

A Scientific Workflow Service with Intuitive Problem Solving Environment for Aerodynamics

Seoyoung Kim*, Kyoung-a Yoon*, Yoonhee Kim**, Chongam Kim***

* ** Dept. of Computer Science, Sookmyung Women's University, Seoul, South Korea

*** Dept. of Mechanical and Aerospace Engineering, Seoul National University, Seoul, South Korea

{ssyyy77, yoonka, yulan}@sookmyung.ac.kr, chongam@snu.ac.kr

**Corresponding Author

Abstract— Designing a scientific workflow for a complicated scientific application is now a trend to make the management of the problem simple and reuse the experiment for other simulations. Scientific workflow automates a repetitive cycle of transferring data and performing complex operations in a straightforward way in order to make it easier for scientists. Therefore, scientists could focus on their research regardless of paying concern about management of computing.

In this paper, we propose a scientific workflow service for multi-stage aerodynamic application in PSE (Problem Solving Environment). This workflow service in PSE provides an integrated workflow construction facility to map an experiment flow with designing, execution, monitoring and visualization services within an environment. Furthermore the service enables a user to interactively control over scientific simulations while the computations are in progress. With the presented high-level front-end, scientists easily modify a flow of their experiment and extend their work with pre-existed work modules without risk of design errors. As a proof-of-concept, we target a science portal for solving a CFD (Computational Fluid Dynamics) application in Aerodynamics study and show the implementation of this proposed scientific workflow service in the portal.

Keywords— Scientific Workflow; Problem Solving Environment; CFD; High Throughput Computing; e-Science

I. INTRODUCTION

CFD (Computational Fluid Dynamics) [1] uses numerical methods to solve and analyse problems which involve fluid flows. It utilizes high-performance computers to perform the calculations required to simulate complex problems modelled by many equations. Accordingly, many computational tools are adopted in applications that involve complex data analysis and visualization steps. Due to diverse characteristics of such scientific application domains, it is considered reasonable fact that scientific workflow should be supported essentially. Scientific workflow automates a repetitive cycle of moving data and performing complex operations in a way to make it easier for scientists to focus on their research and don't need to pay concern about management of computing

In this paper, we propose a scientific workflow service for multi-stage aerodynamic application in PSE (Problem Solving Environment). This workflow service in PSE provides an integrated workflow construction facility to map an experiment

flow with designing, execution, monitoring and visualization services within an environment. Furthermore the service enables a user to interactively control over scientific simulations while the computations are in progress. With the presented high-level front-end, scientists easily modify a flow of their experiment and extend their work with pre-existed work modules without risk of design errors. As a proof-of-concept, we target a science portal for solving a CFD (Computational Fluid Dynamics) application in Aerodynamics study and show the implementation of this proposed scientific workflow service in the portal.

In this paper we discuss some related work in section 2 and explain life-cycle of the science experiment in section 3. In section 4, we introduce details of the proposed workflow service with PSE and internal processing framework of the system. We also show the implementation of the service alongside PSE, and in final section, we present conclusions and future works.

II. RELATED WORK

There exist a variety of projects for workflow system in particular for scientific applications. One of them is about development of workflow design environment with graphical user interface and the others are studies about workflow specification, grid portal system and so on. Although there are many diverse of projects, we discuss only two projects for development of workflow design environment with graphical user interface (GUI), Taverna and Kepler, and another one project which related with workflow specification, Pegasus

A. Taverna

Taverna [2] is scientific workflow management tool created in myGrid project [3]. It is designed to ease the use of workflow and distributed computing technologies with e-Science related with bioinformatics as a main focus domain. A feature in Taverna provides their own workbench which can be pointed at a single web service or a site with links to multiple web services and then use these in workflows. By the use of workbench, the workflow is authored and executed. And a generic workflow enactor interface is used to execute workflows where the users can view results as they come in. Most of the output data are in textual form but there is support for plugging in visualization tools to visualize data. This

Taverna project is currently targeting a wide range of sub-area within only biology.

B. Kepler

Kepler [4] is a cross-discipline project which aims to simplify access to scientific data and the analysis of the data. It utilizes Ptolemy-II [5] platform as an environment, it has extended Ptolemy II towards scientific workflows through adding support for web service invocations and access to grid resources. They defined some units called actors and the communication between actors and workflow execution is controlled by an object denoted as director. The graphical editor in Kepler uses a graph representation of the workflow where the nodes in the graph are actors and the vertices represent links between the actors. The main focus domains of Kepler so far has been in biology, ecology and astronomy, geology.

