NENE 2015 September 14-17 PORTOROŽ SLOVENIA
24th International Conference Nuclear Energy for New Europe

Interfacing of CAD models to a Common Fusion Modelling Grid Description

M. Telenta, L. Kos
Faculty of Mechanical Engineering, University of Ljubljana
Aškerčeva 6
1000 Ljubljana, Slovenia
marijo.telenta@lecad.fs.uni-lj.si, leon.kos@lecad.fs.uni-lj.si

Robert Akers and EU-MST1 Team *
CCFE, Culham Science Centre
OX14 3DB Abingdon, United Kingdom
rob.akers@ccfe.ac.uk

ABSTRACT

Visualisation of the scientific data is essential in analysing the results obtained from the fusion codes. The data, together with the mesh, is preferably stored in a standardised format. One step further is to use a common modelling grid description in which data from various fusion modelling codes, together with mesh, will be stored. In that sense, general grid description (GGD) is used as a format for storing data in the EUROfusion Integrated Modelling (EU-IM) database. OpenCASCADE CAD kernel is used to transform the CAD model into a triangular mesh for GGD storage in database. The ultimate goal of this work is to control the defeaturing of the CAD model applied to the desired level of detail (LOD) in order for it to be used as input for different meshing codes. In this paper, description is given of the grid generation using the open-source CAD-kernel Open CASCADE with the PythonOCC library. In addition, the process of storing the grid in GGD format will be explained. Finally, a ParaView plugin is shown for reading the data. The aim of the project presented in this paper is to automate the tedious task of CAD model defeaturing and meshing, which is often done manually, and to prepare it as a CAD service for use in scientific workflow engines in order to ensure data provenance.

1 INTRODUCTION

Workflow for the scientific visualisation of a CAD model requires a mesh format that can be read by general visualisation tools such as ParaView or VisIt. In addition, the mesh format is the favoured for being stored in a common fusion modelling grid description. The proposed workflow consists of defeaturing the CAD model, meshing it into a triangular mesh using PythonOCC library, and storing the mesh in EUROfusion Integrated Modelling (EU-IM) database using General Grid Description [1] (GGD) for the sole purpose of visualisation of the CAD model together with the results from various fusion modelling codes. GGD is primarily used by the EUROfusion task force for integrated modelling (EU-IM) as a common grid description for fusion modelling codes that interchanges the results on top of the EU-IM database. In order to store the CAD model, OpenCASCADE CAD kernel is used to transform the CAD model into a triangular mesh suitable for storage in GGD. The ultimate goal is to control the defeaturing of the CAD model applied to the desired level of detail (LOD)

*See http://www.euro-fusionscipub.org/mst1

such that the model could be used as input and meshed correctly by different meshing codes. In this paper, a review of the procedures for CAD defeaturing is discussed. CAD defeaturing consists of enveloping, removing, and simplifying unnecessary details. Also, the description of the GGD for visualisation and reuse by fusion modelling codes is presented. A PythonOCC library is used to script the OpenCASCADE kernel in Python programming language. For CAD data exchange, IGES and STEP standards are used and can serve as a common data format for input to meshing codes. If a "mesher" is unable to read the standard CAD format than this usually means that custom input needs to be prepared by a CAD kernel. The goal of the paper is to interface the CAD data to various meshing tools for use within EUROfusion. Meshing is then treated as a black box that outputs the mesh/grid as an input for physics codes. The output mesh, as well as, results of the codes on these grids are stored in a scientific database format. Ideally, it should be possible for all results to be stored in GGD format, as this enhances further processing and analysis by visualisation tools that are available within the EU-IM. Getting data into GGD format found in the EU-IM database means conversion of the results from other compatible formats.

2 BACKGROUND

The mission of our work is to contribute to a common interface for the transfer of the CAD data into *machine descriptions* for use by physics codes at several facilities (AUG, JET, MAST-U, TCV and W7-X) and in EuroFUSION work groups (WPMST1, WPCD, WPISA, WPS1) that may benefit from this work. In addition to that, several working groups for future devices are interested in a common interface between the 3D CAD models and the fusion modelling codes. There are cases where detailed 3D models are required to verify the source trajectories (from NBI) and the fast particles from the code (e.g. in ASCOT) in order to avoid hitting some vital components (e.g. cryogenics, first wall, holes, etc.). In such cases, CAD models need to be *reduced* to a reasonable resolution (size) to be used by the code.

