

Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	A linked data visualiser for finite element biosimulations
Author(s)	Mehdi, Muntazir; Khan, Yasar; Freitas, Andre; Jares, Joao; Raza, Saleem; Hasapis, Panagiotis; Sahay, Ratnesh
Publication Date	2016-02-04
Publication Information	Mehdi, M., Khan, Y., Freitas, A., Jares, J., Raza, S., Hasapis, P., & Sahay, R. (2016). A Linked Data Visualiser for Finite Element Biosimulations. Paper presented at the 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), 4-6 Feb.
Publisher	IEEE
Link to publisher's version	http://dx.doi.org/10.1109/ICSC.2016.96
Item record	http://hdl.handle.net/10379/7471
DOI	http://dx.doi.org/10.1109/ICSC.2016.96

Downloaded 2024-04-26T11:03:32Z

Some rights reserved. For more information, please see the item record link above.



A Linked Data Visualiser for Finite Element Biosimulations

Muntazir Mehdi*, Yasar Khan*, Andre Freitas[†], Joao Jares*, Saleem Raza[‡], Panagiotis Hasapis[§] and Ratnesh Sahay*

*Insight Centre for Data Analytics, National University of Ireland, Galway,

Email: {firstname.lastname}@insight-centre.org

[†]University of Passau, Germany

[‡]Bielefeld University, Germany

[§]INTRASOFT International, Luxembourg

Abstract—Biosimulation models are used to understand the multiple or different causative factors that cause impairment in human organs. Finite Element Method (FEM) provide a mathematical framework to simulate dynamic biological systems, with applications ranging from human ear, cardiovascular, to neurovascular research. Finite Element (FE) Biosimulation experiments produce huge amounts of numerical data. Visualizing and analyzing this huge numerical biosimulation data is a strenuous task. In this paper, we present a Linked Data Visualiser—called SIFEM Visualiser—to help domain-experts to Visualise, analyze and compare biosimulation results from heterogeneous, complex, and high volume numerical data. The SIFEM Visualiser aims to help domain scientists and clinicians exploring and analyzing Finite Element (FE) numerical data and simulation results obtained from different aspects of inner-ear (Cochlear) model - such as biological, geometrical, mathematical, and physical models. We validate the SIFEM Visualiser in both dimensions of qualitative and quantitative evaluation.

I. INTRODUCTION

Mathematical models have been recently introduced into the study of human organ physiology and pathology, giving insight into the system's behavior and attributes that would have been impossible to have with human in-vivo studies. The mathematical models, specifically biosimulation models play a fundamental role in the scientific practice with regard to the understanding of biological systems [1]. To simulate dynamic biological systems, Finite Element Method (FEM) provide a mathematical and computational framework. The applications of FEM in terms of simulating biological systems range from human ear, cardiovascular, to neurovascular research. The Finite Element (FE) biosimulation experiments are performed on biosimulation models, where these models span across multiple, complex and semantically incompatible domains, such as biological models, geometrical structures and mathematical-physical models.

FE biosimulation models are complex and sophisticated systems. Additionally, visualizing and analyzing results of biosimulation experiments performed on these FE models can grow unmanageable, due to volume, heterogeneity, and complexity of numerical data. A biosimulation model is a system of mathematical equations encoded in a computational language, representing different and often heterogeneous mathematical parameters. There are various tasks involved in performing

a biosimulation experiment on a biosimulation model, such as defining geometry of the biological system, creating a mathematical mesh, defining input numerical parameters, solver usage, visualisation and result/output interpretation. Most of these tasks are usually performed in isolated environment, hence, a biosimulation experiment takes many hours. Among these tasks, visualisation, analysis, and interpretation of biosimulation experimental results is a challenging task for the domain-experts.

In this paper, we propose a Linked Data Visualiser that shows biosimulation results along multiple dimensions. The aim is to combine, link and Visualise different biosimulations data (such as biological, geometrical, mathematical, physical) for inner-ear (cochlea) mechanics. The main contributions of this paper are: i) Visualise and analyze high volume, heterogeneous, and complex numerical data; and ii) validating SIFEM Visualiser on a real-life use case including experimental datasets, terminologies, and models provided by clinical organizations. Although, the proposed platform is validated against an inner-ear use case, we argue that the platform can be tailored with minor adjustments to other FE biosimulation domains.

Paper Organization: Section II describes the motivational scenario behind our work; Section III details the SIFEM Visualiser; Section IV highlights the validation of SIFEM Visualiser across both dimensions of quantitative and qualitative evaluations; Section V describes related work preceding conclusions and future work given in Section VI.