C. Pegasus

Pegasus [6] is one of the well-known projects for scientific workflow centered its management and execution. It is designed to cover a set of technologies that help workflow-based applications executed in a number of different environments including desktops, grids, etc. Pegasus-WMS from the project has been actively used in a number of scientific domains including astronomy, bioinformatics, earthquake science, ocean science. Similar with our proposed system, it exploits DAX description to specify both tasks and its relation in workflow. However, Pegasus-WMS supports only functions relevant to workflow such as workflow execution, design and middleware connection. It didn't support PSE together to serve straightforward environment.

III. LIFE-CYCLE OF SCIENTIFIC EXPERIMENT

To serve perfectly intuitive problem solving environment to scientists, we have defined general cycle of a whole simulation. The whole procedure is typically consisted of repetitive executions of multiple applications and decision steps in the intervals of executions. Each of the computation-intensive applications for numerical analysis is a set of parallel jobs that are internally cooperating in the execution of a parallel program. The whole processing of an experiment in numerical studies is categorized to three parts; Pre-defined

step to define several parameter values to be used in simulation from some conditions, 1st design step in which DOE (Design of Experiment) is supported, and 2nd design step as final step for performing delicate design process as well as visualization for creating graph. These processes performed in serial order are illustrated on Figure 1 and each step can be re-executed by decision in the intervals of each step. Each step includes multiple sub-steps which are simple function and various computations. We introduce these steps in detail as the follows.

Pre-defined step is involved with modeling an object to be designed and executing optimization for input file. In this step, user executes manifold computations to make more specific type of input file (refers to mesh file) where user configures and to examine suitability of them. File creation is occurred from the specified parameter values and that file is adopted as an input of the numerical computation. For computation, during this step, additionally, the user can selectively apply a part of computations to input file and get optimized input file as an output by repeating computations. By performing executions, it can provide both precisions of designing a complicated structure and test effectiveness to researchers. With decision of user, he can execute the next step using output files created from this step or re-execute this step by re-defining another input and parameter values.

1st design step (also denoted as Simulation step) is responsible for large-scale, iterative computations for various parameter sets with solvers that is chosen by user and also referred to DOE (Design of Experiment). At this time, there needs multiple parameter values for applying to same application. As shown on Figure 1, we serve Parametric Sweep study Service during this step. User can set parameter values or range values into a form shown on Figure 1 where initial and boundary condition are defined. As the parameter values are varied, user can get various computational results depending on changing condition on same solver only by putting some values. Although PSS (Parametric Sweep study Service) processes more complex than existing application does, it may contribute to HTC (High Throughput Computing) research by soothing the burden of configuring various problems all the time. Jobs which generated by parameter setting are submitted to computing resource, and researcher examines each result with convergence graph. In case of that

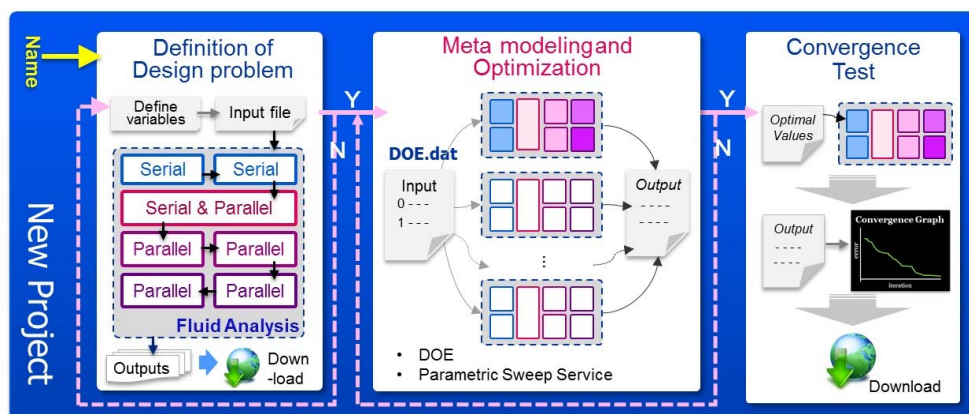


Figure 1. Life-cycle of Scientific Experiment

user wants to re-run with another parameter values, he might get back to the initial process of this step.

