Analysis of Computational Science and Engineering SW Data Format for Multi-physics and Visualization

Gimyeong Ryu[1], Jaesung Kim[1] and Jongsuk Ruth Lee[1*]
[1] Division of National Supercomputing, Korea Institute of Science and Technology Information
Daejeon - ROK
[e-mail: symsonic, jskim0116, jsruthlee@kisti.re.kr]
*Corresponding author: Jongsuk Ruth Lee

Received Septebmer 15, 2019; revised February 7, 2020; accepted February 17, 2020; published February 29, 2020

Abstract

Analysis of multi-physics systems and the visualization of simulation data are crucial and difficult in computational science and engineering. In Korea, Korea Institute of Science and Technology Information KISTI developed EDISON, a web-based computational science simulation platform, and it is now the ninth year since the service started. Hitherto, the EDISON platform has focused on providing a robust simulation environment and various computational science analysis tools. However, owing to the increasing issues in collaborative research, data format standardization has become more important. In addition, as the visualization of simulation data becomes more important for users to understand, the necessity of analyzing input / output data information for each software is increased. Therefore, it is necessary to organize the data format and metadata for the representative software provided by EDISON. In this paper, we analyzed computational fluid dynamics (CFD) and computational structural dynamics (CSD) simulation software in the field of mechanical engineering where several physical phenomena (fluids, solids, etc.) are complex. Additionally, in order to visualize various simulation result data, we used existing web visualization tools developed by third parties. In conclusion, based on the analysis of these data formats, it is possible to provide a foundation of multi-physics and a web-based visualization environment, which will enable users to focus on simulation more conveniently.

Keywords: computational science and engineering, EDISON, data format, metadata, visualization environment

A preliminary version of this paper was presented at APIC-IST 2019, and was selected as a best paper. This research was supported by the KISTI Program (No. K-19-L02-C05-S01), the EDISON Program through the National Research Foundation of Korea (NRF) (No. NRF-2011-0020576). A Grant was awarded by the Ministry of Science and ICT (MSIT) under the Program for Returners for R&D.

http://doi.org/10.3837/tiis.2020.02.024 **ISSN : 1976-7277**

1. Introduction

Recently, research and education services in high-performance computing and high-speed network-based cyber infrastructure environments have garnered attention [1]. In the United States, the Purdue University Network Computing Hubs project has evolved into the HubZero platform by pursuing simulation-based education and research convergence projects in a web environment, supported by the National Science Foundation. HubZero is a web-based platform that dynamically supports more than 60 computational science education and convergence research environments, and it is used by more than two million people annually in universities, research institutes, and industries [2][3]. The International Collaboration to Extend and Advance Grid Education is a multinational education initiative led by the European Union and provides a massive cyber infrastructure based on the Enabling Grids for E-science. College students and teachers develop a grid-based educational program and provide it through an open online counter [4].

In Korea, the Korea Institute of Science and Technology Information (KISTI) has been actively researching various application fields using super computer 5 (Nurion). A typical application is EDucation-research-industry Integration through Simulation On the Net (EDISON), a convergence environment platform for computational science engineering education research [5]. Similar to HubZero, EDISON provides Web-based computational science simulation and simulation learning tools beyond the theoretical education framework. Specifically, it is the only integrated Research and Development (R & D) platform in Korea that allows computational science and engineering researchers to receive one-stop services in a web-based environment such as simulation, data collection, storage, and Artificial Intelligence (AI) analysis by utilizing KISTI supercomputer and cluster resources. Hitherto, EDISON has developed 724 types of SW and 844 types of content in eight application fields (computational fluid dynamics, computational nanophysics, computational chemistry, computational structural dynamics, computer aided optimal design, computational medicine, urban environment, computational electromagnetics) and more than 10,000 people use it annually in 58 universities across the country.

As web-based computational science and engineering platforms have emerged and the number of users in various engineering fields is increasing, the perception of multi-physics, in which two or more physical systems are coupled with each other, is increasing [6]. However, these computational science platforms still focus on the completeness of each computational science simulation, with little concern regarding linkage. In addition, it is difficult to construct a sequential analysis and integration simulation environment because the format is not arranged during data exchange. In many cases, many software (SW) are not linked with proper visualization tools, resulting in an abstract analysis of interpretation results.

The purpose of this study is to analyze simulation data format and systematically organize input and output metadata for the standardization of the EDISON computational science and engineering simulation SW. In particular, the integration of input/output structure considering the interconnection of computational fluid dynamics and computational structural dynamics will be presented. In addition, data visualization through visualization tools will be implemented to reduce the gap between theoretical and actual physical phenomena.

2. EDISON: Computational Science Platform

2.1 Importance of Computational Science and Engineering

In line with global trends and national policy directions, the necessity for establishing and utilizing next-generation science technology education and a research convergence environment based on cyber infrastructure to maximize synergy through the interconnection between national science and technology R & D and higher education in science and engineering is increasing. To cope with rapidly changing technological changes, a flexible computational science education-research-industrial convergence platform that can replace high-risk and high-cost experiments is required [7]. Computational science and engineering (CSE) involves developing computational models and simulation techniques and applying them to specific fields [8]. CSE leverages computing resources to perform natural phenomena, engineering analysis, and design to respond to situations that are inaccessible or expensive experimentally. The typical methodology is modeling and simulation. Modeling includes algorithms from discrete or continuous problems, and simulation includes data analysis and visualization. Basic knowledge includes numerical computation, linear algebra, Fourier transform, optimization, and data science that derives useful information from large data. In addition, if high-performance computing is supported, existing significant problems or challenges can be solved [9]. However, most engineering and scientific simulation software are commercial software, and their usage is hindered because they are expensive. In addition, because most commercial software are finished products in the form of a package containing many unnecessary functions, it is difficult for small and medium-sized manufacturers to modify and set them up.

The current science and technology environment is in a transitional period where each discipline is mature to seek a new frame of mutual convergence. CSE is an academic or research field that can be created by the convergence of mathematics, computer science, and applied science and engineering; it is a field that must be fostered at the national level as it will enter the maturity stage after being developed [10].

2.2 Web-based simulation platform utilizing supercomputer resources

KISTI provides web-based education and research simulation services for problem-solving environments in CSE applications through the EDISON platform. Since 2011, the EDISON platform has provided an education and research convergence environment that can be easily accessed and used anytime and anywhere by utilizing KISTI's supercomputing resources [4]. EDISON is fostering advanced science and engineering in CSE by providing an environment in which research and problem solving can be performed immediately without expensive equipment. In addition, from the perspective of organic convergence and the virtuous cycle of science and technology research and science education, EDISON is laying the foundation for science and engineering students to become the next generation responsible for national science and technology competitiveness. As the EDISON platform can be used free of charge, it can minimize the waste of resources associated with the purchase of expensive commercial software used in universities. Additionally, the effectiveness of education can be maximized by using it in various universities that have insufficient budgets. **Fig. 1** shows the applicability of various research fields through the open computational science platform based on KISTI's peta-scale supercomputing resources.