II. MOTIVATION SCENARIO

Our work is motivated by the need of clinical organizations and labs conducting biosimulation experiments to understand the exact pathophysiological consequences and risk factors of hearing impairment in humans. Our work is conducted in context of the SIFEM EU project¹, which aims at developing an open-source linked data platform for Finite Element multi-scale modeling and biosimulation of the sensorineural hearing loss. There are three major parts to the ear, with distinct functions; The outer ear collects sound waves and funnels them towards the middle ear. The middle ear ossicular chain

¹<http://www.sifem-project.eu/>

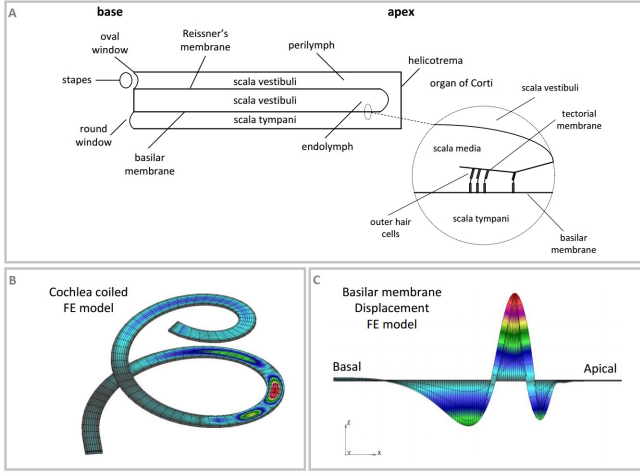


Fig. 1: Schematic representation of the human cochlea

oscillates in response to the airborne pressure waves, generating pressure waves in the inner ear fluid chambers. The inner-ear turns pressure waves into electrical signals that our brain can understand. The hearing impairments, which could lead to hearing loss, are mainly caused by cochlear and cochlear nerve pathology and are classified as sensorineural hearing loss [2].

The inner ear is inaccessible during life, which leads to unique difficulties in studying its normal function and pathology. Thus, biopsy, surgical excision and other conventional techniques of pathologic studies are not feasible, without further impairing function [2]. Mathematical modeling is therefore particularly attractive as a tool in researching the cochlea and its pathology. Mathematical models were introduced into the study of cochlear pathology and physiology, providing a useful tool in order to observe the system's behavior, which was impossible in previous human in-vivo studies [3]. The Finite Elements Method (FEM) – a mathematical framework – is assisting researchers in studying the structure-function relationship in normal and pathological cochlear. Figure 2 shows a set of FE parameters, values and their datatypes required to model the cochlea (inner-ear). Usually, hundreds of such parameters and billions of instances (i.e., values) are required to construct a full-fledged cochlea (inner-ear) model. Once a set of parameters and values are collected a numerical solver performs the finite element calculations with simulation results displayed in graphical format. Figure 1 shows schematic representation of the human cochlea with its main anatomical references, along with examples of simulation outputs from a numerical solver - called PAK Solver [4] - developed by the SIFEM consortium.

The proposed SIFEM Visualiser addresses following challenges in integrating, interpreting and visualizing numerical parameters along with simulation results: (i) resolve different data formats by transforming them into standard RDF format. For instance, all the eight (8) parameters and their values shown in the Figure 2 are taken from separate experimental data files stored in different formats (e.g, .dat, unv, .pak,

Parameter Type	Parameter Value	Datatype
MESH DIVISIONS	4.0 4.0 3.0	DIV_L, DIV_W, DIV_H
GEOMETRY	0.035 0.001 0.001 3.0E-4 5.0E-5	(DOUBLE)LENGTH (DOUBLE)HEIGHT (DOUBLE)WIDTH
LOAD FREQUENCY	50 1.0	(DOUBLE)FREQUENCY (DOUBLE)VALUE
MATERIAL PROPERTY	1000 1500.0	DENSITY_RHO, (DOUBLE)SPEED_OF_SOUND
YOUNG MODULUS FUNCTION	0.000000 14101716.454877	(DOUBLE)X_AXIS, (DOUBLE)Y_AXIS
EXTERNAL LOAD FREQUENCY	1.000000e-03 1 1 1 0 1 1 1 0.000000e+00	DISPLACEMENT_X, DISPLACEMENT_Y, DISPLACEMENT_Z
HEADING CARD	6060 1 1 1 1 0	IFORM, ISOLVER
NODAL POINT DATA	1 0 0 90 0.000000 1.00e-002 2 1 1000.000	FREQUENCY

Fig. 2: Finite Element numerical parameters, terminologies and format

etc.); (ii) providing links across input parameters/values and simulations results in-order to reuse relevant data analysis in future experiments; and (iii) visualisation and data analysis over FE simulation results.

