

An Unstructured Semantic Mesh Definition Suitable for Finite Element Method

Marek Gayer, Hannu Niemistö and Tommi Karhela

VTT Technical Research Centre of Finland

Vuorimiehentie 3, Espoo, P.O. Box 1000, 02044, Finland

marek.gayer@vtt.fi , hannu.niemisto@vtt.fi , tommi.karhela@vtt.fi

Abstract—Numerical methods, such as finite element method (FEM), are widely used for computations on meshes in variety of simulation related software applications. As these methods become more popular in engineering, also information management requirements for these models are growing. Neutral data formats and standards are needed in order to achieve vendor independence and transformation between different tools. Semantic data modelling is one modern and potential technique that can be used for this purpose.

In this paper, we propose our Semantic mesh data structure that can be used in development of semantic software applications, which use meshes. The proposed structure is universal and can be used for representing meshes used in wide number of popular FEM applications such as e.g. Abaqus FEA, I-DEAS or MSC Adams. Our Semantic mesh proposal can be also used for mesh based simulations with other methods than FEM, such as e.g. finite volume method used in computational fluid dynamics (CFD). Also it can be used outside focus of numerical simulations, e.g. for projects related to scientific data visualization and computer graphics.

We provide a description of our Semantic mesh, together with an example of a very simple 2D mesh, illustrating our proposal. The implementation is using the semantic database in Simantics software platform.

Keywords—mesh format; semantic modelling; finite element method; numerical simulation on meshes

I. INTRODUCTION

A. Finite element method and meshing

The finite element method (FEM) is currently one of most often used techniques for numerical solutions of partial differential equations on complex modelled domains. FEM has become a de-facto industry standard for solving wide variety of multi-disciplinary engineering problems. There are many applications using this method, such as solid mechanics, fluid mechanics, heat transfer, acoustics, electromagnetics and computational fluid dynamics (CFD).

For numerical computation using the finite element method on general 3D modelled objects, we need to provide a suitable mesh representation of the continuum. This means to find a suitable discretization of continuous domain to simple volume cell elements (e.g. triangles or tetrahedrons). Such discretization forms a mesh. In the cells, partial differential equations (PDE's) can be replaced by systems of non-linear algebraic equations. Using the finite element basis, discrete algebraic systems are assembled into sparse

matrices and then solved. Computed characteristics are determined in the nodes of elements.

Computation of general partial differential equations using FEM is rather complex to design and implement, and solid understanding of mathematics, numerical methods and computer engineering is required for designing and implementing new performance optimal solutions of real problems based on this method. Therefore very often, instead of making such implementations, existing software solver libraries and commercial FEM solvers are used.

One of important advantages of FEM is availability of software tools, pre- and post-processors as well as software components based on this method. This allows for instance to automatically generate unstructured meshes on arbitrary structures, and there are some good open source mesh generators for FEM. One of advantages of FEM is in general lower requirements for the quality of the mesh than finite volume method. However, for various application tasks, such as flow problems, finite volume model is more natural and suitable.

B. Semantic data model

Our work is based on use of semantic graph as a unifying data model. By defining semantics, we are adding meaning of data objects by specifying their relations and by annotating those using statements. Data consists of resources, statements and values attached to some resources. Each statement connects two resources and belongs to some relation, so the graph can be seen as a collection of binary relations.

Another view is to see the set of all statements as a ternary relation of subject, predicate and object, where predicate is the relation where the statement belongs to. Seen this way, each statement is an atomic fact about the model that the graph describes. Naming of the relations tries to follow this viewpoint, so that for example the statement (*Pressure, HasUnit, Pascal*) can be easily interpreted. A special relation *InstanceOf* associates a resource with its type.

Fig. 1 depicts a very simple example of semantic data model, describing a pipe with diameter 0.142m.

Fig. 2 demonstrates another example, a semantic data model of a simple automation model in Apros¹ process simulation software, based on Simantics software platform².

¹ Apros software website: <http://www.apros.fi>

² Simantics software platform website: <https://www.simantics.org>

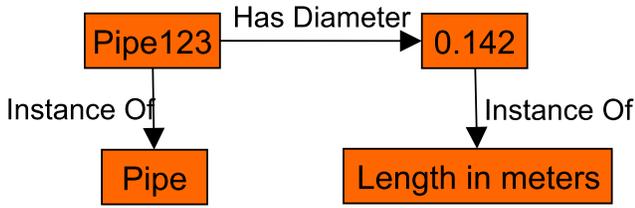


Figure 1. A very simple example of semantic data model

This automation model example contains an *Analog Switch*, which switches between two *Analog Signal* inputs. The *Analog Switch* is controlled by *Binary Signal* and passes its output as *Analog Signal*. To one of input *Analog Signals* is connected a *Set Point*, which can generate a concrete output value.