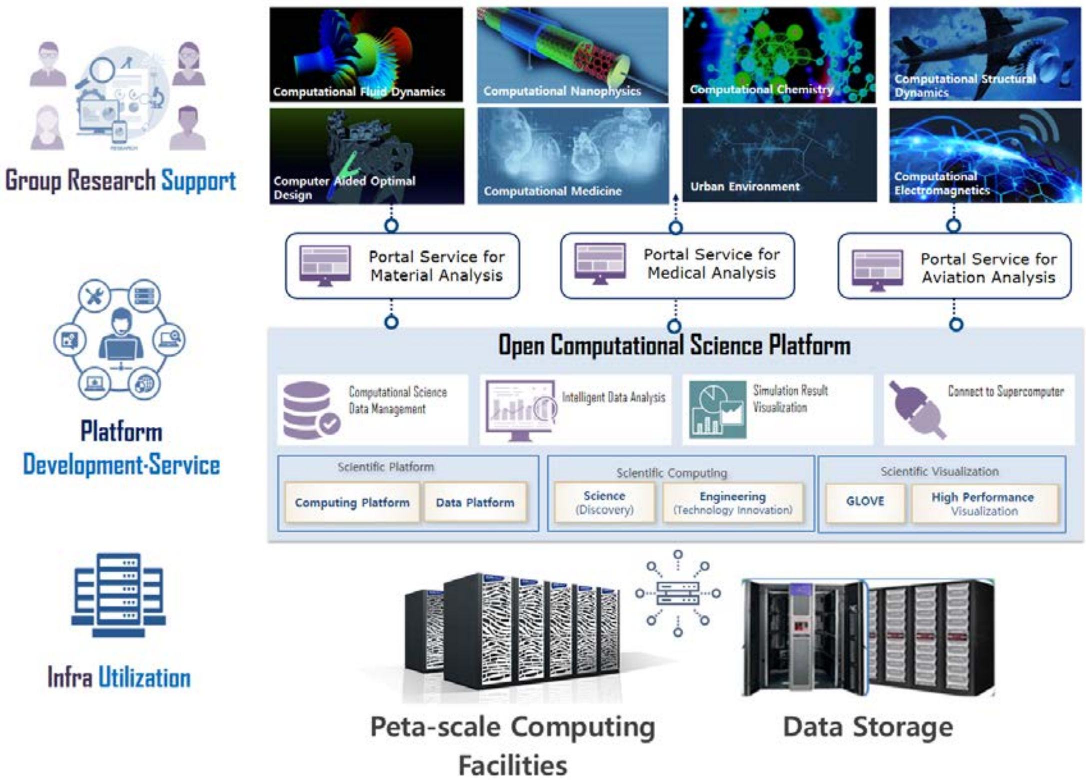

Fig. 1. Computational science platform (EDISON) overview

2.3 Weaknesses of EDISON platform

First, the number of web-based CSE platforms using open source has increased recently; furthermore, the number of users in various engineering fields is increasing, thereby increasing the awareness of multi-physics. Multi-physics systems refer to complex physical systems in which two or more physical systems are coupled to each other for operation [6]. EDISON is equipped with simulation software for different fields of CSE. However, the EDISON platform still focuses on the completeness of each computational science simulation, with little concern regarding linkage.

Next, visualization tools for simulation are lacing. For computational science researchers, visualizing input data (geometry, grids, etc.) and output data (contours, graphs, etc.) can be very intuitive. Visualization requires data I/O, visual transformations, and interactive rendering. Visualization is the process of extracting visual representations from raw data, which can be divided into scalar and vector data according to the dimensions of variables [11]. EDISON SW mainly uses a general-purpose format to visualize data, but in-house code SW uses a data format specific to a particular simulation analysis. The general-purpose format already knows the data structure, so it easily interacts with the visualization tool. However, some in-house code SWs require a conversion process to utilize them as a general-purpose format; therefore, it is necessary to understand the structure of the input / output data in detail. Recently, many open source software and APIs for graphic processing have been developed; therefore, it is important to obtain accurate information from this raw data.

Finally, the input/output format is not standardized during data exchange; thus, it is difficult to develop a sequential analysis and an integrated simulation environment. Most of the 724 simulation software currently registered in EDISON are the products of in-house code developed at each university laboratory; therefore, they contain different input/output information. When conducting simulations using one SW, I / O data type is not significant. However, if different SWs simulate in conjunction, the output of one simulation becomes the input of the next simulation. The EDISON platform provides a workflow environment for linkage between these SWs. As such, data format standardization is required for simulation-linked analysis through a workflow, and the analysis of SW data format is essential. In addition, data formats must be documented in advance for an efficient implementation because they are essential for executing visual transformations and user interface components.

3. Data format analysis and visualization strategy in EDISON

3.1 EDISON engineering fields

EDISON comprises various fields of computational science and engineering, and it is divided into engineering fields and science fields. The biggest issue of optimization through multi-physics analysis is in the field of mechanical engineering. Representatively, EDISON provides solutions to solve various engineering problems in mechanical, aerospace, architecture and civil engineering involving two fields: computational fluid dynamics (CFD) and computational structural dynamics (CSD). In addition, the pre-processing of CFD and CSD (geometry and grid generation) and the visualization of the simulation analysis results are very similar, so we will focus on the SW in these two engineering fields. CFD is used to discretize nonlinear partial differential equations and convert them into algebraic equations; subsequently, numerical methods are used to solve fluid flow problems. EDISON offers 103 CFD solutions to apply physical laws directly to real-world scenarios. It is used in engineering problems such as fluid flow and heat transfer in aircraft, ships, etc. CSD is a field that analyzes the loads and stresses of buildings and automobiles through finite element analysis. EDISON currently provides 80 solutions. **Fig. 2** shows an aerodynamic and turbine blade analysis environment using CFD and CSD in aerospace. The figure on the left is the main page of a specialized platform for flight aerodynamics. It is possible to perform hybrid RANS / LES analysis for flow analysis and unsteady flow analysis using various turbulence models and transition models. The figure on the right is an analysis environment for performing multi-physics analysis related to the fluid-structure-design of gas turbine blades.