The cochlea (inner-ear) represents a complex bio-mechanical device and a complete understanding of its behavior is still an open research problem. To the best of our knowledge, the proposed SIFEM Visualiser is a first attempt to bring together numerical parameters, models, terminologies, storage, querying using Linked Data technologies for providing visualisation and analysis over finite element biosimulations.

III. SIFEM VISUALISER

SIFEM Visualiser is a Linked Data platform which aims at improving the automation in interpretation and visualisation of finite element (FE) models for inner ear (cochlea) mechanics in an integrated fashion. Figure 3 represents the systematic work flow of the SIFEM Visualiser.

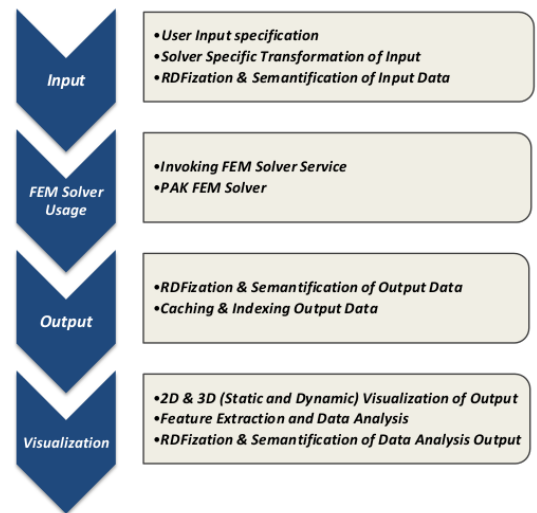


Fig. 3: SIFEM Visualiser Flow

a) **Input:** To start with a simulation experiment, the experimenter has to provide initial inputs for the experiment. These inputs are dependent upon the Finite Element Modeling (FEM) solver which has to be used in the simulation experiment. Each FEM solver has its own input and output data formats. The experimenter specifies the inputs, such as boundary conditions (material properties), mesh type (box model, coil model), through the SIFEM Visualiser interface and is then transformed according to the solver specific input data format. The input transformation is done by representing the input data using the SIFEM conceptual model, i.e. RDFizing the input data. Figure 2 shows a set of finite element input parameters, values and their data types required in a simulation experiment. Solver specific inputs are then generated from the RDFized input data. This approach leads us to overcome the data input heterogeneity of FEM solvers and makes SIFEM Visualiser a solver independent platform. A subset of input data in RDF is shown in Listing 1.

```
@prefix dao:<http://www.sifemontologies.com/ontologies/DataAnalysis.owl#> .
@prefix fem:<http://www.sifemontologies.com/ontologies/FiniteElementModel.
owl#> .
@prefix sim:<http://www.sifemontologies.com/ontologies/Simulation.owl#> .
@prefix pakFemSet:<http://www.sifemontologies.com/ontologies/
FEMSettingsPAK.owl#> .
@prefix bw:<http://www.sifemontologies.com/ontologies/BoxModel#> .

bw:BoxModel1ExcitationFrequencyValue
a sim:ScalarValue;
sim:hasScalarDataValue 1000.000.

bw:BoxModel1ExcitationFrequency
a fem:Frequency;
sim:hasScalarValue bw:BoxModel1ExcitationFrequencyValue.
bw:BoxModel1Load
a fem:Load;
```

Listing 1: Example Input RDF

b) **FEM Solver Usage:** Once the input parameters and their values are collected and transformed into solver specific format, the next step is to select and invoke the numerical Finite Element Methods (FEM) solver service, such as PAK, OpenFOAM etc. In our use-case, we use PAK FEM Solver, and to invoke the solver, we have defined a service. This service has the provision that the experimenter can select the numerical FEM solver according to his requirements. The transformed input data is then fed to the selected FEM solver and it performs the finite element calculations on the input data and respective output is generated.

c) **Output:** The output generated by the FEM solver is in solver's native format, which leads to data heterogeneity - in terms of data interpretation using different tools (like matlab, RStudio etc.). To achieve interoperability and resolve heterogeneity at the output level, the output is transformed into a standard and open format, i.e. RDF. Again, the RDFization of output data is conducted using the SIFEM conceptual model. An example subset of output data in RDF is shown in Listing 2.

```
@prefix dao:<http://www.sifemontologies.com/ontologies/DataAnalysis.owl#> .
@prefix fem:<http://www.sifemontologies.com/ontologies/FiniteElementModel.
owl#> .
@prefix sim:<http://www.sifemontologies.com/ontologies/Simulation.owl#> .
@prefix pakFemSet:<http://www.sifemontologies.com/ontologies/
FEMSettingsPAK.owl#> .
```

```
@prefix bw:<http://www.sifemontologies.com/ontologies/BoxModel#> .
bw:BoxModel1ExcitationFrequencyValue
a sim:ScalarValue;
sim:hasScalarDataValue 1000.000.
bw:BoxModel1ExcitationFrequency
a fem:Frequency;
sim:hasScalarValue bw:BoxModel1ExcitationFrequencyValue.
bw:BoxModel1Load
a fem:Load;
fem:holdsValueFor bw:BoxModel1ExcitationFrequency.
```