Ray tracing from the plasma core centre through the detailed CAD model shown in Fig. 1 produced uniform quadrilateral structured mesh for the wall that is stored in EU-IM database where General Grid Description (GGD) is used as a structure for meshes of all kinds. Similarly, GGD is used for the *edge* Consistent Physical Object (CPO) for the SOL, with exception that here GGD consists of several *subgrids* (extended grids). Equilibrium CPO for the core rather uses simpler curvilinear grid for description. In order to simplify the GGD usage, the *Grid Service Library* was created. Storing the grids for use by physics codes in GGD format is one of the objectives of this work and will provide common ground for integrated modelling and analysis. Mesh generation using 3D CAD models for input is a problem (physics code) dependent. In many cases CAD models need to be *defeatured* by enveloping, removing and simplifying unnecessary details (see Figs. 2, 3) that can cause "bad" meshes and "hurt" the code. For *neutronics* simulations one needs to prepare a CAD model grouped according to the material definitions. To achieve a satisfactory input model, defeaturing of the CAD model (see Fig. 2) is performed as the first step and then generated mesh goes through *healing* process by fixing unnecessary gaps and overlaps.

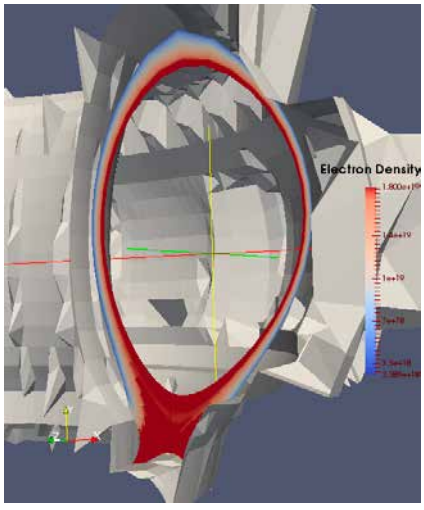

Figure 1: ASDEX Upgrade electron density from SOLPS code combined with ray-traced wall mesh. *edge* and *wall3d* CPOs are read with our ParaView source plugins.