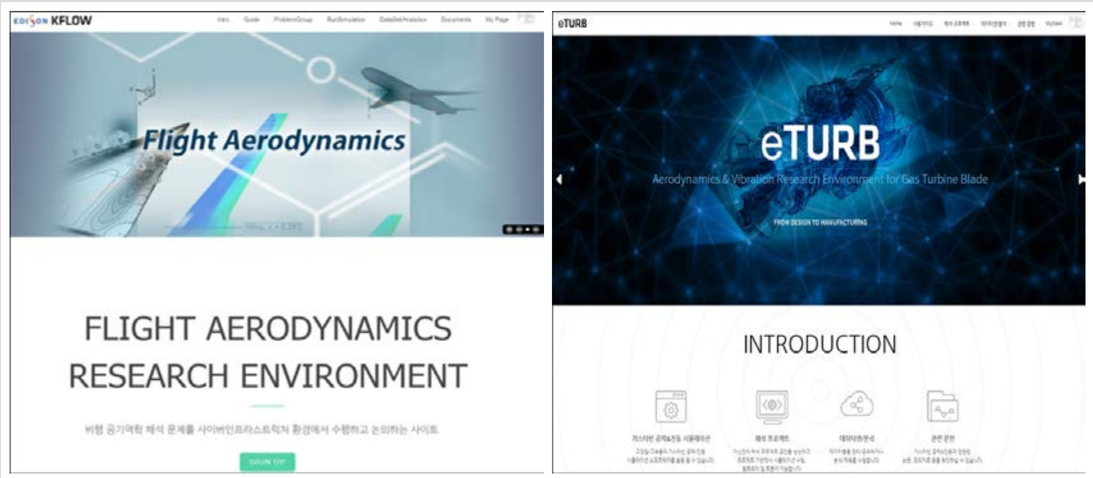

Fig. 2. Flight aerodynamics & gas turbine blade research

3.2 The necessity of data format analysis

In both CFD and CSD, an interactive environment is important. A CAD program is used to create shapes, and after a mesh generation operation to approximate a geometric area, a series of processes are performed to obtain results by numerical analysis. At this time, the user is in an interactive situation, in which the shape is modified and parameters are entered. In fact, many commercial SW (Fluent, Nastran, Simscale, etc.) provide a GUI environment by default as well as environments such as VR. When developing an environment that receives and processes user inputs, it is necessary to clearly know the metadata of the data input and output. In addition, data processing is important for implementing workflows for multi-physics simulations. A computational science workflow is the combination of multiple computational science SW for one purpose. In CSE, it is crucial to quickly predict the characteristics of various aerodynamics and structures; this requires considering a wide range of physical phenomena in engineering simulations and several fields are involved. EDISON has developed a workflow technology that can simultaneously simulate related natural phenomena [12]. To obtain the optimal analysis result through the workflow, it is essential to understand the input and output data information of different solvers. Finally, data analysis is required to implement a visualization environment within the EDISON workbench. Most commercial computational science SW visualize various simulation results on a single workbench screen. However, this visualization environment is difficult for EDISON. This is because most simulation software provided by EDISON have different data formats. Typically, it utilizes the general data format used in each research domain or develops and uses in-house code in its own laboratory. In EDISON, a visualization environment for research collaboration in different fields is required, and the analysis of input/output metadata of each simulation software is essential.

3.3 Representative SW in EDISON CFD & CSD

First, to select representative SW, we verified the utilization statistics of simulation software installed in EDISON. The EDISON platform supports university education by providing courses to the characteristics of eight computational sciences; furthermore, a contest is held annually to enhance the creative thinking and problem solving ability of science and engineering students. **Fig. 3** shows the number of users that use the simulation software of the EDISON platform for CFD and CSD in the past three years.

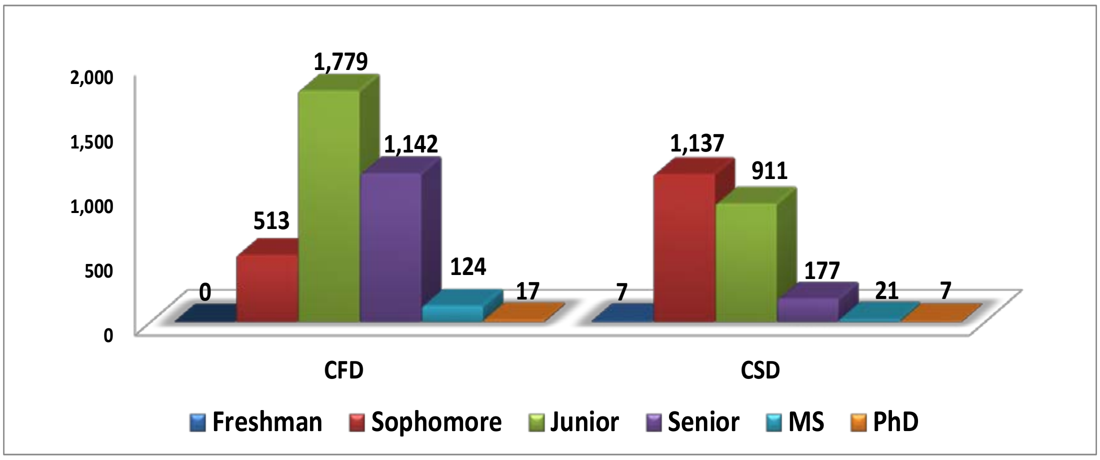

Fig. 3. 3 years user statistics for EDISON CFD and CSD

Over the last three years, 5,835 users have used the EDISON platform in 169 lectures and competitions in the fields of EDISON CFD and CSD. It is primarily used in the upper grades of universities as it is characteristic of the science and engineering field, and simulation learning through hands-on practice is meaningful only when it is in the upper grades where basic knowledge learned in lower grades is applied. In addition, the rewards of the competition are the largest motivator for the participation of undergraduates from many universities across the country. Master's and PhD students typically use the EDISON platform for the development and testing of simulation software in their fields, and most of the in-house source codes are registered during this process.

Next, based on the number of EDISON users above, the most used simulation SW statistics are summarized in **Fig. 4** below. Interestingly, the top five representative SW account for 84.9% of the total usage, based on the number of users in CFD over the past three years. Furthermore, in the field of structural dynamics, the usage statistics show that the top five SW constitute 72.9% of the total. Both CFD and CSD involve analysis procedures in the geometry creation, grid generation, analysis, and result visualization stages. As a small number of high-precision simulation SW are organically connected as the analysis process proceeds, it is necessary to focus on analyzing and linking data formats for representative SW.