Listing 2: Example Output RDF

d) **Visualisation:** The next step is to show the simulation results in a graphical format and to analyze the simulation results. To do so, SIFEM Visualiser extracts the numerical results from the RDF output data to Visualise it in either two-dimensional or three-dimensional form. The output data is persisted in the triple store and is also temporarily indexed in cache. Since, visualizing different parameters of simulation experiments from output, there is a need to frequently query the output, the cache thus supports in order to overcome unnecessary delays caused by querying the whole triple store. The triple store persists data originating from every simulation experiment for the purpose of reusing the simulation experiment data. For instance, if a simulation has been performed in past, the platform will retrieve it from the triple store and Visualise the results rather than performing a whole new simulation. Furthermore, the data persistence will help the experimenter to re-run or cross-compare different simulation experiments already performed. As, the data in triple store will increase with every new simulation experiment performed, the query execution time to fetch visualisation data will also increase. Therefore, the caching will support in order to increase query performance by reducing the time required to execute a query. The SPARQL query used to retrieve all the necessary parameters for visualisation data from the triple store or Jena in-memory model is shown in Listing 3.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX fem:<http://www.sifemontologies.com/ontologies/FiniteElementModel.
owl#>
PREFIX pak:<http://www.sifemontologies.com/ontologies/FEMSettingsPAK.owl
#>
PREFIX sim:<http://www.sifemontologies.com/ontologies/Simulation.owl#>

