Title	A Semantic Web Platform for Automating the Interpretation of Finite Element Bio-simulations
Author(s)	Andr\'e Freitas and Kartik Asooja and Joao Jares and Stefan Decker and Ratnesh Sahay
Publication Date	2014
Publication Information	Andr\'e Freitas and Kartik Asooja and Joao Jares and Stefan Decker and Ratnesh Sahay (2014) A Semantic Web Platform for Automating the Interpretation of Finite Element Bio-simulations Proceedings of the 7th International Workshop on Semantic Web Applications and Tools for Life Sciences, Berlin, Germany, December 9-11, 2014
Item record	http://ceur-ws.org/Vol-1320/paper_6.pdf; http://hdl.handle.net/10379/4822

A Semantic Web Platform for Automating the Interpretation of Finite Element Bio-simulations

André Freitas, Kartik Asooja, João B. Jares, Stefan Decker, Ratnesh Sahay

Insight Centre for Data Analytics
National University of Ireland, Galway

Abstract. Finite Element (FE) models provide a rich framework to simulate dynamic biological systems, with applications ranging from hearing to cardiovascular research. With the growing complexity and sophistication of FE bio-simulation models (e.g. multi-scale and multi-domain models), the effort associated with the creation and analysis of a FE model can grow unmanageable. This work investigates the role of semantic technologies to improve the automation, interpretation of FE bio-simulations. In particular, the paper focuses on the discussion of strategies to bridge the gap between numerical-level and conceptual-level representations, facilitating the interpretation of bio-simulation data. The discussion is grounded on the SIFEM platform, a semantic infrastructure for FE simulations for cochlear mechanics.

Keywords: Finite Element, Simulation Automation, Semantic Web, Data Analysis

1   Introduction

Finite Element (FE) models are numerical approaches used for finding approximate solutions for differential equations, and are examples of computational models which are present in different branches of biological research, from cardiovascular to inner-ear research. The construction of FE models is a high complexity task which depends on multiple steps, including the definition of a discretized geometrical model (a mesh), the definition of a physical-mathematical model, the selection of numerical methods, the visualization and interpretation of the results and the experimental/clinical validation of the model. Building a consistent and realistic FE model is an empirical and time consuming process, which depends on the composition and fine-tuning of different parameters. This complexity is expressed in the difficulty of building and validating FE Models and in reproducing and reusing third-party FE models.

This work describes the creation of an infrastructure to support a more automated interpretation of finite element simulations in bio-mechanics using Semantic Web standards and tools. As most of FE data is numerical, this work explores mechanisms to bridge data on the numerical to the conceptual (ontology) level, facilitating and automating the interpretation of the simulation results. In order to evaluate the proposed approach, a software data representation and analysis

infrastructure is developed and a realistic cochlear mechanics FE model is instantiated. The final approach supports automation in the interpretation of the simulation data.

2   Motivational Scenario

The simulation of the human cochlea (responsible for the transduction of mechanical sound vibrations to electric signals) depends on the creation of multi-domain and multi-scale simulation models. The human cochlea consists in a 35mm snail-shaped duct, which contains 3 different fluid filled canals separated by 2 membranes. The sound vibration comes from the middle ear through the oval window, which vibrates the fluid inside the scala vestibuli. The vibration generates the displacement of the basilar membrane, an anatomical structure which contains sensorineural cells receptors in the organ of Corti, which translates the mechanical vibrations at different positions in the basilar membrane into electric signals which follow to the human brain via the auditory nerve.

The cochlea represents a complex bio-mechanical device and a complete understanding of its behavior is still an open research challenge. The mechanics of the basilar membrane has an active behavior which supports the electromechanical reduction and amplification of sound at different positions in the basilar membrane. The creation of a complete cochlea mechanical model depends on the integration of models at different scales (e.g. basilar membrane, Organ of Corti and outer hair cells) and model domains (e.g. mechanical and electrical). Additionally, the model should be validated by clinical and experimental data.