In Semantics, each resource must have at least one type. A resource with type *Literal* has a value as a data content. Value may be number, text, array, record or tagged union of values. In this respect, the data model is tailored to represent typical simulation configuration and result data efficiently compared to representing data with XML that is more common choice with semantic graphs.

What makes the data model special is that types and relations are also resources. This enables semantic data modeling, where meaning of the concepts can be described as a data and not just hardcoded to the application logic manipulating the data. It also increases flexibility. When

integrating data from multiple sources, data schemas are imported to the graph as easily as the data itself. Simple uniform representation makes it also possible to build data transformation and validation frameworks on top of the data model.

The semantic graph is organized into models and ontologies. Ontologies contain concepts shared by all models. Typically the ontologies describe types and relations and models use these concepts, but there is not so clear separation between schema and instance as in many other data modeling approach: ontologies may contain for example templates, viewpoints to data, and constant value and in some cases it is useful to define new types and relations in models.

II. RELATED WORK

We are not aware of any published work, which would be proposing to use semantic data modelling for description of whole mesh data representation. Nevertheless, there were numerous efforts proposing to use semantic descriptions for detections and segmentation of either volumetric or surface parts of the mesh, such as e.g. for parts of human body. A survey of such efforts can be found in [17]. In [1], segmentations of 3D surface meshes and semantic annotation of detected parts is presented. Another related work is a proposal of simplification of meshes, based on semantic mesh parts gained by face detector [3].