SELECT ?xCoords ?yCoord ?zCoord ?nodes ?translationXs ?translationY ?
translationZ ?ids
WHERE {
?material rdf:type fem:Material; fem:hasMaterialNumber ?y; pak:
hasMaterialSettings ?z .
?z pak:MATTYPE ?materialType. FILTER (?y=1) . ?nodes rdf:type fem:Node .
?nodes fem:isNodeOf ?subDomain. ?nodes fem:hasNodeID ?ids.
?subDomain fem:makesUp ?subDomainGroup. ?subDomainGroup fem:
hasMaterial ?material.
?nodes fem:hasXCoordinate ?xCoords . ?nodes fem:hasYCoordinate ?yCoord .
?nodes fem:hasZCoordinate ?zCoord . ?nodes fem:holdsValueFor ?b .
?b rdf:type fem:Translation. ?b sim:hasVectorValue ?a .
?a sim:isReal true . ?a sim:hasVectorXValue ?translationXs .
?a sim:hasVectorYValue ?translationY . ?a sim:hasVectorZValue ?translationZ . }
```

Listing 3: SPARQL query for retrieving visualisation data

The parameters listed in the "SELECT" clause of the SPARQL query (Figure 3) and their respective values are stored in the cache while visualizing a simulation experiment

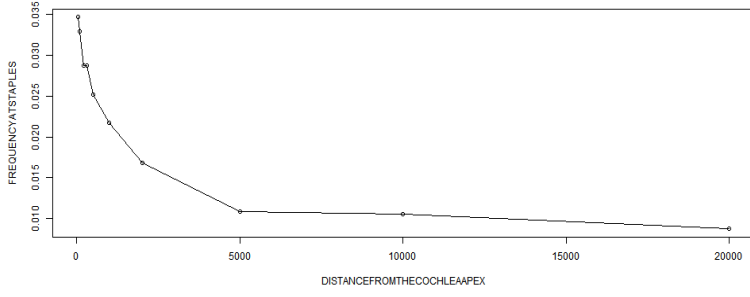


Fig. 4: 2D visualisation Displacement Example

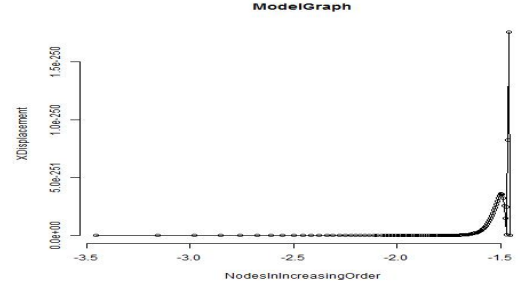


Fig. 5: 2D visualisation Greenwood Example

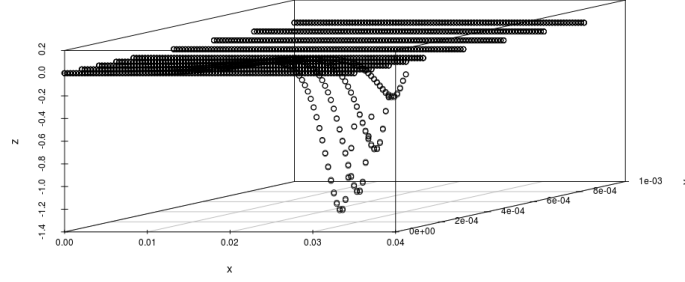


Fig. 6: 3D visualisation Example (static)

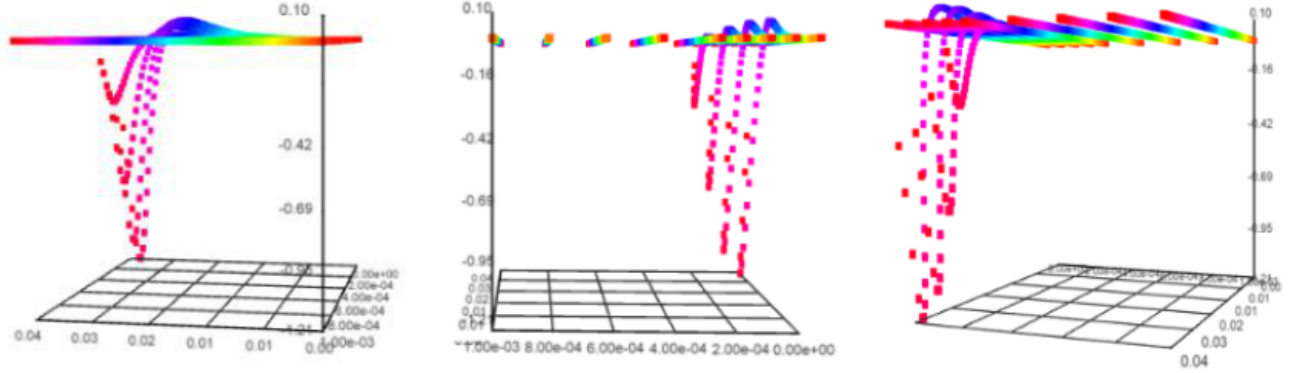


Fig. 7: Dynamic 3D visualisation Example (X-, Y- and Z-axis respectively)

performed in past. However, in case of visualizing a new simulation experiment, the same parameters along with their values are cached at run-time (before storing data in triple store).