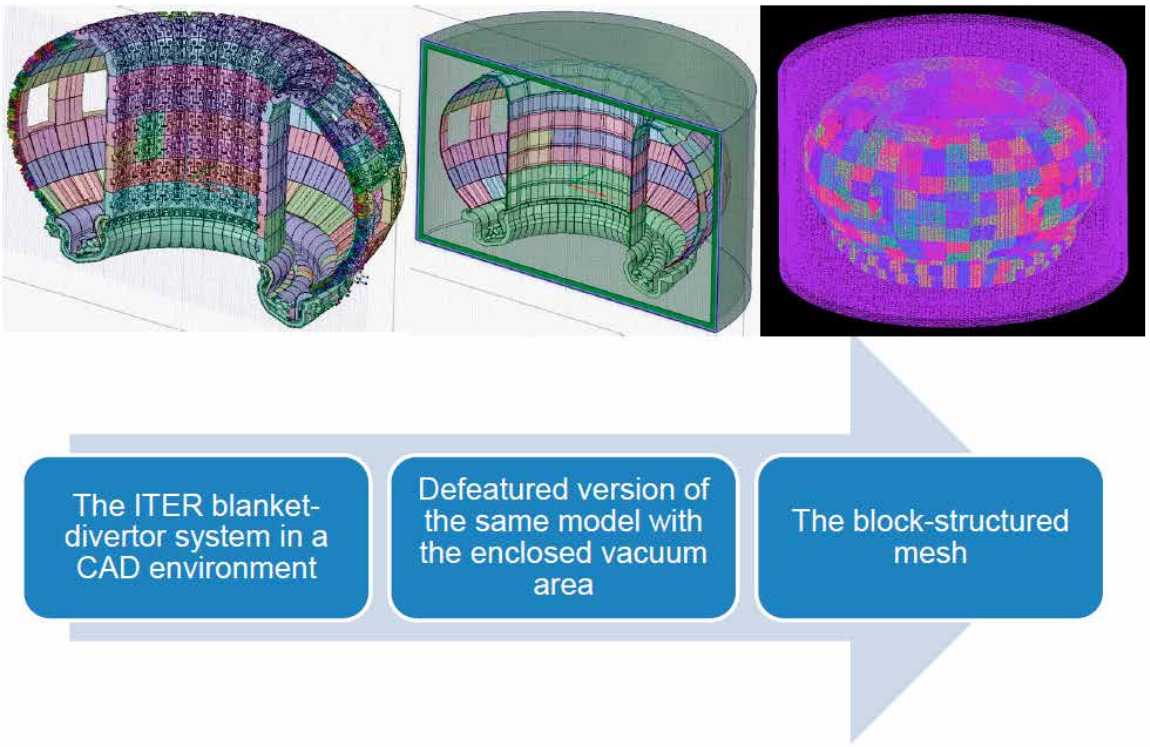

Figure 2: CAD defeaturing workflow with ITER blanket-divertor system. Models courtesy of Jason Hess, CCFE.

Fig. 3 shows CAD assembly from JET archive where parts can be grouped depending on the physics and material properties for the preparation of the MCNP model. Therefore, decomposition of the larger structures and analysing individual parts leads to homogenization of the geometry and materials. Note that mesh is not always the "end product" of defeaturing. Fig. 3 defeaturing results in a CAD model that is read by MCNP code's mesher directly. In addition to the above examples other possible areas would benefit from the common interface that provides workflows for processing CAD data into format adequate for meshing 3D structures. Clearly, for CAD model to be meshed "properly", at the desired level of detail (LOD), one need to be able to "automate" the tedious tasks, often done manually, in CAD modeller and CAD post-processing tools. Meshing itself depends on the physics code and often custom input and output formats are used by meshing codes. In an effort to provide common mesh description EU-IM uses GGD for storage and integrated modelling where output from one code serves (e.g *mesher*) as an input to subsequent code. For CAD data exchange, IGES and STEP standards exist and can serve as a common data format as input for *mesher* codes. If a *mesher* is unable to read the standard CAD format then this usually means that custom input needs to be prepared by a CAD *kernel*.

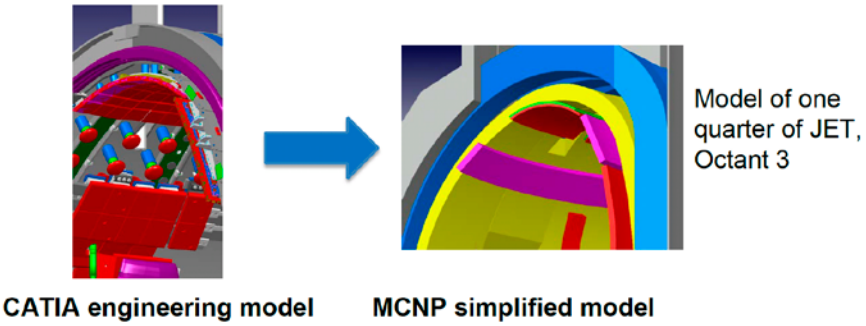

Figure 3: Manual CAD defeaturing for MCNP code conversion.