After finishing 1st design step, the *2nd design* as final step would perform computations using the former step's results. With showing the interim finding and performance status of these jobs, comparisons with graphs makes possible to complete a whole design-analysis process for aerodynamic research.

As reflecting the above steps to modules in our workflow design service, we enable that each module of workflow service can be re-usable and avoid wasting effort and time.

IV. SCIENTIFIC WORKFLOW SERVICE WITH PSE

A. Service Architecture

We have developed scientific workflow system that is based on grid portal. In this section, we introduce entire service organization of the portal system. As main services of this system, there are Workflow Service, PSS (Parametric Sweep study Service) and CFD (Computational Fluid Dynamics Service, Metadata Service. Figure 2 depicts the organizations.

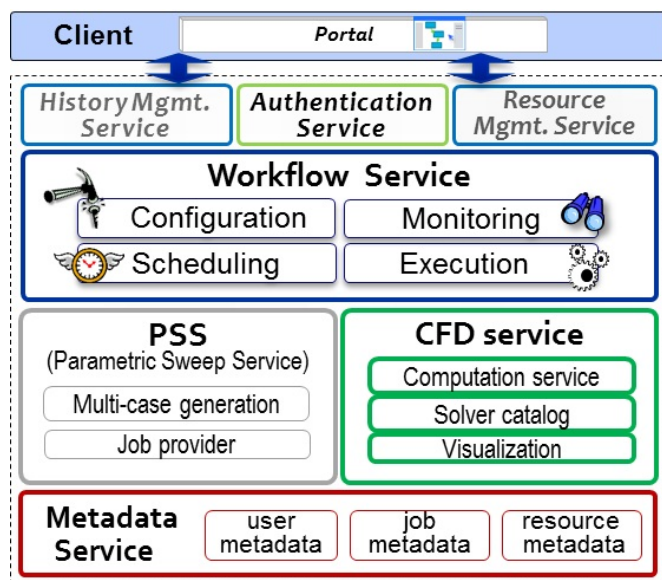


Figure 2. Whole Service Architecture

Workflow Service supports execution and management of entire workflow that is made of consecutive and parallel jobs as created execution procedure by user in visual workflow editor of portal system. Sub-workflows or a set of jobs are configured to a workflow in workflow configuration phase. In this phase, it usually interfaces with Metadata management service, so it allows users to manage history of workflow execution and to configure reformed workflow by enables users to re-load their works. In addition, the configuration phase deals with organizing DAX file for workflow description. Workflow execution phase means managing job execution using data which is generated by parsing the specified workflow file, DAX file. After then, workflow monitoring phase supplies some state information of jobs to user, and

determines whether user wants to proceed, or not. Moreover, the phase for monitoring serves state information of workflow execution in real-time.

PSS (Parametric Sweep study Service), as shown on Figure 2, is used for conducting multiple simulations with various input conditions and typically called during CFD service which will be discussed next paragraph. The PSS service is one of the most important services supporting HTC research. If user decides a parameter range from start to end, the service creates some sub-cases depending on the range. Then job provider assigns each job to available computing resources. After completing the jobs, Visualization CFD Service aggregates the job results. Completed results on each job are then aggregated in a graph by Visualization of CFD Service in order to plot the change of flow characteristics.

CFD (Computational Fluid Dynamic) Service is responsible for offering solvers referred to numerically analyses and design concerning with an aerodynamic problem. The CFD service supplies solvers composed of many computations that include a lot of complex operations. The CFD service is mainly used in two computational parts: pre-processing and simulation step. In the former part, mesh optimization for optimizing a set of input files is carried out and composed of various parallel programs. In simulation step, the optimized input files are computed using solver. The all of two parts deal with large number of numerical data and take a lot of times. After then, visualization tool is carried out to visualize result sets using embedded custom tool.

Metadata management Service exists with the above services. It covers experiment history, solver data, parsed data from workflow management phase and computing resources metadata and all of data which used in simulation processes on each user. In execution of workflow, it allows user to manage history of execution so that users can repeat certain workflow without reconstructing the same workflow. Moreover, this management service takes out output files from remote storage, so users can access output files of different simulations or workflows efficiently. With resources metadata, user can select the registered resources and can find their characteristics.