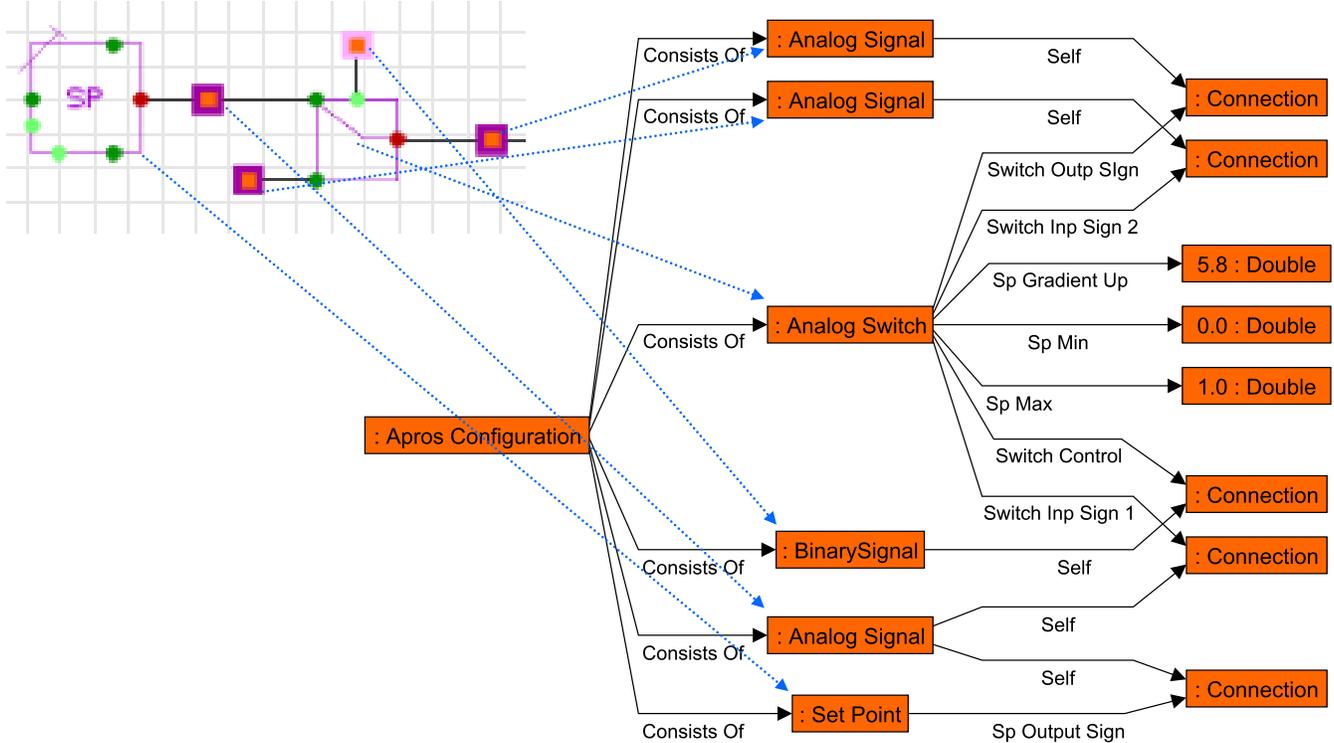


Figure 2. A semantic data model of a simple automation model in Apros software

There are many different ontology definition languages. For semantic graphs, the best known is OWL [2]³. It is however based on open world assumption, while in the context of simulation modeling closed world assumption is more useful. Therefore the ontology definition language of Simantics resembles more MOF⁴, which also makes UML class diagrams suitable for visualizing the ontologies.

The development of the Semantic Web⁵ and its technologies has increased activity and interest on semantic data management and its applications. Technologies such as the Resource Description Framework (RDF) [12], OWL, SPARQL Protocol and RDF Query Language (SPARQL) [15], and Semantic Web Rule Language (SWRL) [5], introduce a set of basic technologies to describe the data, to model ontologies, to describe semantic database queries, and to model rules and complex restrictions into ontologies respectively. The overall concept that these technologies introduce has inspired domains outside the Web technology to consider applying these methods for data management.

The foundations of core of Simantics software platform, which we use as the implementation environment for our Semantic mesh proposal, are in detail described in [8, 20, 10, 6]. Some Simantics related methods and concepts, using semantic and ontological representation have been demonstrated and documented separately. Using ontology-based 2D vector graphics, based on scalable vector graphics (SVG) was proposed in [9], a proposal of 3D models based on ontologies in large-scale process simulation can be found in [11], a marker based learning environment for detection of equipment in plants was described in [14]. Process simulation software applications, developed at VTT Technical Research Centre of Finland and widely used in industry, Apros [18, 13] and the next version of BALAS⁶ is integrated to Simantics. A computational fluid dynamics software environment, based on semantic modelling approach, designed to be suitable for integration with large-scale system level simulation tools for industry process simulation of plants was introduced in [4].

III. OUR PROPOSAL OF THE SEMANTIC MESH

A. *Need for a general Semantic mesh datastructure*

Features for different mesh formats differ, however the most basic concepts remain the same for all of them. A mesh (suitable not only for FEM) is a collection of nodes and elements (cells). Each of the elements is composed by at least one node. The cells can have dimension 0D (e.g. vertices), 1D (e.g. beams), 2D (e.g. triangles) or 3D (e.g. tetrahedrons). In practise, currently most commonly used are 3D and 2D meshes.

Meshes can have certain regularities. These regularities can be used for forming and describing structured meshes. In cases of arbitrary topology, unstructured meshes can be instead used. These meshes do not contain any regularity and therefore can be used to describe general structures and their corresponding meshes. Our current efforts as well as examples of our Semantic mesh in this paper use unstructured meshes.

The concept of nodes and elements is the most elementary and can be found also in applications, such as ParaView⁷ [19] based on VTK scientific visualization library⁸ [7, 16], which both we are using for external and internal visualization of meshes and data fields defined on them.

Another important aspect in mesh descriptions is organizing parts of meshes in groups. The group is basically a subset of nodes or elements. Groups are defined on nodes or elements (cells). For example, in a mesh representing a car, there can be a group on nodes or a group on elements present at the engine of the car.

The aim of this article is to propose a general, universal Semantic mesh data structure. Because of the flexibility of the semantic data representation, such structure could be used as a general base for eventual extension of Semantic mesh with other, for example less common features, which do not have to be present in all applications working with meshes (e.g. loads, boundary conditions, materials, etc.). For other, more specific needs and projects, these features might be added by extending the proposed Semantic mesh. For example, one of possible extensions includes also material databases, since e.g. simple FEM applications do not have to necessarily use these databases. Also, for solely visualization reasons, these databases also may not be necessary.

Although the proposal and examples were directed for FEM applications, the proposed Semantic mesh can be used also for simulation software applications using other numerical methods, such as e.g. finite volume method.

Our Semantic mesh definition unifies the organization and storage format of the meshes, but not the semantics of different features and properties, that may be format and application specific. We see two benefits in this approach: it allows importing data from many existing mesh formats, because new features and properties can be added and described when needed. On the other hand, it enables describing transformations between different mesh formats on a higher level: transformations do not need to consider the way the mesh data is serialized and transformations can be described by composing descriptions of how different properties are transformed.