To Visualise the experimental data, we generate two types of graphs (2D and 3D). The 2D graphs are static (non-movable) while the 3D graphs are static as well as dynamic (movable across axis). The dynamicity or ability to move 3D graphs help the experimenter to analyze the results in a more detailed and convenient way. Figure 4 and 5 shows simulation results Visualised in two-dimensional format for *displacement*² and *greenwood*³ functions respectively. The graph plotted for

²The displacement function, uniquely defines strain within an element in terms of nodal displacements.

³ The Greenwood function describes the relationship between the frequency of a pure tone and the position of the hair cells measured as the fraction of the total length of the cochlear (inner-ear) spiral in which it resides.

displacement (ref: Figure 4) function is against “Frequency at staples” parameter and “Distance from the cochlea apex” parameter. Similarly, the graph plotted for *greenwood* (ref: Figure 5) function is against “X Displacement” parameter and “Nodes in increasing order” parameter. Figure 6 shows a static (non-movable) 3D visualisation of parameters related to the *greenwood* function application in the cochlea mechanics. The dynamic (movable) visualisation of the same example is shown in Figure 7, where each axis can be seen individually (X-, Y- and Z-axis from left to right respectively).

The plotted graphs are then analyzed and important features (Examples of such features are graph behaviors, such as Periodic, Linear etc) are extracted from the graph. This is done by analyzing the numerical data plotted in the graph. The output of the data analysis is then transformed to RDF using the SIFEM conceptual model to increase data interoperability

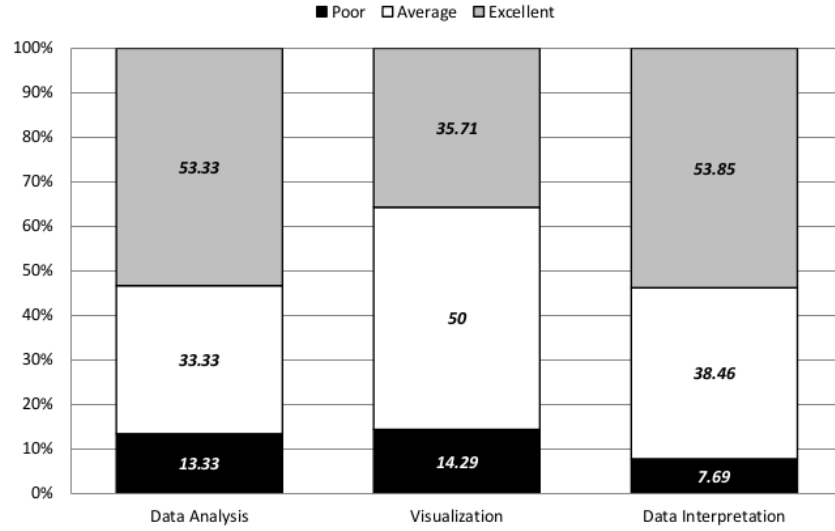


Fig. 8: Qualitative Results

of the SIFEM Visualiser. Example data analysis output in RDF is shown in Listing 4.

@prefix dao:<http://www.sifemontologies.com/ontologies/DataAnalysis.owl#> .