Interaction of these services, PSS, CFD, Workflow service, Metadata management service and basic additional services allows researchers, particularly scientists for aerodynamics to have their own intuitive PSE.

B. Workflow Execution Framework

As aerodynamics study is one of typical researches which rely on numerical modelling and analysis, workflow system is principal support since a large number of processes are included in an application. A diversity of applications for intensified scientific research has characteristics of complex, repetitive processes in different types. In this reason, when a researcher designs a simulation, it is required that he considers both of macroscopic and microscopic processes. To satisfy these conditions for design, we define a set of units for hierarchical workflow processing. Figure 3 depicts relations of the defined units and details are following. As shown in

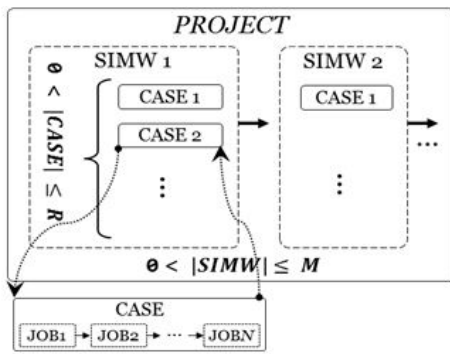


Figure 3. Relations of Units

Figure 3, *Project* > *SimW* > *Case* > *Job* are existed in a top-down order. Each low unit is thus considered a component of the high unit.

The units are classified into two kinds of type; Sequential (dependent) and Parallel (independent). ‘Project’ and ‘Case’ belong to parallel type since they can execute independently, ‘SimW’ and ‘Job’ are included in sequential type.

Once a user tries to create an experiment, a *Project* is created. In a project, there exists a set of steps (refer to Section 3) which might be composed with combinations of several steps. New project can be made any times if user wants and even though already a project is executing in that time. A *Project* can be broken into one or multiple *SimW*.

A *SimW* is corresponding to each step of simulation. As mentioned the Section 3, each step might be re-execute with new parameter values and all of steps are executed in fixed sequence. Moreover, in intervals of the steps user can control the flow by checking results. In these reasons, *SimW* is involved with Sequential type. *SimWs* corresponding three steps are identified by internally tagged namespace, and detailed processing are mentioned later, subsequent Section 4.C. The *SimW* is divided into one or multiple *Case*.

Unit for *Case* is used for HTC (High Throughput Computing) execution. According to various parameter values, a set of *Cases* are generated and each *Case* is executed independently. For example, if user inputs 0, *R*-1 as initial and boundary condition, the *R* numbers of cases are created. Within a *Case*, there included multiple *Jobs*. One *Job* can be mapped onto an instance which is executable application with parameter. Application to be executed in a *Job* might be serial processing or parallel processing. In our system, generally a set of *Jobs* are carried out in fixed order which is defined by user.

The hierarchical management according to the defined units enables user to control flow of scientific workflow which consisted in complicated process logics as well as to make dynamic workflow design.

C. Integrating Workflow Design and Execution

Figure 5 shows a procedure to execute a *Project*. Originally a workflow project formed by user is made of JSON [12] descriptions. Therefore it is required to translate to workflow representation which can express a flow between various tasks as well as easily interoperate with any of types. In our system, we adopt DAX [7] (DAG in XML) which is based on XML for DAG (Directed Acyclic Graph) and is the most commonly used for workflow system. All of information are extracted

from JSON and then formulated into DAX representations. The created forms are existed as many *SimWs* as user makes. It broke into multiple DAX forms according to *SimWs*. Each DAX can be identified by an embedded namespace, even if there are a great number of *SimWs* in a *Project*. Figure 4 shows a part of example for the description. Each an identified job within the example has a field named as ‘namespace’ and a value correspondently. If the value starts with ‘Pre’ as shown in the figure, it belongs to ‘Pre-defined Step’. A token along with the prefix, such as ‘f’ or ‘e’, is used to distinguish between simple utility functions, such as file uploading, and numerical computations. The only job having suffix ‘e’ can be translated into JSDL format. The converted JSDL [8], which is a kind of standard expression to describe the requirements for task submissions to computing resources, generates a set of jobs according to range of parameter. After that, the jobs are executed by middleware service. In our system, GRAM service is called to execute on grid resources with Globus Toolkit.

```
<job id="ID1001" namespace="Pre_f" name="def_prob" >
  <argument> 'AMACH: 0.73d0, RE: 6.5d6, AOA: 2.79d0,
  TOL:1.0d-4, TINF: 290.0d0,INTWRT2: 200, ITMAX2: 200,
  STEADINESS: 0'</argument> </job>
<job id="ID1002" namespace="Pre_f" name="upload_mesh">
  <argument>-o /storage/.../mesh/mesh.grd <filename
  file="mesh.grd"/> </argument>
  <uses file="mesh.grd" link="output" type="mesh"/></job>
<job id="ID1003" namespace="Pre_e" name="execute_pre">
  <argument> auto <filename file="mesh.grd"/></argument>
  <uses file="mesh.grd" link="input" type="mesh"/></job>
<child ref="ID1002">
  <parent ref="ID1001"/> </child>
  <child ref="ID1003">
  <parent ref="ID1002"/> </child>
</adag>
```