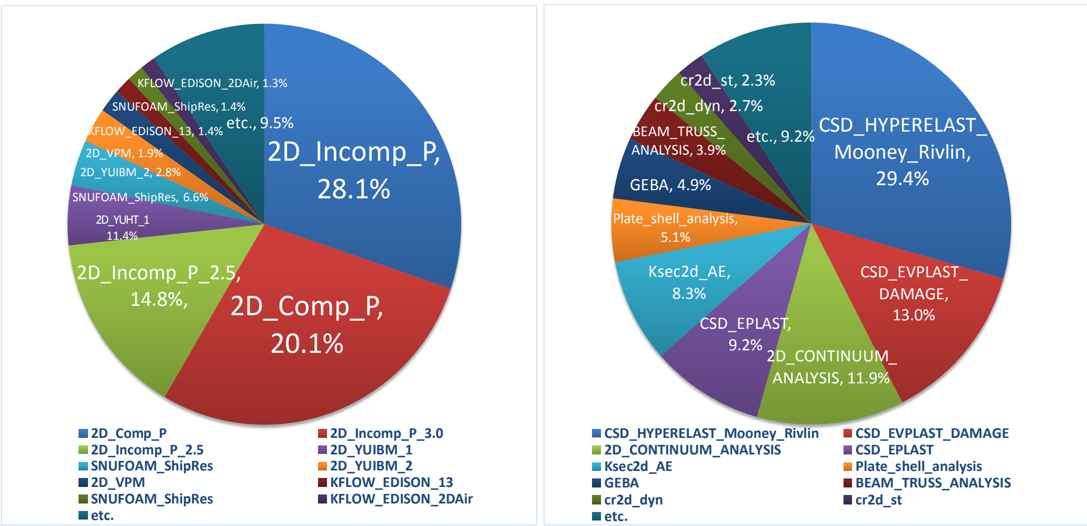

Fig. 4. Top 5 Simulation SW of EDISON CFD and CSD

3.4 Analysis of data format in EDISON CFD

EDISON CFD data files typically include shape files, mesh files, and analysis result files. The shape file is provided with two-dimensional (2D) point data format, STL and IGES. During preprocessing, grids are generated by SW such as eMEGA, Gridgen, Gambit, Pointwise, and OpenFOAM. Grid files are provided in Plot3D, OpenFOAM, TecPlot, and VTK formats. In particular, if a converter that converts the TecPlot format to the VTK format exists, it can be visualized as an open-source Paraview Glance (JS-based web-based Paraview). During post-processing, the simulation results are visualized by SW such as eDAVA, Paraview, Gambit, FLUENT, TecPlot, and Ensight. More than 70% of all 103 simulation software can be visualized with eDAVA (54%), Paraview (15%), and TecPlot (23%). eMEGA & eDAVA are pre/post-processing SW currently registered as standalone in EDISON. The top 10 CFD SWs account for 90% of the total number of simulation runs. The data formats for the top 10 SWs are shown in **Table 1**.

Table 1. Data format information of top 10 SWs of EDISON CFD

EDISON CFD SW	data formats
2D_Comp_P	*.msh (Fluent), *.p3d (Plot3D), *.dat (TecPlot)
2D_Incomp_P (3.0.0)	*.msh (Fluent), *.p3d (Plot3D), *.rlt
2D_Incomp_P (2.5.0)	*.msh (Fluent), *.p3d (Plot3D), *.dat (TecPlot)
2D_YUIBM_1	*.rlt, *dat (TecPlot)
SNUFOAM_ShipRes	*.VTK (Paraview)
2D_YUIBM_2	*.rlt, *dat (TecPlot)
2D_VPM	*.rlt, *dat (TecPlot)
KFLOW_EDISON_1	*.p3d(plot3D). *.dat (TecPlot)
SNUFOAM_ShipRes_ADV	*.VTK (Paraview)
KFLOW_EDISON_2DAir	*.rlt, *dat (TecPlot)

First, we analyze 2D_Comp_P and 2D_Incomp_P, which are used the most. As shown in **Fig. 5**, 2D_Comp_P is a CFD SW provided by EDISON and a finite volume method (FVM)-based 2D compressible flow analysis program. 2D_Comp_P uses a structured grid, and a multiblock grid and flow analysis can be used for complex shapes. Moreover, using the Reynolds averaged Navier–Stokes (RANS) equation as the governing equation, both invisible and compressible flows can be analyzed. From subsonic to supersonic, both viscous and nonviscous flows can be analyzed. Furthermore, in viscous flows, laminar flows and turbulent flows can be analyzed and various boundary conditions such as inflow and outflow boundary conditions, wall boundary conditions, and other boundary conditions are provided.

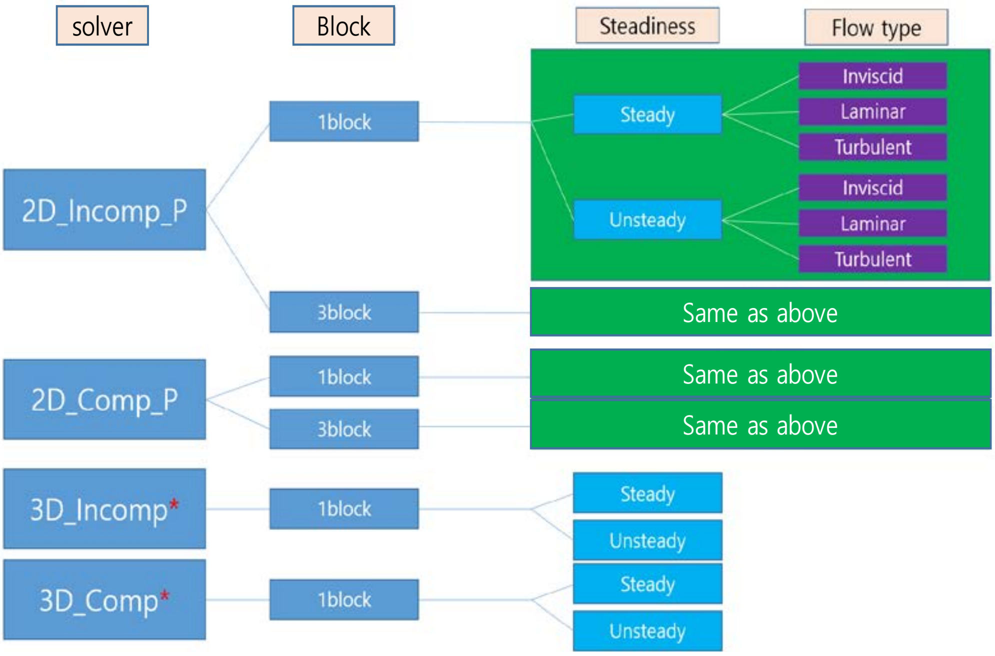

Fig. 5. Classification of compressible / incompressible flow analysis program in EDISON CFD

2D_Comp_P provides Roe and RoeM methods based on flux difference splitting and AUSM+ and AUSMPW+ methods based on the advanced upstream splitting method to calculate the numerical flux of the differential RANS equation. For the time forward technique, an Euler explicit method and the 3rd-order TVD Runge–Kutta method, which are explicit methods, are provided; additionally, the LU-SGS method, which is an implicit method, is provided. For turbulent flow analysis, the Spalart–Allmaras turbulence model and Chien's k-e and Menter's k-e shear stress transport models are provided. Same as 2D_Comp_P, 2D_Incomp_P is an FVM-based general purpose flow analysis program that analyzes incompressible flow. It is typically used to solve problems in the low Mach number region (Mach number 0.3 or less), in which the compressibility effect is neglected. In addition, Osher's upwind method is used to calculate the numerical flux of the differential RANS equation and provides Koren's limiter, which is a MUSCL-based limiter for higher-order spatial accuracy [13-17].