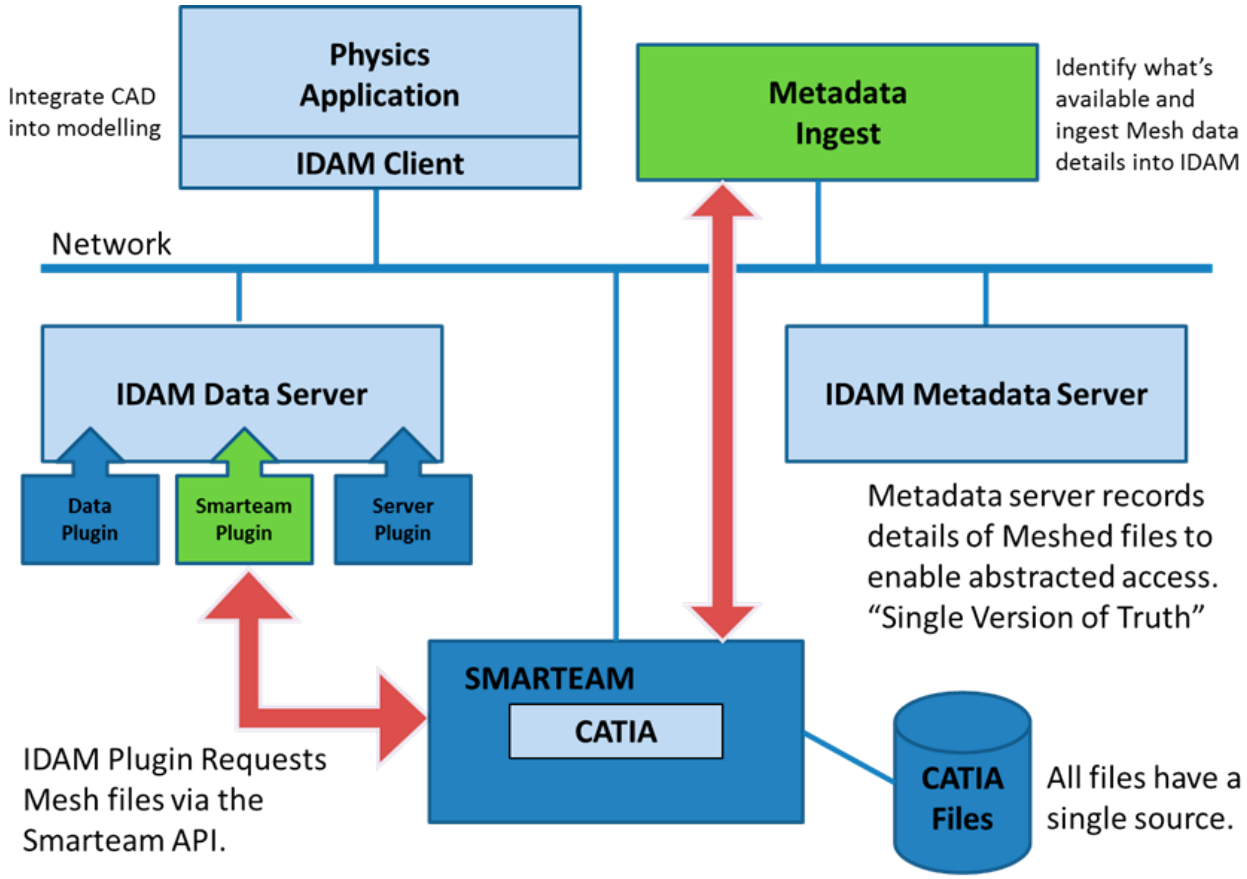

Figure 4: Integrated Data Access Management (IDAM). Courtesy of David Muir, CCFE.

3 OBJECTIVES AND RELATED WORK

Objective of this work is to build an interface for the CAD data to various "meshing tools" in use within the EUROfusion. Meshing will be treated as a "black box" that outputs mesh/grids as an input for physics codes. Output mesh, as well as results of the codes on these grids, is stored in some scientific database format such as MDSplus [2], HDF5 [3], Silo [4], NETCDF, ADIOS [5], GGD etc. The latter format is objective for storage whenever code output is used in EU integrated modelling framework. Ideally, all results should be able to be stored in GGD as this enhances further processing and analysis by (visualization) tools that are available within EU-IM. Getting the data into GGD format in the database EU-IM, in principle, means conversion of the results from other compatible formats. EU-IM database is accessed through Universal Access Layer [6] (UAL) library that handles low level operations and codes use UAL functions for input and output at corresponding time steps. Complete description of the database (data structure) is in XSD/XML format that allows several languages to be supported. For easier usage of the GGD data, Grid Service Library was developed in Python and Fortran. Currently C++ support for GGD is unavailable and is generally not needed for 3D scientific visualization tools (VisIt [7], ParaView [8]), while Python suffices for 2D plots used within ITMvis library [9] available in EU-IM. Not all CAD data is used as 3D. 2D cross-sections are in common use for tokamak geometry stored as a curvilinear grid rather than using GGD. Finally, the product would be to provide a GUI for workflows from CAD to physics code output and for visualisation. EUROfusion/WPCD and ITER/IMAS use Kepler as a scientific workflow engine. Effectiveness for interactive use of the Kepler engine and possible alternatives will be evaluated for use within this project. Fig. 4 outlines the IDAM [10] data flow in conjunction with