Figure 4. Example of a DAX Description

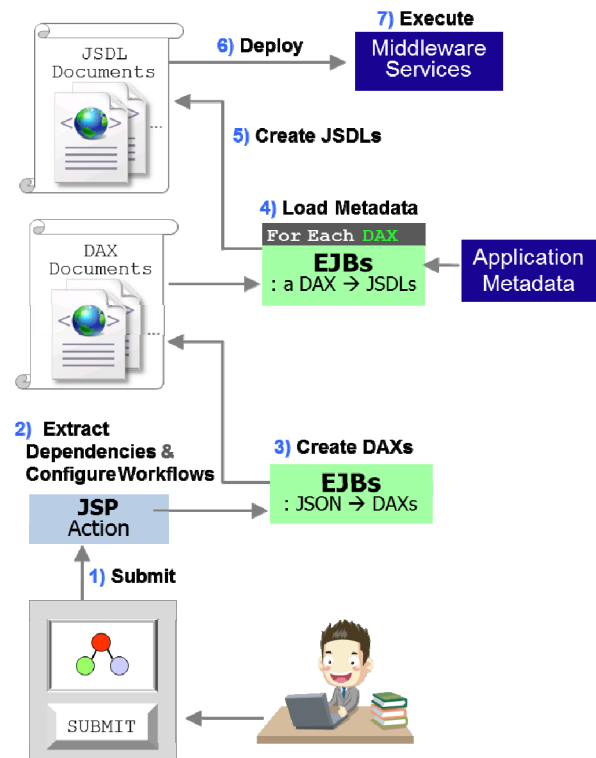


Figure 5. Procedure of a Workflow processing

V. IMPLEMENTATION OF SCIENTIFIC WORKFLOW SERVICE

For offering an integrated scientific workflow service with execution environment, we provide graphical editing service on the portal using Wire-It. Wire-It [9] is a JavaScript wiring library made of based on YUI [10] library and can create web wirable interface for dataflow applications, graphical modelling, graph editors, or visual programming. On the presented portal, visual workflow design tool can support graphical modelling flexibly as well as executing the workflow configured through modelling.

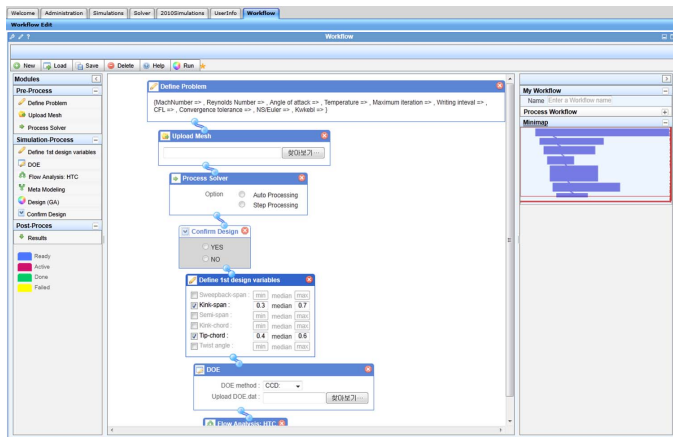


Figure 6. Workflow Service

The process of workflow design is comprised of Pre-defined step for modeling to analyze input files in advance, 1st design step and 2nd design step that execute the workflow operations and Optional step to apply workflow configuration selectively. Also each sub-step as each part building a workflow is defined as module and has two types of modules. One is the basic function type that will be processed within the portal and another is application type performed on