Because 2D_Comp_P and 2D_Incomp_P differ only in compressibility, we analyzed only 2D_Comp_P including an incompressible flow analysis. First, the input file format is divided into a grid file, a boundary condition file, and a flow condition file. Among these, the flow condition file is automatically generated as the parameter input on the EDISON workbench and passed on to the analysis program. The grid file and the boundary condition file are uploaded directly by the user. The grid file is in the Plot3D format in ASCII and has the “msh” extension. The file structure is shown in **Fig. 6**. NGrid means the number of grid blocks. Subsequently, the number of I- / J- / K-direction grid points of each block is followed, and the X, Y, and Z coordinate information is displayed. NX, NY, and NZ refer to the number of grid points in the I-, J-, and K-directions of the grid block, respectively; the X, Y, and Z coordinates refer to the x, y, and z coordinate values of the grid point, respectively. The x coordinate values of the entire grid points are first output, and then the y and z coordinate values of the entire grid points are sequentially output.

```
NGrid
NX (Block 1) NY (Block 1) NZ (Block 1)
NX (Block 2) NY (Block 2) NZ (Block 2)
…
NX (Block N) NY (Block N) NZ (Block N)
X(1,1) of Block 1 X(2,1) of Block 1 … X(NX,NY) of Block 1
Y(1,1) of Block 1 Y(2,1) of Block 1 … Y(NX,NY) of Block 1
Z(1,1) of Block 1 Z(2,1) of Block 1 … Z(NX,NY) of Block 1
X(1,1) of Block 2 X(2,1) of Block 2 … X(NX,NY) of Block 2
Y(1,1) of Block 2 Y(2,1) of Block 2 … Y(NX,NY) of Block 2
Z(1,1) of Block 2 Z(2,1) of Block 2 … Z(NX,NY) of Block 2
…
X(1,1) of Block N X(2,1) of Block N … X(NX,NY) of Block N
Y(1,1) of Block N Y(2,1) of Block N … Y(NX,NY) of Block N
Z(1,1) of Block N Z(2,1) of Block N … Z(NX,NY) of Block N
```

Fig. 6. 2D_Comp_P, 2D_Incomp_P grid block structure

The boundary condition file is in ASCII and has a "bc" extension. The file structure is shown in **Fig. 7**. The BC # of the IMIN, IMAX, JMIN, and JMAX line is the number of boundary conditions that enter the IMIN, IMAX, JMIN, and JMAX boundary, respectively. BC begin and end indexes are cell indexes at the beginning and end of the boundary condition, respectively, and BC type is the number separating the boundary conditions.

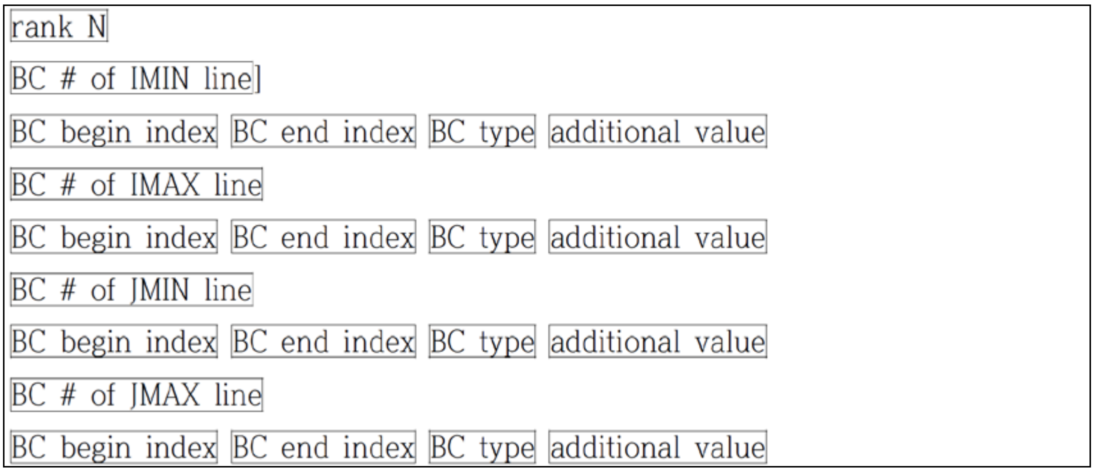

Fig. 7. 2D_Comp_P, 2D_Incomp_P grid boundary condition (BC) structure

Next, the output file can be divided into 2D flow field results, 2D graph results, and text data results files, as shown in **Table 2**. Data can be represented by numbers (integer or real) or strings. The 2D graph shows the x/y/iter/nontime on the X axis and the remaining variables on the Y axis. Typically, the name of the variable to be printed is shown at the beginning of the output file, and the data of each variable is output in order on one line. cf.rlt is a file that outputs the surface friction coefficient distribution. The friction coefficient according to the distance s is output based on the starting point of the surface having the wall boundary condition. coefhist.rlt is a file that outputs a history graph of aerodynamic coefficient values at each iteration step. The output aerodynamic coefficients are the lift coefficient, drag coefficient, and moment coefficient; additionally, the aerodynamic coefficients on each surface are output. cp.rlt is a file that outputs the surface pressure coefficient distribution. The x-coordinate and pressure coefficient of a surface with wall boundary conditions are output, where the pressure coefficient is output as a multiplied negative number. error.rlt is a file that outputs a density-based error residual strength graph. If the residual error is smaller than the convergence error set by the user, the interpreter concludes that the solution has converged and ends the calculation. force_com.dat outputs the coefficient of the aerodynamic coefficient that is calculated in steady flow analysis. The total lift/drag/moment coefficient and the lift/drag/moment coefficient on each side are output in order. The values on each side are output in the order of JMIN, JMAX, IMIN, and IMAX. time.dat is a file that outputs the time information required for simulation calculation and represents information such as total time and number of iterations. result.rlt / mid_result_######.rlt is a file that outputs the flow value of each grid point. In unsteady flow calculation, the intermediate calculation result (mid_result_######.rlt) is additionally output according to the output interval of the intermediate calculation result. Density (rho), pressure (p), mach number, velocity component (u, v) in the x-/y-direction, turbulent kinetic energy (k), rate of dissipation per unit turbulent kinetic energy (w), and eddy viscosity (vis) are also output. The flow field values except the x and y coordinates are dimensionless.

Table 2. Input / Output metadata in 2D_Comp_P, 2D_Incomp_P

Input metadata		
Data	**Description**	**Display**
Science App	2D_Comp_P, 2D_Incomp_P	Data
Flow Type	Laminar Flow, Turbulent Flow, Inviscid Flow	Data
Mach Number	A dimensionless quantity representing the ratio of the flow speed to the speed of sound.	Data
Renolds Number	A dimensionless quantity representing the ratio of inertial forces to viscous forces.	Data
Angle of Attack	The angle between the reference line of the body and the direction of motion.	Data
Steadiness	Steady flow / Unsteady flow	Data
Total Iteration	If the number of iterations exceeds this number, the computation will be terminated.	Data
Center Point for Moment	The reference position to calculate the moment coefficient	Data