CAD database (e.g. CATIA SmarTeam PLM). IDAM is a *data aggregator* for unified (integrated) networked storage of data described within the IDAM Metadata server. Comparing EU-IM UAL and IDAM access layers shows that UAL emphasises data exchange for integrated modelling where similar codes must use the same data exchange structures, whereas IDAM provides a layer that unifies data exchange without prescribing exchange data structures. Switching similar codes within EU-IM Kepler workflows is easy by providing single *actor* per code whereas IDAM coupling needs one-to-one mapping between codes. IDAM can *ingest* CAD data easily and serve it unaltered over the network. EU-IM database can only ingest CAD data within *code parameters* placeholder in its XML described structure unless respective CPO is upgraded or created.

Much of the previous work on the defeaturing of the CAD geometry was done for tokamak wall data [11] and on the conversion of the CAD geometry models for neutronics analyses with the MCNP code [12]. In general, defeaturing simplifies meshing for simulation [13] by simplifying and providing simulation equivalent models that result in simpler meshes. Model equivalence is, of course, problem dependent and usually consists of removing small features (holes, fillets, chamfer, ribs, protrusion, etc.). Our work extends defeaturing of the CAD data for meshing and grid storage with a common fusion modelling grid description GGD.

4 INTERFACING CAD MODELS TO GGD

Intermediate objective presented in this paper skips CAD pre-processing, rather it concentrates on data-flow from CAD model to GGD mesh stored in EU-IM database. Immediate benefits of this objective are composite views for visualisation (e.g. Fig. 1) and direct import as a template for manual meshing. Further GGD mesh processing is possible too, depending on use case.

CAD model can be exported by several modelling software packages into general STEP format. For reading STEP and processing CAD description one needs access to CAD kernel routines in one of the programming languages. OpenCASCADE [14] is open source CAD kernel that offers flexibility and can potentially be used in defeaturing of the CAD models and mesh processing for fusion codes. Additionally, CAD clean-up and healing issues can be analysed and programmed at later stages as part of the batch defeaturing workflow. By using the open source CAD kernel OpenCASCADE, the STEP file is then exported in the most suitable form for reuse in downstream system or we can apply simple meshing algorithms inside CAD kernel directly to create simplex meshes such as triangles or tetrahedron in 3D-space. This is exactly the case within the present paper with objective to export mesh in EU-IM database where several CPOs use GGD for discretisation of space into subspaces (subgrids).

Since meshing itself is not our primary concern here, any meshing that can be used for visualisation is considered appropriate. *BRepMesh IncrementalMesh* class found in OpenCASCADE kernel builds the triangular mesh of a shape. Each quadrilateral of the input STEP object is divided into two triangles. While meshing, OpenCASCADE relatively enumerates each point of the surface. Hence a global list of unique points needs to be created when creating VTK [15] unstructured grid. Complete CAD to VTK conversion code written in Python wrapped OpenCASCADE kernel PythonOCC [16] works off-screen and is therefore useful for batch processing and heuristic control of meshing. We employed a Python dictionary to store globally all triangles point coordinates as keys and newly inserted point indexes as values. K-d tree structure might be used to speed up larger meshes instead of the hashed dictionary.

To visualise meshes in ParaView, the list of triangles is firstly written in VTK format and then saved/imported in *wall3d* CPO grid. Such an approach is useful for debugging 3D meshes. Choosing VTK file as an intermediate format simplifies initial interfacing for simple (non-composite) meshes