B. *Requirements for a proposal of Semantic mesh*

One of most important problems to address is possible large amount of components in the mesh. Because there can be millions of nodes and elements, this can cause practical

³ OWL Web Ontology Language: <http://www.w3.org/TR/owl-features/>

⁴ MOF Version 2.0: <http://www.omg.org/spec/MOF/2.0/>

⁵ Semantic Web: <http://www.w3.org/standards/semanticweb/>

⁶ BALAS software website: <http://balas.vtt.fi>

⁷ ParaView website: <http://www.paraview.org>

⁸ VTK website: <http://www.vtk.org>

difficulties. Due to performance reasons and memory consumption requirements, it may not be possible to use semantic model, which would use too many (e.g. tens of millions) semantic statements. It would be necessary to propose a model with such features, which would use reasonable amount of semantic resources and statements, yet it would keep advantages that semantic modeling and abstraction offers.

Simplicity is another requirement – even in cases of very large meshes, we would like to use limited number of semantic resources and statements, yet at the same time utilize advantages of semantic modeling for rich description of the mesh data model.

One of means necessary to achieve such simplicity would be grouping of common features and operations by using means of data abstraction. Examples of such similarities in the mesh are nodes and elements. Both nodes and elements are specified using arrays of numerical values, both of them can be formed in groups; there are values assigned to them (such as physical properties gained by simulation analysis). Therefore, awareness and recognition of such similarities and their utilization is important for us, so we could achieve the requirement of simplicity.

Our proposal of the Semantic mesh also efforts to be extendible for various mesh data related features.

C. Our Semantic mesh datastructure definition

Our Semantic mesh model ontology is formed as a graph, see Fig. 3. The root element of the graph is *FEMModel*, which is a starting point to all other succeeding elements, which are children of this node, and which contain all data of the mesh model. We will now describe the most important features of our model.

1) Abstract data features notation

The proposed model is designed to be general. The same ontology can be used for various data, such as lists of nodes (*NodeList*), elements (*ElementList*) and coordinate systems (*CoordinateSystemList* - for cases when coordinates in elements and nodes can use different coordinate systems). There may also be other kind of features, which can be in case of necessity added by a developer. This is reflected by the abstract class *MeshItemList*, which is an abstraction of data features of the model. Each feature item has an integer identifier. The identifiers are stored in array *Indices*, which dimension reflects total count of features.

The further described concepts can be easily used for all classes inherited from *MeshItemList*. It is important to note that none of such classes actually contains any numerical data or fields. All the data, including e.g. nodes coordinates and element connectivity are contained in assignments relations, which we will describe below.

2) Groups

Groups allow us to properly focus on some subset of mesh features. These can be used for defining e.g. groups on mesh nodes or elements. The groups can be used in subsequent assignments: e.g. we can use the same group to assign two different physical properties. There is an abstract class *Group*, which is used to specify the subset to which a property will be assigned. *MeshItemList* is itself a group representing the whole mesh. Smaller groups can be defined as *GroupSelection* by enumerating the items of the group.

Groups as well as all other data items can be named. The model could be extended to allow definition of groups in some other way than *GroupSelection*, by inheriting from *Group*.