3   Automating Simulation Data Analysis

The cycle of building a FE model consists in setting up the models' parameters, running a simulation, interpreting the results against experimental data. The process of building FE models depends on the exploration and tunning of large configuration space of input parameters' combinations. Currently, each simulation cycle execution is dependent on the human to interpret the results and to redefine the model parameters.

The interpretation of FE results is highly dependent on the visualization of 2D and 3D numerical data. A typical data interpretation workflow is depicted in Figure 1, where a 2D function plot is used to interpret the results of the simulation output. In the example, the flow of a fluid in a box with moving lid is depicted and validated by the properties of the graph *velocity vs. distance from cavity base*. A valid FE model for this scenario can be described in natural language as: *'formation of a velocity vortex, where the speed of the fluid is maximum at the lid proximity, decreases to zero by the middle of the box and then changes direction'.*

A user creating or tunning a FE model typically have an expectation for the simulation output behavior which can be used to validate the model. However, there is a conceptual gap between the *high-level features* that users expect in a
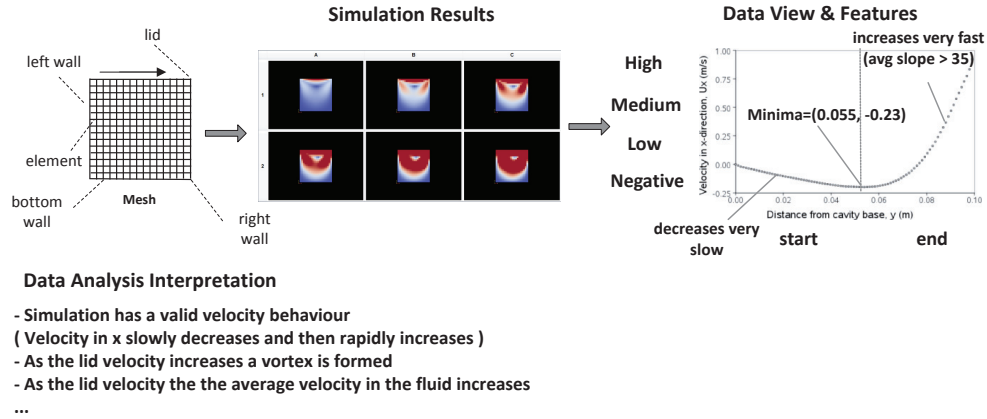
model and the numerical data. In order to address this gap, a set of *data analysis features* are defined which, combined with a domain conceptual model, supports the interpretation of the data and allows users to express the *expected model* as rules. These elements, combined with the possibility of defining value ranges for the input parameters, allows the definition of an architecture for automating the simulation tunning process, where the configuration space of input parameters are explored until the specified condition is reached. The triples below shows the semantic interpretation of the 2D plot at Figure 1.

```
:DataView1 :hasDimension :VelocityX .
:DataView1 :hasDimension :DistanceFromTheCavityBase .
:DataView1 :x0 0.0"^^xsd:double .
:DataView1 :y0 0.0"^^xsd:double .
:DataView1 :hasMinimumX -0.055"^^xsd:double .
:DataView1 :hasMinimumY -0.20"^^xsd:double .
:DataView1 :hasFeature :PositiveSecondDerivative .
:DataView1 :hasBehaviour :BehaviourRegion1 .
:DataView1 :hasBehaviour :BehaviourRegion2 .
:BehaviourRegion1 :avgFirstDerivative -3.63"^^xsd:double .
:BehaviourRegion1 :hasFeature EndRegion .
:BehaviourRegion1 :hasFeature :Decreases .
:BehaviourRegion1 :hasFeature :DecreasesSlowly .
:BehaviourRegion2 :avgFirstDerivative 33.35"^^xsd:double .
:BehaviourRegion2 :hasFeature EndRegion .
:BehaviourRegion2 :hasFeature :Increases .
:BehaviourRegion2 :hasFeature :IncreasesFast .
:BehaviourRegion1 :isFollowedBy :BehaviourRegion1 .
:LidSimulation :hasInterpretation :ValidVelocityBehaviour .
```

The components of the *data analysis workflow* are depicted in Figure 2.