computational resource. The frame on the left side of the portal in Figure 6 is denoted as a Palette in which all of modules are listed and user can configure workflow model user want to design using the modules. In configuring the workflow, the modules can be dragged in the canvas on the middle and made of connection between two modules by clicking the module and dragging from an output port of one module to an input port of another. Modules connected on the canvas illustrated. Figure 6 are an example of a configured workflow model in which the module labeled 'Confirm Design' is option module and it allows user to can decide a direction of workflow execution according to previous result. While workflow is executed, user can check running state of workflow in 'Process Workflow' of screen on the right side of portal. After checking the result of Predefined step, if its result hasn't proper values user wants, 'No' option might be chosen by user. Then, progress of workflow is continued to 'Define Problem' module again for re-modeling workflow not next step module (Define 1st design variables).

By managing implemented job information, jobs comprised as part of 'Project' are consecutively processed by executing according to the order in defined workflow. As the workflow process, user can be provided with monitoring service in real time as well as checking the intermediate data during experiments. This is to verify experiments on an intermediate step in a progress to determine directions in research by checking its results. Furthermore, we utilized database system which is to store and update job's state information as interfacing with job manager. When starting job's execution, we update the state of job on every check point into database system. This progress is made over the local job manager, PBS [11] (Portable Batch Scheduler) or SGE (Sun Grid Engine), which sends measurement data to database system. To express job status on the screen, there presents in four state signs such as READY, ACTIVE, DONE and FAILED.

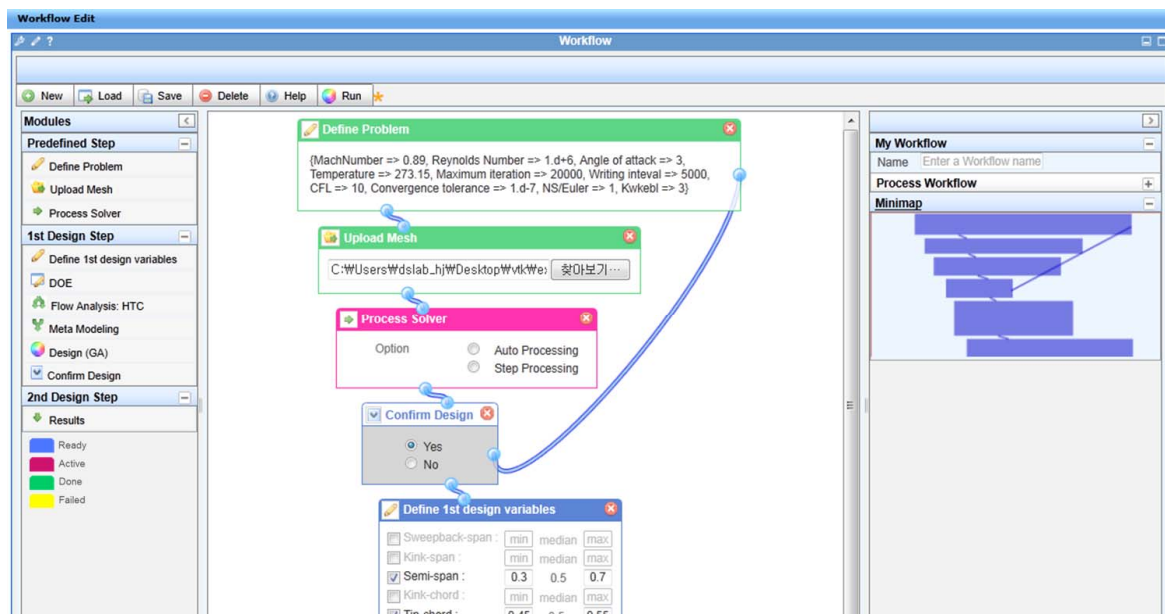


Figure 7. Workflow Status Monitoring Service

HTC Monitoring...										
C	v1	v2	v3	v4	v5	v6	Result1	Result2	Result3	Status
1	0	0	0	1	1	0	0.375	0.484	2.501	DONE
2	0	0	0	0	0	0	0.421	0.274	1.752	DONE
3	0	0	0	0	1	0	0.275	0.157	1.440	DONE
4	1	1	0	0	0	0				ACTIVE
5	0	0	0	1	1	0	0.721	0.043	1.184	DONE
6	0	1	1	1	0	0				ACTIVE

Figure 8. HTC Service

In 1st design step, cases(C) composed of multiple different parameter values (columns of first row, v1~v6) through PSS service process are executed in parallel on computational resources. Regardless of what case number is, monitoring information of cases is presented individually whenever the case is finished.