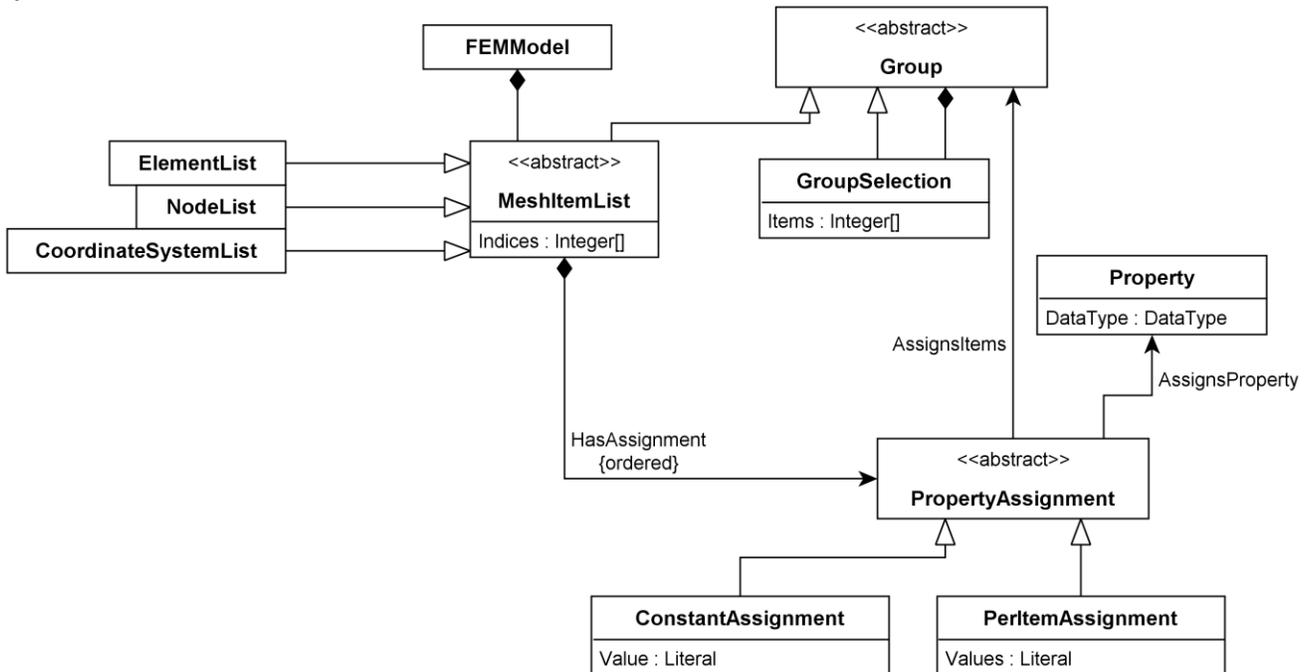


Figure 3. Semantic mesh abstraction described by an UML class diagram

3) Property assignments

The actual data of *MeshItemList*, is described using assignments. An assignment points to a *Group*. Subset of a mesh feature selected using the group will be assigned.

Data are attached as literals and specify concrete values that will be assigned to the items in the group, to which the assignment is attached. Assignments link to a *Property* (each of these is pointed with its corresponding relation *AssignsProperty*), which specify type and name of the corresponding data to be assigned.

Assignments are inherited from abstract class *PropertyAssignment*. In case of *ConstantAssignment*, all features will be assigned the same value. In case of *PerItemAssignment*, the data is an array of values that assign different values to all features in the selection in the same order the features are listed in the selection. All the various attached data can be in this way provided as an array (list) of values with type (e.g. double or integer).

Assignments are ordered. If two assignments assign different values of same property to the same item, the value that was assigned later is chosen.

More structured types can be used as well. For example, for a node it could be a structure with x, y, z coordinates and an identifier specifying chosen coordinate system for the node.

New property assignments can be added to our Semantic mesh proposal, for example to describe a property by a formula that depends on other properties.

IV. AN EXAMPLE MODEL OF OUR SEMANTIC MESH

We present an example of a very simple Semantic mesh, on which we will illustrate described mesh Semantic mesh definition. Fig. 4 illustrates a simple 2D mesh with five nodes, four elements (e.g. triangles) and two groups on nodes. Due to

limited space and effort to keep the example simple and brief, assignments of colour property to *GroupSelection* groups is demonstrated only on nodes, although it's possible to use these assignments with the same approach also on elements. The example contains three assignments to nodes and one assignment to elements. Two assignments of colour property overlap on nodes 10 and 40. Because the assignment that sets the green colour is defined later, the final colour of both nodes is green.

V. TRANSFORMATION OF SEMANTIC MESHES

An advantage of ontology description and semantic approach is in possibility of obtaining software and data abstraction, suitable for using with various types of mesh generators and other applications used in FEM modelling

Because various mesh based software applications do not rely on semantic software related approach, it is necessary to provide code that transforms our Semantic mesh to a format used by other software. For example, a code would make export of the Semantic mesh to an XML file, which can be imported to a FEM application.

By means of special semantic transformation features, we can utilize advanced means to help us with that. For example in Simantics software platform, there is Simantics Constraints Language (SCL), which can be used for this purpose. It is a programming language that can transform a data model using an ontology description, to another data model with a different ontology, or transform a data model into a different, non-semantically based format, such as a e.g. plain text or XML file. It can be e.g. used for exporting to binary or text file formats and databases, used by FEM applications for storage of meshes. At the same time, it is possible to use other means, such as e.g. using Java code to generate these files.