```
:DataViewBMMagnitude dao:hasDimensionX :translationX ;
dao:hasMinimumX '4.2027E-258' ;
dao:hasMaximumX '2.57604E-258' ;
dao:hasDimensionY :translationY ;
dao:hasMinimumY '2.95927E-257' ;
dao:hasMaximumY '3.59028E-257' ;
dao:hasGlobalMinima '(4.2027E-258, 2.95927E-257)' ;
dao:hasGlobalMaxima '(2.57604E-258, 3.59028E-257)' ;
dao:hasFeature dao:Periodic .
```

Listing 4: Semantic Interpretation of Data Analysis Output

IV. EVALUATION

This section elaborates the results obtained from empirical evaluation of SIFEM Visualiser. We have performed both qualitative and quantitative evaluation. **To perform qualitative evaluation** of the platform, the domain-experts and the clinicians were involved in a survey (questionnaire) to evaluate SIFEM Visualiser. And **to perform quantitative evaluation**, the experiments to record execution time of individual tasks leading to visualisation and interpretation of biosimulation experimental data were conducted in a distributed environment containing different dedicated servers.

Technological Profile: The system has been developed as JAVA J2EE Web-application using Apache Jena, Spring Framework, rJava⁴, and Shiny by RStudio⁵. Openlink Virtuoso has been used as a triple store for storing semantified FEM data and MongoDB has been used for caching and indexing purposes. The application is deployed on Wildfly⁶ (JBoss 8) Application Server on a dedicated machine. Both MongoDB and Virtuoso have been hosted on two separate machines. Additionally, since the platform uses PAK Solver for performing

simulation experiments, a dedicated machine, running windows OS has been used for hosting PAK Solver.

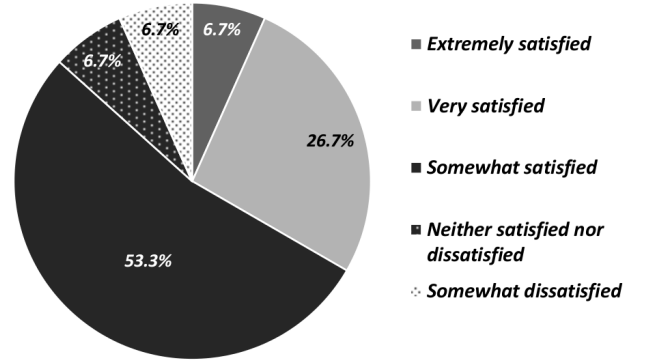


Fig. 9: Over-all satisfaction

Qualitative Evaluation: To perform qualitative evaluation, we created a survey to gather feedback from domain-experts. The survey was targeted to assess the quality of Data Analysis, visualisation and Data Interpretation. Figure 8 depicts the results of the survey. We observe and conclude in the Figure 8 that 50% of domain experts were averagely satisfied with the visualisation results, while around 36% were extremely satisfied with the visualisation. Similarly, for both data analysis and interpretation most of the domain-experts were highly satisfied with the output. A very few domain experts rated the results of data analysis, visualisation, and data interpretation to be poor.

The overall satisfaction of the users (domain-experts) is shown in the Figure 9, where 6.7 percent of the users are extremely satisfied with the system and 26.7 percent of the users are very satisfied. Similarly, around 27 percent of the users are somewhat satisfied with SIFEM visualiser. It can be noted that around 14 percent of the users are below the satisfaction level of the system. The detailed results of the survey are available

⁴<https://www.rforge.net/rJava/>

⁵<http://shiny.rstudio.com/>

⁶<http://wildfly.org/>

TABLE I: Execution time (ms) for biosimulation tasks in a single work-flow

Task	Execution Time (milliseconds)
<i>input.cfg</i> creation	219
<i>input</i> RDFization	1950
Solver execution	64233
<i>output</i> RDFization	4897
Querying Triple Store	98919
Querying in-memory Jena model	9603
Querying Cache	509
Data Analysis	738
Data Analysis output RDFization	217

at <https://goo.gl/WCulxW>⁷ and <https://goo.gl/PSvvLY>⁸.

It is important to note that currently – at the time of writing this article – the proposed platform is in the **beta-version 1** stage, therefore, the poor scoring given by the users largely reflect the improvement needed to be implemented for releasing a matured platform. For this purpose, a ticketing system to gather feedback of individual components of the SIFFEM visualiser has been setup. The users report any type of bug they have encountered or communicate any new type of feature they want to be added in the SIFEM visualiser.

Quantitative Evaluation:: Table I shows a set of tasks in a biosimulation work-flow – starting from input parameter collection to analysis and visualisation – completed in a total time of 181285 milliseconds compared to several hours taken to complete the similar work-flow in a normal biosimulation environment. Usually, all the tasks in Table I are performed in isolated environments and take substantial manual effort to align data, models, and simulation results to complete a work-flow. Since, varying simulation might have different input parameters, and different simulation functions, in order to ascertain the credibility of execution time - all the experiments were performed for a single type of simulation and function (i.e. *Displacement* function).

From Table I, it is also pertinent to note that the caching mechanism significantly reduces the execution-time as compared to while querying the triple store or in-memory model. We employed a heuristic approach in determining the importance of a caching mechanism. We RDFized FEM output of *Displacement* function for a single simulation experiment. RDFizing the FEM output produces a 7.8 MB file, serialized as *.ttl* format. The RDFized output contains 518305 number of triples and 22630 unique entities. We loaded the generated FEM RDF output into triple store and noticed a high amount of query execution time of 98919 milliseconds (ref: Table I). On querying the in-memory Jena model loaded with FEM output data, the query execution time of 9603 milliseconds was recorded. While querying the cache, the query execution time was reduced to 509 milliseconds.

This impacts the over-all execution time of a simulation experiment performed using the proposed platform. For instance, to Visualise the results of a simulation experiment for *displacement* function (after performing the simulation) takes: 1) 171 seconds - in case of triple store, 2) 82 seconds - in case of in-memory Jena model, and 3) 72 seconds - in case of using the cache.

V. RELATED WORK