Output metadata		
File name	**Metadata variables**	**Display**
cf.rlt	(Flow_type = Turbulent/Laminar Flow) x, y, cf (Flow_type = Inviscid Flow)	2D graph
coefhist.rlt	(Steadiness = Steady Flow) iter, cl, cd, cm, clim, cdim, cmim, clip, cdip, cmip, cljm, cdjm, cmjm, cljp, cdjp, cmjp (Steadiness = Unsteady Flow) nontime, cl, cd, cm, clim, cdim, cmim, clip, cdip, cmip, cljm, cdjm, cmjm, cljp, cdjp, cmjp	2D graph
cp.rlt	x, y, -cp	2D graph
Force_com.dat	(Steadiness = Unsteady Flow) Total cl, total cd, total cm (Steadiness = Steady Flow)	2D graph
error.rlt	(Flow_type = Turbulent Flow) iter, density (Flow_type = Inviscid/Laminar Flow) iter, density	2D graph
result_###*.rlt	(Flow_type = Turbulent Flow) x, y, rho, p, mach, u, v (Flow_type = Inviscid/Laminar Flow) x, y, rho, p, mach, u, v	Contour
mid_result_$$$$$$**.dat (Steadiness = Unsteady Flow) If steady flow, no file created	(Flow_type = Turbulent Flow) x, y, rho, p, mach, u, v, vorticity, k, w, vis (Flow_type = Inviscid/Laminar Flow) x, y, rho, p, mach, u, v, vorticity	Contour
time.dat	total time, total iteration, pre/post-process time, simulation time, simulation time per iteration	data

*###: block number, block
**$$$$$$: iteration number

3.5 Analysis of data format in EDISON CSD

The data files of EDISON CSD generally use the data format of commercial software. Particularly, EDISON CSD has its own pre/post-processor called CSD Pre/Post. The files output from different simulation software can be displayed as data, graphs, or analysis space visualizations with contour information. File formats exported from CSD-Pre include Nastran, CSD, and IGES, and file formats imported from CSD-POST include CSD, OP2, BDF, INP (Abaqus), and PLT. First, PLT is a file format containing plot information (*.plt), and OP2 and PCH are visualization file formats used by Nastran. BDF is a Nastran input file format that contains both geometry and grid information. ABAQUS INP is ABAQUS's preprocessing file format, and PLT is TecPlot's result file visualization format. **Table 3** shows the data format for the top 10 SW with high utilization [18][19].

Table 3. Data format information of top 10 SWs of EDISON CSD

EDISON CSD SW	data formats
CSD_HYPERELAST_Mooney_Rivlin	*.inp (ABAQUS input), *.plt (Tecplot)
CSD_EVPLAST_DAMAGE	*.inp (ABAQUS input), *.plt (Tecplot)
2D_CONTINUUM_ANALYSIS	*.pch (Nastran), *.op2 (Nastran)
CSD_EPLAST	*.inp (ABAQUS input), *.plt (Tecplot)
Ksec2d_AE	*.bdf (Nastran), *.ensi (paraview)
Plate_shell_analysis	*.pch (Nastran), *.op2 (Nastran)
GEBA	*.inp, *.dat
BEAM_TRUSS_ANALYSIS	*.pch (Nastran), *.op2 (Nastran)
cr2d_dyn	*.bdf (Nastran)
cr2d_st	*.csd, *.dat (Tecplot)

We analyzed CSD_HYPERELAST_Mooney_Rivlin, which is of high utilization like EDISON CFD. CSD_HYPERELAST_Mooney_Rivlin is a hyperelastic material behavior analysis program. The input file format is similar to that of the ABAQUS input. Hyperelastic materials describe material behavior in terms of strain energy per unit volume. The finite element method is used, and iterative calculation is performed using the Newton–Raphson method. Analysis results are generated in the text file and Tecplot file formats. The metadata information of the input and output files are shown in **Table 4**.

First, the input file is analyzed and input in the text form. HEADING defines the interpretation problem. Next, the material property information is input through MATERIAL according to the definition of super elastic behavior, and the element information is input through ELEMENT. Next, the definition and creation values of the constraints of the node are entered with SET DEFINITIONS and whether geometric nonlinearity should be considered is decided. In STATIC, static analysis definitions such as initial time increments and periods in one stack are entered. In BOUNDARY, boundary conditions are entered. Concentrated and distributed loads are distinguished using CLOAD and DLOAD, respectively. Finally, the value of a node or an element's variable is stored in the * .out file for each increment.

After analyzing the output file, a text file containing the input information and a text file in the Tecplot file format are output. In the PLT file, the maximum von Mises stress and the maximum displacement information such as the coordinate system, displacement, stress, equivalent plastic strain, and community integral are extracted and displayed as metadata.

Table 4. Input / Output metadata in CSD_HYPERELAST_Mooney_Rivlin

Input metadata		
Data	**Description**	**Display**
HEADING	Title of input file	Data
NODE	Specify nodal coordinates [Node number, X coordinate, Y coordinate, Z coordinate]	Data
ELEMENT	C3D4 - 4 node linear tetrahedron	Data
	C3D10 - 10 node quadratic tetrahedron	
	C3D8 - 8 node linear brick (Full integration)	
	C3D8R - 8 node linear brick (Reduced integration)	
	C3D20 - 20 node quadratic brick (Full integration)	
	C3D20R - 20 node quadratic brick (Reduced integration)	
	C3D27 - 27 node brick (Full integration)	
	C3D27R - 27 node brick (Reduced integration)	
MATERIAL	Definition of hyperelastic behavior of MOONEY-RIVLIN model	Data
Set definition and creation	NSET=NAME : Node Set	Data
	ELSET=NAME : Element set	
STEP, NLGEOM	Geometric nonlinearity	Data
STATIC	Initial time increment, Time period of the step	Data
BOUNDARY	Set boundary conditions	Data
CLOAD, DLOAD	Specify concentrated forces & distributed loads	Data
Output metadata		
File name	**Metadata variables**	**Display**
coordinate system	x, y, z	Contour
displacement	u, v, w	Contour
stress	$\sigma_{vm}, \sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{23}, \sigma_{13}$	Contour
stain	X	Contour
von Mises stress	Normal stress, Shear stress	Data

3.6. Flexible visualization tools

It is difficult to represent all types of data in one visualization tool. Another reason for analyzing the data format of EDISON SW is to provide flexible visualization tools for the increasing number of simulation software. For EDISON to be scalable, it is necessary to consider flexible data presentation methods that can effectively visualize the increasing number of SW. Recently, tools for visualizing standardized simulation data have been developed owing to the development of web technologies, and some are provided as open-source software. The use of open-source-based visualization tools developed by third parties in EDISON not only reduces the effort of simulation software developers, but also renders it easier for users to understand simulation data using various visualization tools. For various reasons, many EDISON SW do not use standardized data formats. We take advantage of ways to reuse existing web visualization tools developed by third parties to support various types of simulation data representations. EDISON-VIEW framework allows import of web-based visualization tools developed by third parties, allowing each tool to be added, removed and updated. Therefore, we analyzed the data format and developed the mapping and conversion code for each software, as shown in **Fig. 8**. Each of the following figures explains the sequential process of using jsmol components to visualize material properties and molecular structure, and shows the definition of a visualization component, I/O component and execution code, and the process of importing that component.

```
var componentsGroup = Vvveb.ComponentsGroup['Library'] || [];

//adding components name
componentsGroup.push('Library/atomtransistor');
Vvveb.ComponentsGroup['Library'] = componentsGroup;

Vvveb.Components.extend("_base", "Library/atomtransistor", {
    name: "atomtransistor",
    attributes: ["data-component-atomtransistor"],
    image: "icons/custom/icon_20_atomtransistor.svg",
    dragHtml: '<img src="' + Vvveb.baseUrl + 'icons/chart.svg">',
    html: '<div data-component-atomtransistor class="atomtransistor" style="width:800px; height:700px; background-color: #eeeeee;" \
            data-file-type="{{finalPath}}" data-file-path="/Structure_V_R.js">\
                Atom Transistor is visible when you preview\
            </div>',
```