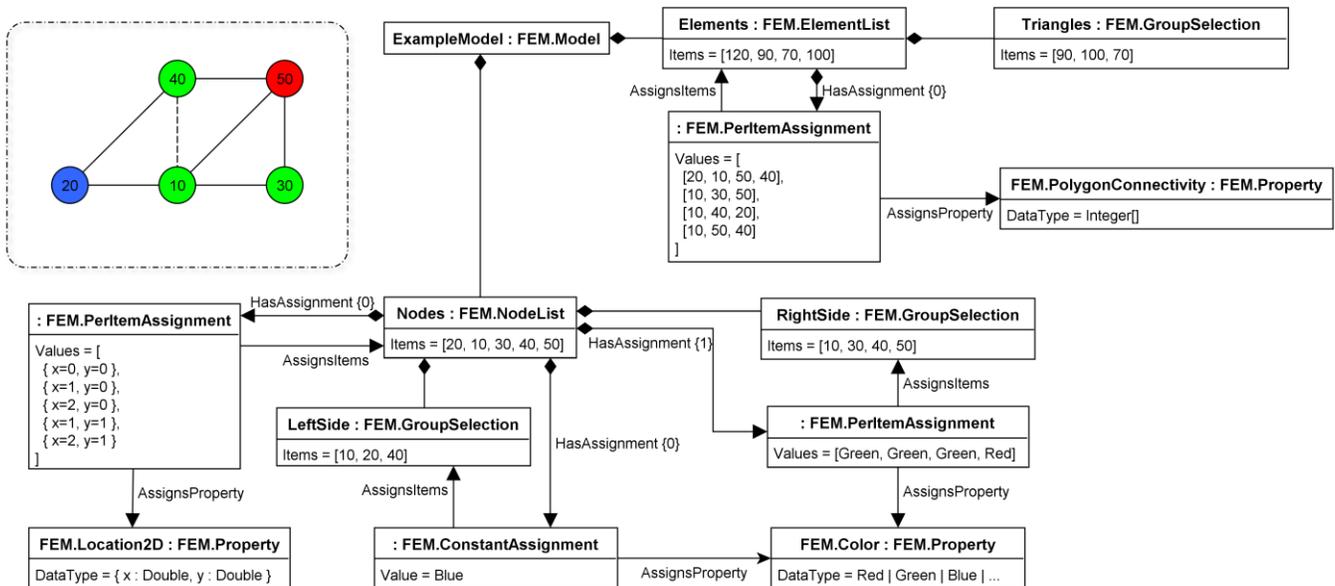


Figure 4. An example of Semantic mesh for a simple 2D Mesh

VI. IMPLEMENTATION

The implementation of our Semantic mesh definition proposal is realised in our CFD/FEM integration application, designed and implemented for working with various CFD and FEM solvers and mesh generators.

The application is using Simantics software platform and is implemented within the Rich Client Platform (RCP) of Eclipse Java development environment. The RCP environment provides suitable framework for creating appropriate user interface with similar look and feel for

applications based on this technology. The application has simple pre- and post-processing features which utilize VTK visualization library.

The application reads meshes from I-DEAS UNV format (Universal File Format) to Semantic mesh. The application allows making changes to the mesh model. It can visualize whole mesh, groups of nodes or elements, and assigned properties. It also allows exporting Semantic mesh model to the I-DEAS UNV format. User interface of this application with Semantic mesh ontologies displayed in the object tree with graph explorer is depicted on Fig. 5.

