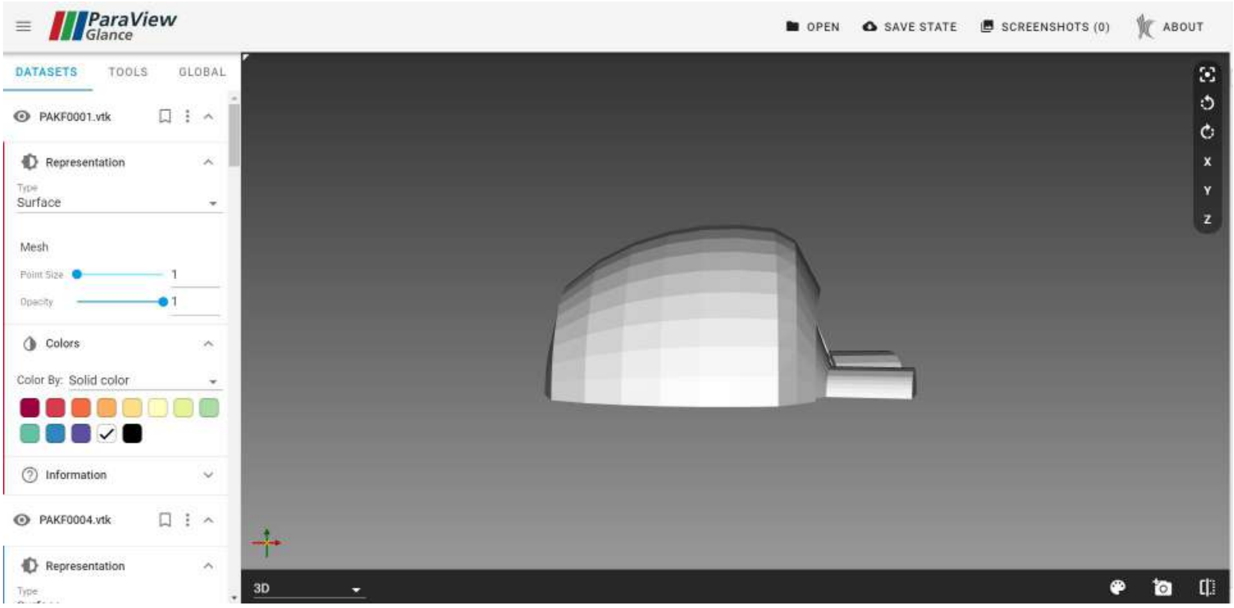


Figure 10. *Paraveiw tab of the results section*

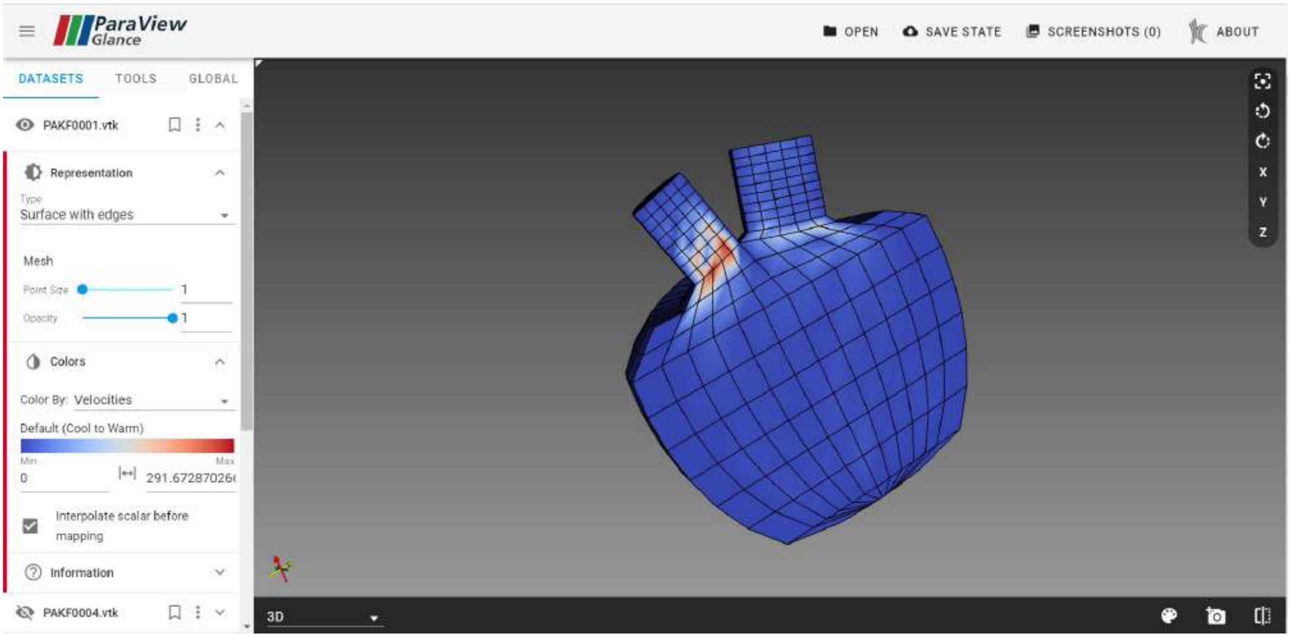


Figure 11. *Paraview velocities show of PAKF0001.vtk*

Figure 11 presents the results section in terms of velocity field. Inlet velocities rise during diastole and during systole inlet velocities are equal to zero. During systole muscles are activated and they push the blood out of left ventricle. At start of the systole the velocities are largest at the bottom of the heart. At the middle of the systole the velocities are the largest at the aortic valve.