- **Data View:** Data views provide a selective view of the values for input and output variables expressed in the RDF model and are used to define the functions relevant for the interpretation of the FE model (e.g. *basilar membrane displacement vs. distance from the basilar membrane basis*). Data views are defined using a SPARQL query over the simulation data.
- **Data Transformation:** Consists of data transformation operations over the *data view*. Typical data transformation operations in the FE context include unit conversions, definition of new reference frames (e.g. computing the displacement in relation to the cochlea apex instead of the basis), derivation of values for new physical quantities. Data transformation operations do not have a fine-grained semantic representation, i.e. the semantics of the data transformation is not captured in the conceptual model, just the reference to the associated program function. The main rationale behind the data transformation functionality is to facilitate the reuse of common operations in the context of a specific domain. Data transformations are defined under a publish-subscribe architecture, where the input variable names serves as subscriptions and the output is materialized back into the triple store. In the implemented architecture, *Python* is used as the data transformation language.
- **Feature Extraction:** Consists in the symbolic lifting of the simulation results by the definition of a set of *predefined data analysis features*, which lifts data from the numeric level to the conceptual level. The set of features were determined by taking into account the elements which domain experts use to analyze the data during the data visualization process (visual cues). The rationale is to pre-compute the data analysis features, allowing users to specify target features in the simulation data using high level features in combination with rules. Features are calculated using the same mechanism of data transformations. Examples of features are: *is periodic*, *monotonically increases*, *monotonically decreases*, *has maxima at*, *has minima at*, *has average first derivative*, *has average second derivative*. Features can be related by the *followed by* relationship.
- **Rule Application:** The component manages SPARQL Rules (SPARQL construct queries) expressing both data validation (rules applied over input data) and data analysis rules (rules applied over output data). It includes the visual interface for the rule editor, which abstract users in the creation of SPARQL construct queries.
- **Function Comparison:** Supports the comparison of the simulation results data with input reference data. Reference data are typically experimental measurements on a physical system (through physical experimentation or clinical analysis). Comparison between the output and reference data is done using *feature similarity* and *fitting errors*.

In the data analysis platform users can bind the variables returned by the data view SPARQL query to the input variables of the data transformation and feature extraction functions. Additionally, users can map the output of the function to the triple store. This is achieved by a publish-subscribe architec-

ture using aspect-oriented programming and a naming convention for SPARQL and program input/output variables. The set of data transformations, features, validation and data analysis rules can be reused across different simulations.
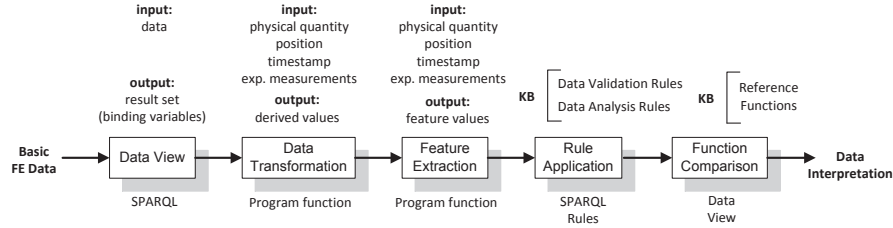

Fig. 2. Components of the data analysis workflow.