Scientists and domain experts generally use well-established tools such as Matlab⁹, R¹⁰, etc. for the numerical computation and visualisation of biosimulation results. This works well in a closed environment where the numerical data is typically stored in a standalone machine. However, the process of integrating and visualizing biosimulation results from heterogeneous, distributed and large-scale numerical data repositories is a highly complex task and an open research problem. In order to integrate such diverse numerical data sets, authors in [5] proposed an ontology-based framework for finite element analysis in a product development environment. It uses a three-stage automated finite element modeling (FEM) method to identify and classify structural configurations and analysis modeling knowledge into a set of formal ontologies described in OWL. Gennari et al. [1], [6] integrates and Visualise three different biosimulation models of the heart, at three different scales (i) a cardiovascular fluid dynamics model; (ii) a model of heart regulation; and (iii) a sub-cellular model of the arteriolar smooth muscle. The representation and visualisation of biosimulation models have been steadily developed in the literature (Sauro & Bergmann [7]), including CellML [8] and SBML [9]. Ontologies such as the Systems Biology Ontology (SBO) [10] and Terminology for the Description of Dynamics (TEDDY) [10] have been used to conceptualize biosimulation and its results.

All the approaches mentioned above focus on either building and/or visualizing ontologies for the finite element constructs into the various domains such as cardiovascular, product development, and systems biology. To the best of our knowledge,

⁷Results compiled in Excel

⁸Results compiled by SurveyMonkey

⁹<http://uk.mathworks.com/products/matlab/>

¹⁰<http://www.r-project.org/>

the proposed SIFEM Visualiser is a first linked data based initiative to develop a tool that combines together numerical parameters, models, terminologies, storage, and querying for visualizing and analysis finite element biosimulation results for the human cochlea (inner-ear).

VI. CONCLUSIONS AND FUTURE WORK

We present a linked data enabled visualiser for biosimulation experiments. We highlight the use-case of SIFEM project, specifically, to visualise FEM biosimulation experimental data of human inner-ear cochlea. The platform is developed on the use-case to understand the exact pathophysiological consequences and risk factors of hearing impairment in humans. The experimental biosimulation data and models are from scientific and clinical studies about the structure-function relationship in normal and pathological cochlear. Since, the platform is still in development phases, based on the qualitative user feedback, we conclude that the satisfaction level of domain-experts for data analysis, visualisation, and data interpretation is moderate. Recall, that most of the biosimulation tasks are performed in isolation with maximum human interference and manual labor. Therefore, based on the quantitative evaluation, we conclude that we have been successful in implementing a visualisation technique, which provides an integrated environment to domain-experts for visualising and perform biosimulation experiments in minimum time (72 seconds to be precise) as compared few hours of day.

As part of the future work, we are currently working on a comparison technique which will allow users to compare different biosimulation experimental data as well as analyze the major differences between two or more 2D or 3D graphs. Furthermore, we are also working on document analysis technique to extract graphs from already existing domain-specific literature, such that the users can compare the newly generated graphs with existing ones. At the time of writing this article we are switching to beta-version 2 based on the user feedback. Therefore, we anticipate further innovative techniques to be incorporated in the platform to improve user experience of visualizing FEM biosimulation data to a greater extent.

NOTE: *The initial version of the SIFEM Visualiser is released as an open source version, the source code of **beta-version 1** is available at: <https://goo.gl/sxePc8>. The SIFEM Visualiser can be accessed on <http://213.249.38.66:7071/Sifem/> and User-Manual on <https://goo.gl/hbTYZt>. A video, depicting a sample simulation experiment of the running SIFEM platform can be found on <https://goo.gl/wDC0am>.*

ACKNOWLEDGMENTS: *This publication has emanated from research supported in part by the research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 and EU project SIFEM (contract Number 600933).*

REFERENCES

- [1] J. H. Gennari, M. L. Neal, B. E. Carlson, and D. L. Cook, "Integration of multi-scale biosimulation models via light-weight semantics," in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. NIH Public Access, 2008, p. 414.
- [2] S. Merchant, M. McKenna, J. Adams, J. Nadol Jr., J. Fayad, R. Gellibolian, F. Linthicum Jr., A. Ishiyama, I. Lopez, G. Ishiyama, R. Baloh, and C. Platt, "Human temporal bone consortium for research resource enhancement," *Journal of the Association for Research in Otolaryngology*, vol. 9, no. 1, pp. 1–4, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s10162-008-0111-5>
- [3] S. T. Neely, "Finite difference solution of a two-dimensional mathematical model of the cochlea," *The Journal of the Acoustical Society of America*, vol. 9, no. 1, p. 69, 1981.
- [4] V. Isailovic, M. Obradovic, D. Nikolic, I. Saveljic, and N. D. Filipovic, "SIFEM project: Finite element modeling of the cochlea," in *13th IEEE International Conference on Bioinformatics and BioEngineering, BIBE 2013, Chania, Greece, November 10-13, 2013*, 2013, pp. 1–4. [Online]. Available: <http://dx.doi.org/10.1109/BIBE.2013.6701611>
- [5] W. Sun, Q. Ma, and S. Chen, "A framework for automated finite element analysis with an ontology-based approach," *Journal of mechanical science and technology*, vol. 23, no. 12, pp. 3209–3220, 2009.
- [6] M. L. Neal, J. H. Gennari, T. Arts, and D. L. Cook, "Advances in semantic representation for multiscale biosimulation: A case study in merging models," in *Biocomputing 2009: Proceedings of the Pacific Symposium, Kohala Coast, Hawaii, USA, 5-9 January 2009*, R. B. Altman, A. K. Dunker, L. Hunter, T. Murray, and T. E. Klein, Eds., 2009, pp. 304–315. [Online]. Available: <http://psb.stanford.edu/psb-online/proceedings/psb09/neal.pdf>
- [7] H. Sauro and F. Bergmann, "Standards and ontologies in computational systems biology," *Essays Biochem*, vol. 45, pp. 211–222, 2008.
- [8] C. M. Lloyd, M. D. Halstead, and P. F. Nielsen, "CellML: its future, present and past," *Progress in biophysics and molecular biology*, vol. 85, no. 2, pp. 433–450, 2004.
- [9] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden *et al.*, "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.
- [10] M. Courtot, N. Juty, C. Knüpfer, D. Waltemath, A. Zhukova, A. Dräger, M. Dumontier, A. Finney, M. Golebiewski, J. Hastings *et al.*, "Controlled vocabularies and semantics in systems biology," *Molecular systems biology*, vol. 7, no. 1, 2011.