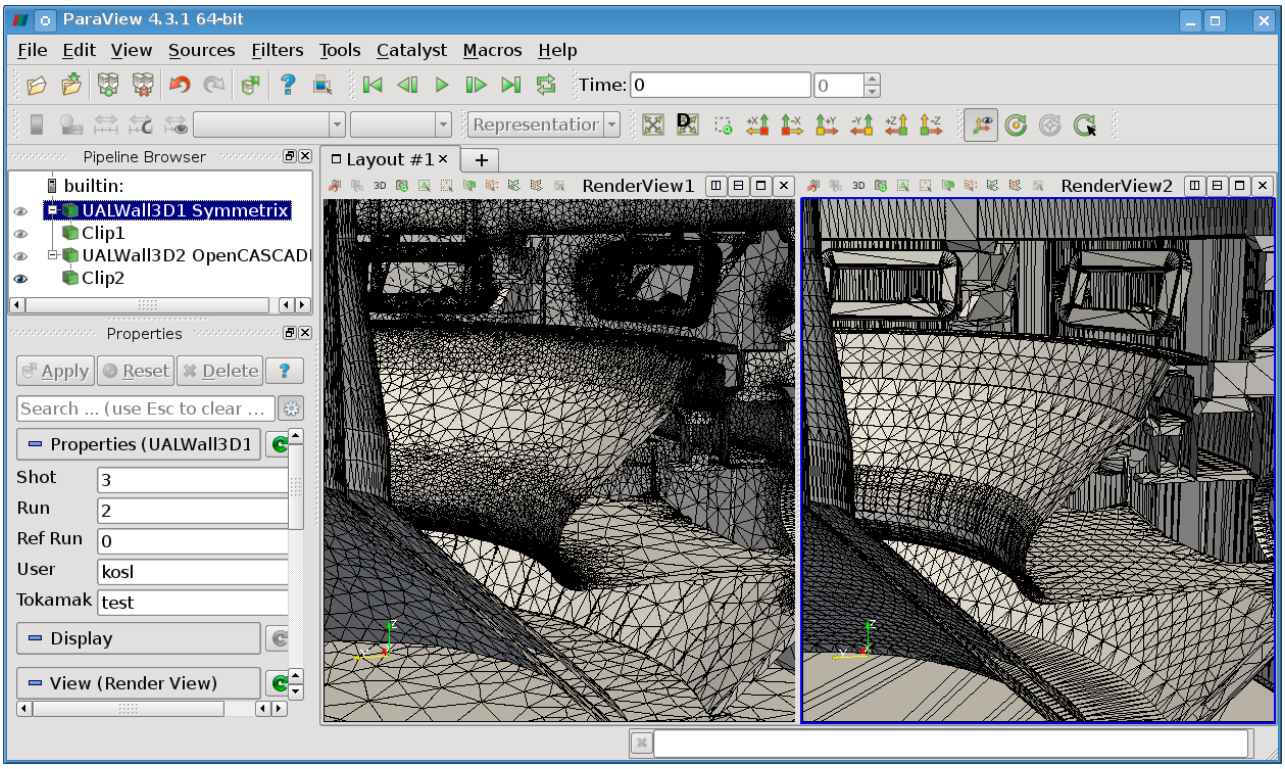

Figure 5: MAST-U divertor meshes imported with ParaView Wall3D UAL plugin from Symmetrix MeshSim and OpenCASCADE in RenderView 1 and 2 windows respectively.

as they can be directly read by ParaView. For multi-block VTK data-sets it is better to have direct control on the CPO's grid/subgrid composition in the EU-IM database. An example of a STEP model that was read, meshed, and written by use of OpenCASCADE and stored in wall CPO is shown in Fig. 5. ParaView *Source plugin* was developed [17] for UAL interfacing to a EU-IM database. In Fig. 5 two CPOs are imported by specifying shot, run, user, and tokamak selector. The mesh in RenderView 1 window is from MeshSim at scale 10 times bigger that mesh from our OpenCASCADE mesher in RenderView 2. Still, both windows can be scaled and linked together for visual comparison where one can notice that Symmetrix mesh was substantially cleaned before meshing and have curvature dependent density. OpenCASCADE mesh is smaller and have thin triangles that are useful for rendering only. However, we could increase density and even create our own meshing algorithm to be applied on each surface. Similarly to surface meshing we can apply volume meshing and obtain tetra-mesh.