4   Architecture

The simulation workflow starts with the specification of the simulation input parameters and of different simulation scenarios on the *Simulation Manager* component (1). This specification of the simulation is stored in the RDF triple store using a *prospective provenance representation* [1,4]. The user then specifies the expected qualitative behavior of output variables through the *Rule Creator* component (2), using data analysis rules over *data analysis features*. After the specification of the simulation scenarios, the user starts the simulation using the *Simulation Manager* (3), which invokes the *Execution Manager* to coordinate the execution of the components of the simulation. The *Execution Manager* reads the solver input data from the *RDF triple store* and converts into the solver input format (*Solver Configurator* component) (4). The Execution Manager then invokes the solver for each simulation scenario, which generates the solver output (5). The *Semantic Converter* (6) component maps the Solver output data into the conceptual model (RDF vocabulary), persisting it as RDF. Based on the specified data analysis rules, the *Data Analyzer* (7) component extracts a set of data analysis features from the simulation data on the triple store, and verifies if the specified data analysis rules are satisfied. The *Data Analysis* output is also persisted on the triple store. After the data is persisted, users can access the data using the *Data Visualization* and the *SPARQL Endpoint* components.

5   Preliminary Validation

A preliminary validation of the proposed infrastructure was performed using the instantiation of the motivational scenario (the *Cochlea Box Model* as proposed

by [3]). The core goal of the preliminary evaluation was to verify the ability of the platform to support the representation of the resources, the integration of its components and an initial level of automated interpretation of FE data.

In the example scenario, the Greenwood function[1], the experimental baseline for the frequency mapping of the cochlea, is used as a reference function for the validation of the frequency behavior of the basilar membrane, as well as two validation rules.

The setup of the simulation starts with the user uploading the *geometric discretized model (mesh)* into the platform. Based on the mesh model, the geometric patches are loaded and the user specifies the *boundary conditions* for different simulation scenarios. The user then specifies the *material properties*, the *solver* and the *numerical method* for the simulation. In the demonstration example, the user specified a range of input frequency values at the *stapes*. Input parameters such as *fluid density* are validated against a reference model, by using validation rules (encoded as SPARQL Rules). The specification of the simulation scenarios is loaded into a triple store, following the conceptual model.

Simulation specification

- Frequency at stapes.
- Basilar membrane (Young's Modulus, Poisson's Ratio , Density).
- Fluid (Density, Speed of sound).
- Numerical Method: Incremental-iterative implicit solution.

Additionally, two support validation rules are used:

```
if excitation frequency is in the range 20-20000 Hz then
                simulation is valid
if not single wave then
                simulation is invalid
```

From the simulation specification, through the execution manager, the platform instantiates different simulation scenarios in the RDF triple store. The RDF simulation specification data for each scenario is loaded into the solver, which executes the simulation. For the Cochlea Box Model, the solver is solving the *FE equilibrium equation for solid and fluid* to calculate the *basilar membrane displacement*, *velocity* and *pressure*. The solver outputs the displacement, velocity and pressure fields for each element in the mesh, at different points in time, and for each input frequency, which is materialized at the triple store. The pre-defined data views are selected and the data transformations and features are computed and persisted as RDF. The last step consists in the application of the *data analysis rules* over the dataset, using SPARQL construct queries and the comparison with the reference validation data. After the simulation is finished, the interpretation of the simulation and different data visualizations can be selected by users. The associated interpretation for the numerical results is listed below:

---

[1] http://en.wikipedia.org/wiki/Greenwood_function

```
- Data View: (Frequency at staples x Maximum Basilar Membrane Magnitude point
  as a distance from the Cochlea apex)
- Fitting: Greenwood function with 10% error.
- Simulation satisfies rule:
        [if excitation frequency is in the range 20-20000 Hz then simulation is valid]
- Simulation satisfies rule:
  [if not single wave then simulation is invalid]
```

The features set for the *Basilar membrane magnitude × Distance from the Cochlea basis* Data View are listed below:

```
:DataView1 :hasDimension :BasilarMembraneMagnitude .
:DataView1 :hasDimension :DistanceFromTheCochleaBasis .
:DataView1 :hasFeature :isSingleWave .
:DataView1 :hasMaximumAmplitude 0.0031 "^^xsd:double.
:DataView1 :hasMaximumY 0.0020 e^-6 "^^xsd:double .
:DataView1 :hasMaximumX 14"^^xsd:double .
:DataView1 :hasMinimumY -0.0011 e^-6 "^^xsd:double .
:DataView1 :hasMinimumX 17"^^xsd:double .
```

The SIFEM platform was built as a Web application. A video of the running SIFEM platform can be found in `http://bit.ly/1rEZYh7`.