As following Figure 8, user can confirm not only case's results (Result 1~ Result N), but State of case. Besides, for more accurate analysis, user is provided to a compressed result's folder by clicking button below Down. In case of monitoring for whole workflow, namely it means 'Project' unit, monitoring service is provided by changing colors of each module and depicted in Figure 7.

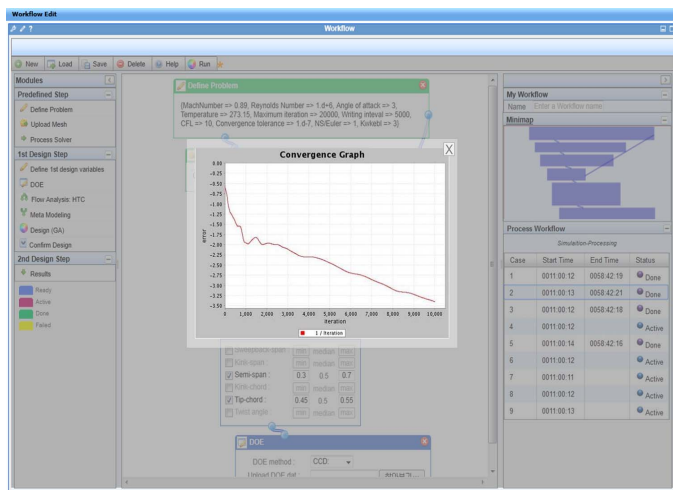


Figure 9. Graph support Service

On the last step of design, result values of the executed workflow are presented on a graph as illustrated Figure 9 and experiments is checked according to whether graph is converged or not. Through checking convergence graph, user can be offered to visualization service of each job and confirm if workflow is well designed or not. Likewise, through a high level user interface, like monitoring and visualization service, user can be provided with the workflow information in real

time without needing to know about an internal complex implementation. Therefore, users can take advantage of the environment to improve the productivity of study.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we describe a scientific workflow service on problem solving environment for Aerodynamic study and accentuate detailed execution framework of scientific workflow with an intuitive research environment for an experiment cycle. Also we demonstrate feasibility of the environment enables researchers to design a scientific workflow which is composed of several steps of an experiment, analyze and reason conducted computations, which are automated by the workflow service. By supplying an experiment service via a web portal, it makes users to carry out all steps of an experiment as a straightforward way wherever they connect to web. As the workflow service has been designed to provide controllable flow and interface in each step, user can lead experiment on his way. This portal promotes experts to improve their productivity of research in numerous applied science domains.

In near future, we plan to extend various experiment models and services considering meta-scheduling service as varying states of computing resources. Supplying such meta-scheduling for dynamic resource allocation in grid and cloud resource is considered for improving efficiency on an experiment

ACKNOWLEDGMENT

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No.2010-0027719).

REFERENCES

- [1] CFD, <http://www.cfd-online.com>
- [2] Taverna, <http://www.taverna.org.uk>
- [3] Stevens R, Robinson A, Goble C.A. 'myGrid: Personalised bio-informatics on the information grid.' Bioinformatics 2003; 19 (Suppl. 1) : i302-i30
- [4] Kepler project, <https://kepler-project.org>
- [5] Ptolemy II, <http://ptolemy.eecs.berkeley.edu/ptolemyII/>
- [6] Pegasus, <http://pegasus.isi.edu/>
- [7] DAX, Directed Acyclic Graph (DAG) in XML, <http://pegasus.isi.edu/wms/docs/3.0/perl/Pegasus/DAX/ADAG.html#name>
- [8] JSDL, Job Submission Description Language
- [9] WireIt, <http://neyric.github.com/wireit/>
- [10] YUI: Yahoo User Interface, <http://developer.yahoo.com/yui/>
- [11] PBS: Portable Batch Scheduler, <http://www.adaptivecomputing.com/research/docs/maui/pbsintegration.php>
- [12] JSON: JavaScript Object Notation, <http://www.json.org/>