5 CONCLUSION AND FUTURE WORK

Programming CAD kernel OpenCASCADE is an approach we took as a route that can be used for moving CAD/STEP data into GGD. Of course, taking the open source route is not something to take it blindly, but requires careful investigation based on past experiences in common interface problematics. Concrete insight into CAD to code interfacing from the complement point of view to the problematics of CAD healing is planned with CCFE that uses CADFix and SpaceClaim tools at the moment, and it often takes weeks or even months to prep a given CAD model for meshing. Even then, many of the meshers fail without further work (Symmetrix, ICEM etc.). CAD models at CCFE and meshes that need to dealt with in neutronics are very complex, and they are increasing the

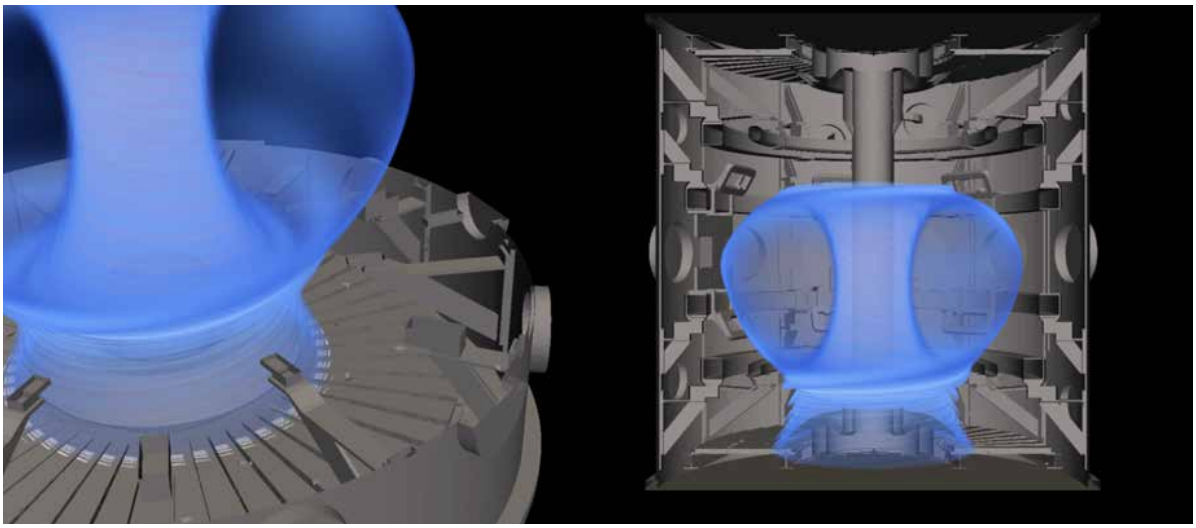

Figure 6: Model of MAST (Courtesy of Ivan Lupelli, CCFE)

complexity of load assembly models in equilibrium reconstruction, plasma facing components for power loading studies etc. Problems like building the pipe assembly under the ITER divertor dome in order to calculate power loading due to RMP driven losses took many months. Model of MAST in Fig. 6 has taken two months for a "new" engineer to be able to deliver a suitable mesh — for an experienced engineer, this might still take many days — not something that is easy to automate. That is what we will be challenged with when building workflows and developing GUI that automates procedures and records provenance. 3D visualisation tools (VisIt, ParaView) based on visualisation toolkit [15] (VTK) is ongoing part of the developed GUI along with common tools used by codes in workflows.

A wider overview and additional code support will be added with the inclusion of AUG/JET and/or other MSTs/W7-X. A collection of workflows and suite of tools for common interfacing of CAD to codes will be developed in the future. Commercial and open source libraries/tools will be assessed according to the analysis of relevance/benefits for specific tasks. Wherever possible, alternative solutions will be provided by allowing the addition and switching to different tools for comparison and replacement in a workflow. Data provenance within workflows will track data transformation for future referencing. While this work emphasizes GGD as mesh description, it should be noted, that we will look after supporting multiple mesh standards (as far as is practical), especially in cases where the engineering community can help deliver a more flexible product. Although ITER adopted EU-IM data structure, project will not focus strictly on EU-ITM/UAL (and IMAS/IDS), but rather look to deliver "internationally accepted infrastructure", i.e. supporting JT60-SA, MAST-U, JET, the MST1 machines, the US machines, KSTAR etc.