4. CONCLUSIONS

This paper presents a platform for adaptive knowledge discovery of multiscale models and datasets. The primary benefit is the use of fresh, current, and distinct technologies for different layers of its architecture. These technologies are fully independent of underlying infrastructure, allowing for the platform to be easily migrated to any private or public cloud if necessary. There are several benefits for students and researchers who utilize the platform. Since SGABU platform is executable within any web browser, users may utilize it without installing any new software on their local devices.

The platforms such as SGABU are not only software products with certain specific purpose and short to medium lifecycle. If the platform aims at sustainability and long-life cycle, special software engineering approach has to be carried out. Two important aspects were covered during SGABU planning and development. Firstly, all the components were built using standard, well accepted frameworks and interfaces.Secondly, FAIR principles were strictly implemented thanks to the usage of well accepted workflow management technologies. The final result is a portable and extensible product, ready to serve students and researchers.

5. ACKNOLEDGEMENT

This research is funded by Serbian Ministry of Education, Science, and Technological Development [451-03-68/2022-14/200107 (Faculty of Engineering, University of Kragujevac)]. This research is also supported by the project that has received funding from the European Union’s Horizon 2020 research and innovation programmes under grant agreement No. 952603 (SGABU). T. Šušteršič also acknowledges the support from L’OREAL-UNESCO “For Women in Science” National Fellowship program Serbia (2021 Fellows).

6. REFERENCES

[1] A. N. F. Versypt, *Multiscale modeling in disease*. Current Opinion in Systems Biology, vol. 27, (2021) 100340.

[2] Personalised and generalized integrated biomaterial risk assessment (PANBioRa), Accessed: 10.08.2021., Available: https://www.panbiora.eu/

[3] In Silico trials for drug tracing the effects of sarcomeric protein mutation s leading to familial cardiomyopathy (SILICOFCM), Accessed: 01.03.2022., Available: https://silicofcm.eu/

[4] Bioengineering and Technology platform (BET), Accessed: 15.03.2022., Available: https://www.epfl.ch/research/facilities/ptbet/

[5] Increasing scientific, technological and innovation capacity of Serbia as a Widening country in the domain of multiscale modelling and medical informatics in biomedical engineering (SGABU), Accessed: 16.08.2022., Available: http://sgabu.eu/

[6] Common Workflow Languarge, Accessed: 03.11.2021., Available: https://www.commonwl.org/

[7] Paraview-Glance, Accessed: 20.12.2021., Available: https://kitware.github.io/paraview- glance/index.html

[8] M. Bean, *Laravel 5 essentials*, Packt Publishing Ltd, (2015).

[9] S. Seshadri, *Angular: Up and running: Learning angular, step by step.* O’Reilly Media, Inc, (2018).

[10] Plotly, Accessed: 15.01.2022., Available: https://plotly.com/

[11] A. Vulović , N. Filipovic, *Computational analysis of hip implant surfaces*. Journal of the Serbian Society for Computational Mechanics, vol. 13 (2019) 109-119.

[12] M. Anić, S. Savić, A. Milovanović, M. Milošević, B. Milićević, V. Simić and N. Filipović, *Solutionof Fluid Flow Through the Left Heart Ventricle*, Applied Engineering Letters, vol. 5 (2020) 120-125, 2020.

SGABU ПЛАТФОРМА – ИНТЕГРИСАНА ПЛАТФОРМА ЗА СКУПОВЕ БИОМЕДИЦИНСКИХ ПОДАТАКА И МОДЕЛЕ НА ВИШЕ СКАЛА

Сажетак: Сврха SGABU платформе је да укључи различите моделе и скупове података у области моделирања на више скала. Главни аспект SGABU платформе јесу различити скупови података и модели на више скала који се односе на области рака, кардиоваскуларне болести, коштане и ткивне поремећаје. Са тачке гледишта интеграције скупа података, задатак захтева имплементацију корисничког интерфејса који укључује манипулацију или са табеларним подацима, или првенствено са скуповима података који захтевају даље подешавање у front-end-у помоћу технологија као што су Angular, Plotly.js, Paraview Glance, итд. Са тачке гледишта интеграције модела на више скала, већина модула које обезбеђује SGABU платформа имплементирана је у форми радних токова типа Common Workflow Language (CWL). Овај метод је очигледан избор јер користи Docker контејнере и на стандардизован начин представља улазе, излазе и међурезултате, обезбеђујући приступачност, интероперабилност и поновну употребу (FAIR принципи). Обезбеђивање радних токова типа CWL састоји се од две различите акције: (1) развој имплементације CWL- а на позадинском делу FES (Functional Engine Service) и (2) развој одговарајућег корисничког интерфејса. Оваква платформа демонстрира употребу различитих примера моделирања и илуструје процес учења од идеје до имплементације.

Кључне речи: моделирање на више скала, скупови података, интеграција, платформа.

Paper received: 23 August 2022 **Paper accepted: 05 December 2022**