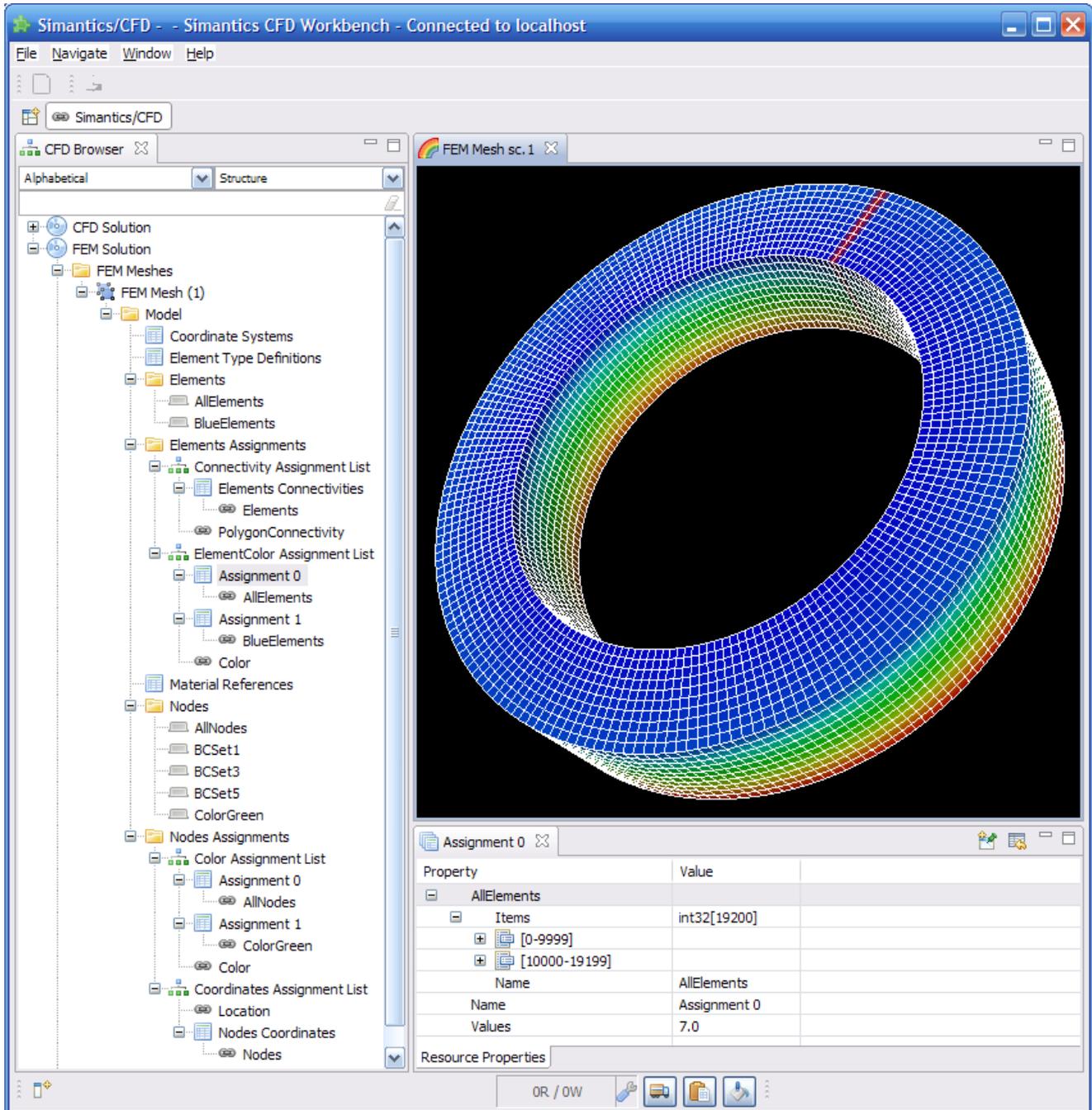


Figure 5. Inspecting a Semantic mesh model in our application based on Simantics software platform

VII. FUTURE WORK

Our on-going and also future work includes creating importers and exporters for the proposed Semantic mesh representation also from other FEM software applications than I-DEAS UNV format. We are currently working on supporting file formats used in Abaqus FEA. Another work is aimed to provide external visualization features, by using ParaView XML formats, so that we would be able to visualize the Semantic mesh and its data in ParaView.

Also we consider proposing of special mesh ontologies, which would be suitable for concrete FEM software applications. These would be different for each software applications (e.g. Abaqus FEA, I-DEAS and MSC Adams). An ontology oriented programming language, would be able to perform transformation from such a concrete ontology format into a more general ontology description, which would be same or similar as the ontology format described in this paper. Our Semantic mesh development is also a part of our projects dedicated for creation of a new CFD/FEM environment, with special impact of integration of 1D industry process simulators (especially Apros) with 3D simulations. The project is expected to provide pre-processing features (such as an interactive geometry editor based on OpenCASCADE⁹ geometry modeling kernel), mesh integration (free and commercial meshers), simulation model configuration using semantic approach, solver integration (such as OpenFOAM and FLUENT), controller for synchronization and data transfer between the 1D and 3D solvers, storage and management of data sets with results and post-processing features (namely visualization of data-sets with simulation results).

VIII. CONCLUSION

We have presented our proposal for Semantic mesh representation, suitable not only for meshes for finite element method applications. The proposal is described by an UML schema and we also provided an example model of the schema. The proposal was tested with meshes imported and exported from/to I-DEAS UNV format (Universal File Format) files, but can be used also with other mesh file formats. The Semantic mesh is proposed in the way that number of semantic components necessary to allocate in a semantic database is limited, and therefore can be used for even very large meshes with millions of nodes and elements with reasonable memory requirements. Our proposal has interesting features resulting in generality and extendibility. Our proposal was tested in the semantic database in Simantics software platform and in our FEM integration software application.