Part 1: Define Component

```
    dragStart: function (node)
    {
        var body = Vvveb.Builder.frameBody;

        if ($(".atomtransistor-script", body).length == 0)
        {
            $(body).append(
                "<script class='atomtransistor-script' type='text/javascript'>\
                $(document).ready(function() {\
                    $('.atomtransistor').each(function (index) {\
                        var self = $(this).attr('id','atomtransistor_'+index);\
                        if(!validate(self)) return true;\
                        var width = self.width();\
                        var height = self.height();\
                        var path = encodeURIComponent(this.dataset.fileType + this.dataset.filePath);\
                        var html = '<iframe src=\"/SDR_base-portlet/designer/plugin/components-atomtransistor.jsp?path='+path+'&width
                        $(this).html(html);\
                    });\
                });\
                </script>"
            );
        }
```

Part 2: Define I/O of Component and Execute Code

```
        return node;
    },
```

```
<!-- custom plugin -->
<script src="plugin/plugin-iframe-connector.js"></script>
<script src="plugin/plugin-manager.js"></script>
<script src="plugin/components-layout.js"></script>
<script src="plugin/components-metadata.js"></script>
<script src="plugin/components-jsmol.js"></script>
<script src="plugin/components-paraview.js"></script>
<script src="plugin/components-pdf.js"></script>
<script src="plugin/components-epub.js"></script>
<script src="plugin/components-chart.js"></script>
<script src="plugin/components-rlt2chart.js"></script>
<script src="plugin/components-html.js"></script>
<script src="plugin/components-x3dom.js"></script>
<script src="plugin/components-p3d.js"></script>
<script src="plugin/components-ngl.js"></script>
<script src="plugin/components-text-viewer.js"></script>
<script src="plugin/components-onedviewer.js"></script>
<script src="plugin/components-sc3dviewer.js"></script>
<script src="plugin/components-atomtransistor.js"></script>
<script src="plugin/components-csv-viewer.js"></script>
```

Part 3: Import as a Component

Fig. 8. Mapping and transformation work for each software for visualization of simulation data

Representative CFD and CSD use cases are shown in **Fig. 9** and **Fig. 10,** respectively. First, meaningful information was extracted from the result generated by the simulation process of EDISON and converted into a standardized file. The libraries required for execution were imported from third-party developers' open-source libraries, and the executable code was written based on JavaScript. Subsequently, standardized files were visualized through a lightweight JavaScript-based Paraview glance.

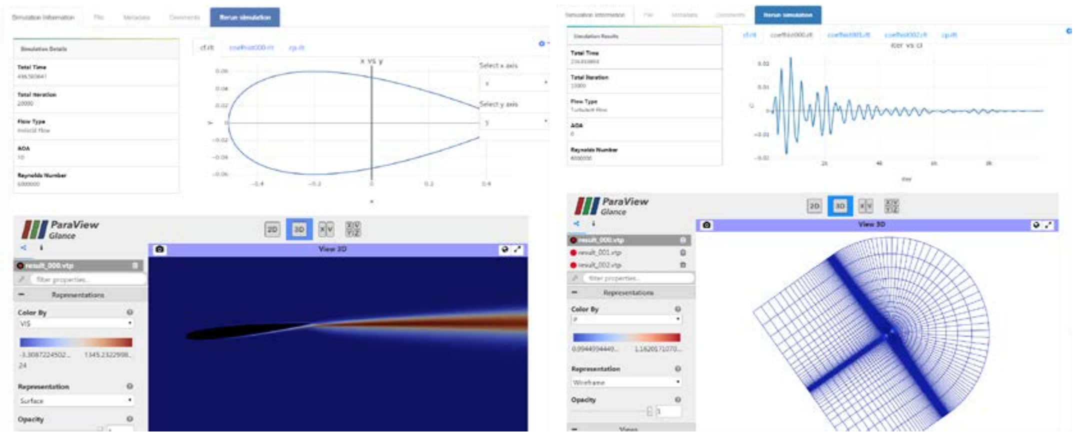

Fig. 9. Visualization of fluid flow in airfoil and aerodynamic analysis.