The triple store data is accessible through a SPARQL Endpoint where users can retrieve simulations with specific properties and compare different simulations. Examples of queries expressed in natural language are: (i) *What is the maximum basilar membrane amplitude and its associated frequency?* and (ii) *Which simulations are using the incremental-iterative implicit solution scheme?*.

The simulation and the generated RDF data can be directly linked from a publication: a user can click on a graph in an article and the associated simulation is opened on the SIFEM web platform. This allows the reader (who now becomes an interactive user) to re-run the simulations described in the paper, thereby supporting reproducibility and verifiability. The user is also enabled to run their own 'what if' scenarios using their choice of input parameters within the model, facilitating further experimentation (possibly exploring scenarios not envisaged by the original model creators).

The conceptual model was able to capture all the data generated in the simulation. A set of data-analysis features were defined as data interpretation primitives, which were used in data validation rules and for the comparison of the output to reference functions.

6   Related Work

Sun et al. [2] propose an ontology-based framework to increase finite element modelling (FEM) reusability in the design and solution of new models is proposed. It uses a three-stage automated FE method used in problem definition (for artifact markup), in problem formulation (for domain knowledge re-use) and in the solution routine (for the automation of FEM solution setup via parameter

reuse). Sun et al. [2] focuses on the mechanical engineering domain. Differently from [2], this work provides a conceptual model and framework which targets the *automated interpretation* of FE simulation data for the bio-simulation domain.

The representation of bio-simulation models has been steadily developed in the literature (Sauro & Bergmann [5]), including CellML [7] and SBML [8]. Ontologies such as the Systems Biology Ontology (SBO) [6] and Terminology for the Description of Dynamics (TEDDY) [6] have been used to conceptualize bio-simulation and its results. Comparatively, this work focuses on the provision of a reference architecture and implementation based on Semantic Web standards and tools for numerical data analysis. There is a strong connection between the data analysis feature set and the concepts modelled in TEDDY. We plan as future work to provide a better alignment between the data analysis features and the concepts described in the TEDDY ontology.

7   Conclusions & Future Work

This work describes an infrastructure which uses Semantic Web technologies to improve the automation of the interpretation of Finite Element (FE) bio-simulations. A reference architecture for automating the interpretation of FE data was proposed and a preliminary evaluation of the platform was performed targeting the validation of the data interpretation model. The evaluation used a realistic scenario based on a the FE simulation of a cochlear mechanics model.

References

1. Freitas, A., Kaempgen, B., Oliveira J.G., Curry, E., Representing Interoperable Provenance Descriptions for ETL Workflows, In Proc. of SWPM 2012, Extended Semantic Web Conference (ESWC), (2012).
2. Sun, Wei, Ma, Q., Chen, S., A Framework for automated finite element analysis with an ontology-based approach, Journal of mechanical science and technology 23.12, pp 3209-3220, (2009).
3. Elliott, S., Ni, G., Mace, B.R., Lineton, B., A wave finite element analysis of the passive cochlea, J. Acoust. Soc. Am. 133, 1535, (2013).
4. W3C PROV-O: http://www.w3.org/TR/prov-o.
5. Sauro, H. M., Bergmann, F. T., Standards and ontologies in computational systems biology, Essays in biochemistry, Vol. 45, pp. 211-222, (2008).
6. Courtot M., et al., Controlled vocabularies and semantics in systems biology. Molecular Systems Biology, 7:543, (2011).
7. Lloyd C. M., Halstead, M. D., Nielsen P. F., CellML: its future, present and past. Prog Biophys Mol Biol. ;85:433450, (2004).
8. Hucka M., et al., The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models, Bioinformatics, 19:524531, (2003).