ACKNOWLEDGMENT

Our research work had received financial support from Tekes (the Finnish Funding Agency for Technology and Innovation), from VTT Technical Research Centre of Finland and from Finnish industry.

We thank our colleagues from VTT, especially Paul Klinge and Juha Kortelainen for their advices related to the structural analyses methods and tools.

REFERENCES

- [1] Marco Attene, Francesco Robbiano, Michela Spagnuolo, and Bianca Falcidieno. Semantic annotation of 3D surface meshes based on feature characterization. In Bianca Falcidieno, Michela Spagnuolo, Yannis Avrithis, Ioannis Kompatsiaris, and Paul Buitelaar, editors, *Semantic Multimedia*, volume 4816 of *Lecture Notes in Computer Science*, pages 126–139. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-77051-0_15.
- [2] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneijder, and Lynn Andrea Stein. OWL web ontology language reference. Recommendation, World Wide Web Consortium (W3C), 2004. See <http://www.w3.org/TR/owl-ref/>.
- [3] A. Colombo, C. Cusano, and R. Schettini. Semantic 3D face mesh simplification for transmission and visualization. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 157–160, 2006.
- [4] Marek Gayer, Tommi Karhela, and Juha Kortelainen. CFD modelling as an integrated part of multi-level simulation of process plants - semantic modelling approach. In *Proceedings of the 42th Summer Computer Simulation Conference (SCSC'10)*, pages 219–227. Society for Modeling and Simulation International (SCS), 2010.
- [5] I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosf, and M. Dean. SWRL: A semantic web rule language combining OWL and RuleML. W3c member submission, World Wide Web Consortium, May 2004.
- [6] Toni Kalajainen. An access control model in a semantic data structure: Case process modelling of a bleaching line. Master's thesis, Helsinki University of Technology, 2007.
- [7] Inc. Kitware. *The VTK User's Guide*. Kitware, Inc., 2010.
- [8] Tuukka Lehtonen. Ontology-based diagram methods in process modelling and simulation. Master's thesis, Helsinki University of Technology, 2007.
- [9] Tuukka Lehtonen and Tommi Karhela. Ontology approach for building and visualising process simulation models using 2D vector graphics. In *SIMS Proceedings of the 47th Conference on Simulation and Modeling*, pages 141–146. Finnish Society of Automation, SIMS - Scandinavian Simulation Society, 2006.
- [10] Marko Luukkainen. Use of 3D graphics for configuration and visualization of large scale process simulation: ontology-based approach. Master's thesis, Helsinki University of Technology, 2007.
- [11] Marko Luukkainen and Tommi Karhela. Ontology approach for co-use of 3D plant modelling and large scale process simulation. In *The 48th Scandinavian Conference on Simulation and Modeling (SIMS 2007)*, pages 166–172. Linköping University Electronic Press, 2008.
- [12] F. Manola and E. Miller. RDF Primer. W3C Recommendation, The World Wide Web Consortium, <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>, February 2004.
- [13] Anne Niemenmaa, Jari Lappalainen, Ismo Laukkanen, Sami Tuuri, and Kaj Juslin. A multi-purpose tool for dynamic simulation of paper mills. *Simulation Practise and Theory*, 6(3):297–304, 1998.
- [14] Srihathai Prammanee, Marko Luukkainen, Timo Seuranen, and Tommi Karhela. A marker-based mobile learning environment for a process plant. In *IADIS International Conference Mobile Learning*. IADIS, 2009.
- [15] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Recommendation, The World Wide Web Consortium, January 2008.
- [16] Will Schroeder, Kenneth M. Martin, and William E. Lorenson. *The visualization toolkit: an object-oriented approach to 3D graphics, 4th edition*. Kitware, Inc., 2006.
- [17] Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.
- [18] Eero Silvennoinen, Kaj Juslin, Markku Hanninen, Olli Tiihonen, Jorma Kurki, and Kari Porkholm. *The Apros software for process simulation and model development*. VTT publications, 1989.
- [19] Amy Squillacote. *The ParaView Guide*. Kitware, Inc., 2008.
- [20] Antti Villberg. Design challenges of an ontology-based modelling and simulation environment. Master's thesis, Helsinki University of Technology, 2007.

⁹ OpenCASCADE website: <http://www.opencascade.org>