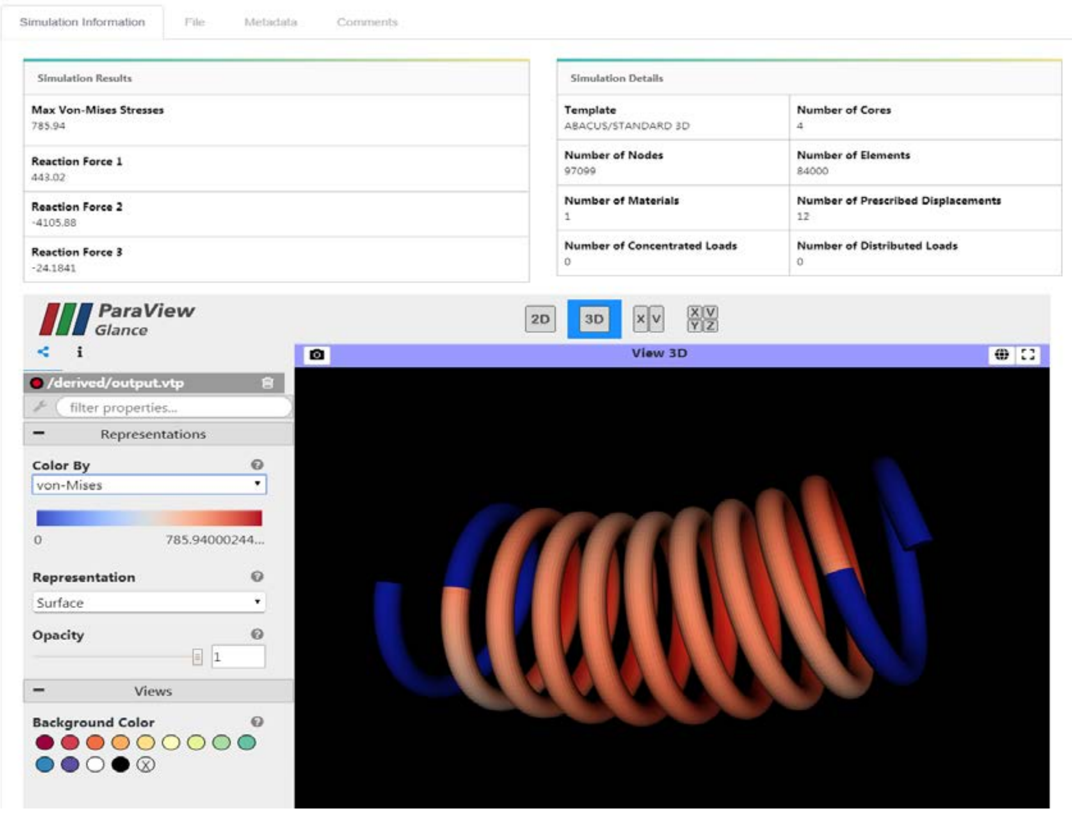

Fig. 10. Visualization of automotive suspension spring analysis

4. Conclusions

EDISON has been providing web-based simulation platforms for nine years since 2011. Focusing on providing a robust simulation environment and a variety of computational science software, it lacked support for fusion research and multi-physics. In addition, there was a lack of visualization of the simulation data for the user to easily understand. We herein analyzed the data format based on simulation SW that is the most used in EDISON and summarized the metadata. First, we analyzed the simulation software of EDISON CFD and CSD where several physical phenomena (fluids, solids, etc.) occur. 10 types of representative software are summarized according to their utilization, and metadata is analyzed based on input / output information of some software. Second, visualization tools that could visualize the results of computational science simulations were summarized and installed on the EDISON platform for users to use easily. It is difficult to link different research fields or different SW development types. In research and industry, demand for multi-physics is increasing, and efforts to complement each other's simulation analysis results are in progress. Hence, we must accurately analyze the simulation data for different SW, share the I/O data, provide the optimal environment for the workflow, and expect EDISON's representative SW to facilitate such a situation.

Acknowledgement

This research was supported by the KISTI Program (No. K-19-L02-C05-S01), the EDISON Program through the National Research Foundation of Korea (NRF) (No. NRF-2011-0020576). A Grant was awarded by the Ministry of Science and ICT (MSIT) under the Program for Returners for R&D.

References

[1] Suh Young-Kyoon and Cho Kum Won, “Construction and Service of a Web-based Cyber-learning Platform for the Computational Science and Engineering Community in Korea.” *Journal of Internet Computing and Services (JICS)*, vol. 17, no. 4, pp. 115–125, Aug. 2016. Article (CrossRef Link)

[2] M. McLennan, R. Kennell, “HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering,” *Computing in Science & Engineering*, vol. 12, no. 2, pp. 48-53, Mar. 2010. Article (CrossRef Link)

[3] HubZero. Available from: https://hubzero.org, accessed 23 August 2019.

[4] Jin Du-Seok, Jung Young-Jin, Jung Hoe-Kyung, "EDISON Platform to Supporting Education and Integration Research in Computational Science," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 16, no. 1, pp. 176-182, Jan. 2012. Article (CrossRef Link)

[5] EDISON. Available from: https://www.edison.re.kr, accessed 23 August 2019.

[6] Yoon Gil Ho, "Structural optimization and numerical analysis of multiphysics system," in *Proc. of the Computational Structural Engineering Institute Conference*, pp.157-160, 16 April, 2009. Article (CrossRef Link)

[7] Jin Ma, Sung Chan Park, Jung Hun Shin, Nam Gyu Kim, Jerry H. Seo, Jong Suk Ruth Lee, Jeong Hwan Sa, "AI based intelligent system on the EDISON platform," in *Proc. of the 2018 Artificial Intelligence and Cloud Computing Conference (AICCC)*, pp. 106-114, Dec. 2018. Article (CrossRef Link)

[8] Adam Himes, "The Use of Computational Modeling and Simulation to Create Virtual Patients: Application to Cardiac Pacing and Defibrillation Systems," *Journal of Cardiovascular Translational Research*, vol. 11, no. 2, pp. 89-91, Apr. 2018. Article (CrossRef Link)

[9] Jeffrey C. Carver, Richard P. Kendall, Susan E. Squires, Douglass E. Post, "Software Development Environments for Scientific and Engineering Software: A Series of Case Studies," in *Proc. of the 29th international conference on Software Engineering (ICSE)*, pp. 550-559, May. 2007. Article (CrossRef Link)
[10] Ulrich Rüde, Karen Willcox, Lois Curfman McInnes, and Hans De Sterck, "Research and Education in Computational Science and Engineering," *Society for Industrial and Applied Mathematics (SIAM)*, vol. 60, no. 3, pp. 707-754, 2018. Article (CrossRef Link)
[11] Lee Joong-Youn, Kim Min Ah, Lee Sehoon, Hur Young Ju, "GLOVE: Distributed Shared Memory Based Parallel Visualization Tool for Massive Scientific Dataset," *KIPS Transactions on Software and Data Engineering*, vol. 5, no. 6, pp. 273-282, 2016. Article (CrossRef Link)
[12] HanGi Kim, Jongsuk Ruth Lee, YoungJun Park, KumWon Cho, "Design and Implementation of Workflow Execution Engine using Heterogeneous Computing Resources & Job Management Framework," *proceedings of the Korean Institute of Information Scientists and Engineers (KIISE)*, vol. 2015, no. 12, pp. 61-62, 2015. Article (CrossRef Link)
[13] F. R. Menter, "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *American Institute of Aeronautics and Astronautics (AIAA) Journal*, vol. 32, no. 8, pp. 1598-1605, Aug. 1994. Article (CrossRef Link)
[14] B. van Leer, "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, vol.32, pp. 101-136, Jul. 1979. Article (CrossRef Link)
[15] C.-W. Shu and S. Osher, "Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, vol. 77, pp. 439-471, Aug. 1988. Article (CrossRef Link)
[16] S. Yoon and A. Jameson, "Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations," *American Institute of Aeronautics and Astronautics (AIAA) Journal*, vol. 26, no. 9, pp.1025-1026, Sep. 1988. Article (CrossRef Link)
[17] M. M. Rai and S. R. Chakravarthy, "An Implicit Form for the Osher Upwind Scheme," *American Institute of Aeronautics and Astronautics (AIAA) Journal*, vol. 24, no. 5, pp. 735-743, May. 1986. Article (CrossRef Link)
[18] NSTRAN, Available from: https://www.mscsoftware.com, accessed 23 August 2019.
[19] ABAQUS, Available from: https://www.3ds.com, accessed 23 August 2019.

Gi-Myeong Ryu is a Researcher at the Center for Computational Science Platform of Korea Institute of Science and Technology Information (KISTI), South Korea. He received BEng, MEng degrees in Aerospace Engineering from Chungnam National University in 2013, 2016 respectively. At present, he participates in EDISON(ED-ucation-research Integration through Simulation On the Net) project, which is a web-based simulation service for education and research. His research interests computational science, aerospace and fluid analysis.

Jaesung Kim received his Master's degree in Computer Science from KAIST, Korea. He is a researcher at National Institute of Supercomputing and Networking, Korea Institute of Science and Technology Information (KISTI). At present, he participates in EDISON(ED-ucation-research Integration through Simulation On the Net) project, which is a web-based simulation service for education and research.His research interests are distributed computing, big data handling, and data platf.

Jongsuk Ruth Lee received her Ph.D. in Computer Science from Univ. of Canterbury, New Zealand. She is a principal researcher at National Institute of Supercomputing and Networking, Korea Institute of Science and Technology Information (KISTI) and an adjunct faculty at Univ. of Science & Technology of Korea. Her research interests are smart learning, parallel/distributed computing & simulation, and big data handling.