ACKNOWLEDGEMENTS

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No. 633053, task agreement AWP15-EEG-JSI/Telenta. The views and opinions expressed herein do not necessarily reflect those of the European Commission. Also, the authors would like to acknowledge Ivan Lupelli and David. G. Muir and Jason Hess from CCFE and Igor Lengar from JSI for helpful discussions.

REFERENCES

[1] H.-J. Klingshirn, *Adaptive grids and numerical fluid simulations for scrape-off layer plasmas*, Ph.D. thesis, Technical University Muenchen (2010), http://mediatum.ub.tum.de/node?id=970446.

[2] "MDSplus – data acquisition and storage tools," http://www.mdsplus.org (2015).

[3] "HDF – hierarchical data format," http://www.hdfgroup.org/ (2015).

[4] B. Whitlock, *Getting Data Into VisIt*, LLNL, Livermore, CA, 2nd ed. (July 2010), https://wci.llnl.gov/codes/visit/manuals.html.

[5] R. Tchoua, J. Choi, S. Klasky, Q. Liu, J. Logan, K. Moreland, J. Mu, M. Parashar, N. Podhorszki, D. Pugmire, and M. Wolf, "ADIOS visualization schema: A first step towards improving interdisciplinary collaboration in high performance computing," in *eScience (eScience), 2013 IEEE 9th International Conference on* (2013) pp. 27–34.

[6] F. Imbeaux, J. B. Lister, G. T. A. Huysmans, W. Zwingmann, M. Airaj, L. Appel, V. Basiuk, D. Coster, L.-G. Eriksson, B. Guillerminet, D. Kalupin, C. Konz, G. Manduchi, M. Ottaviani, G. Pereverzev, Y. Peysson, O. Sauter, J. Signoret, and P. Strand, "A generic data structure for integrated modelling of tokamak physics and subsystems," Computer Physics Communications **181**, 987–998 (2010).

[7] "VisIt official homepage," http://visit.llnl.gov/ (2015).

[8] U. Ayachit, B. Geveci, K. Moreland, J. Patchett, and J. Ahrens, "The ParaView visualization application," Enabling Extreme-Scale Scientific Insight(Nov 2012), ISSN 2154-4492, doi:10.1201/b12985-23, http:/dx.doi.org/10.1201/b12985-23.

[9] L. Kos, H.-J. Klingshirn, P. L. García Müller, F. Imbeaux, and EFDA ITM-TF contributors, "Unified approach to visualizations within the european integrated tokamak modelling framework," in *22nd Int. Conf. Nuclear Energy for New Europe* (Bled, Slovenia, 2013) pp. 1402.1–1402.8, http://lecad.si/~leon/research/itm-visualization.pdf.

[10] D.G. Muir, L. Appel, N.J. Conway, A. Kirk, R. Martin, H. Meyer, J. Storrs, D. Taylor, N. Thomas-Davies, and J. Waterhouse, "MAST's integrated data access management system: IDAM," Fusion Engineering and Design **83**, 406 – 409 (2008), ISSN 0920-3796, proceedings of the 6th {IAEA} Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, http://www.sciencedirect.com/science/article/pii/S0920379607005595.

[11] S. Akaslompolo, T. Koskela, T. Kurki-Suonio, T. Lunt, and E. Miettunen, J. Hirvijoki, *39th EPS Conference & 16th Int. Congress on Plasma Physics* **36F**.

[12] L. Lu, U. Fischer, Y. Qiu, and P. Pereslavtsev, "The CAD to MC geometry conversion tool McCad: Recent advancements and applications," in *ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method* (2015).

[13] W. R. Quadros and S. J. Owen, "Defeaturing CAD models using a geometry-based size field and facet-based reduction operators," in *Proceedings of the 18th International Meshing Roundtable* (Springer Berlin Heidelberg, 2009) pp. 301 – 318.

[14] "OpenCASCADE – Open CASCADE technology," http://opencascade.org (2015).

[15] "Visualization toolkit," http://www.vtk.org/ (2015).

[16] "PythonOCC website – Open CASCADE for Python," http://pythonocc.org (2015).

[17] L. Kos, J. Krek, M. Telenta, and EU-IM Team, "Visualisation of fusion related models stored in general grid description," in *Proceedings of the International Conference Nuclear Energy for New Europe* (Portorož, Slovenia, 2015) pp. 704.1 – 704.6.