



# Deploying Kepler Workflows as Services on a Cloud Infrastructure for Smart Manufacturing

Prakashan Korambath<sup>1</sup>, Jianwu Wang<sup>2</sup>, Ankur Kumar<sup>3</sup>, Lorin Hochstein<sup>4</sup>,  
Brian Schott<sup>4</sup>, Robert Graybill<sup>4</sup>, Michael Baldea<sup>3</sup>, and Jim Davis<sup>5</sup>

<sup>1</sup>Institute for Digital Research and Education, UCLA, <sup>2</sup>San Diego Supercomputer Center, UCSD,  
<sup>3</sup>University of Texas, Austin, <sup>4</sup>Nimbus Services Inc. McLean, VA, USA <sup>5</sup>University of California, Los Angeles, CA, USA.

*ppk@ucla.edu, jianwu@sdsc.edu, ankur\_kr@utexas.edu, jdavis@oit.ucla.edu,  
{lorin.hochstein,brian.schott,robert.graybill}@nimbisservices.com, mbaldea@che.utexas.edu*

## Abstract

21st Century Smart Manufacturing (SM) is manufacturing in which all information is available when it is needed, where it is needed, and in the form it is most useful [1,2] to drive optimal actions and responses. The 21st Century SM enterprise is data driven, knowledge enabled, and model rich with visibility across the enterprise (internal and external) such that all operating actions are determined and executed proactively by applying the best information and a wide range of performance metrics. SM also encompasses the sophisticated practice of generating and applying data-driven Manufacturing Intelligence throughout the lifecycle of design, engineering, planning and production. Workflow is foundational in orchestrating dynamic, adaptive, actionable decision-making through the contextualization and understanding of data. Pervasive deployment of architecturally consistent workflow applications creates the enterprise environment for manufacturing intelligence. Workflow as a Service (WfaaS) software allows task orchestration and facilitates workflow services and manage environment to integrate interrelated task components. Apps, and toolkits are required to assemble customized SM applications on a common, standards based workflow architecture and deploy on infrastructure that is accessible by small, medium, and large companies. Incorporating dynamic decision-making steps through contextualization of real-time data requires scientific workflow software such as Kepler. By combining workflow, private cloud computing and web services technologies, we built a prototype test bed to test a furnace temperature control model.

*Keywords:* Workflow as a Service, Smart Manufacturing, Cloud Computing, Kepler Workflows

## 1 A Comprehensive Smart Manufacturing (SM) Approach through Workflow as a Service (WfaaS)

Smart Manufacturing Leadership Coalition (SMLC) [1] identifies SM as the dynamic orchestration of manufacturing steps across different time constants and operational seams, including the entire value chain, without losing state of control. Figure 1 illustrates these seams at the micro level (people

and machines, machines to machines, and technology insertion in the manufacturing process), the meso level (cross system, vendor, and department decision making), and the macro level (cross factory, company, and supply decision making). These layers of operating seams reflect an overarching seam that exists among and between control and automation systems, and business and performance systems. An analysis of the interaction of these seams in a manufacturing operation, and alternatives to use information to stitch these seams together, provides insight into the challenges associated with SM Platform development. The center of Figure 1 depicts the seams created by compartmentalized decisions and actions with widely varying time constants. At the top, seams exist across the product design, planning, and manufacturing lifecycle which is trending into greater complexity with recycling and sustainability considerations. Through a workflow based framework, unique problem areas can be addressed through the orchestration of workflows that give data the necessary context in which actions can be executed manually or automatically in real-time to achieve a specific goal. SM workflows can interface with control and automation systems, and factory optimized sharable or proprietary workflows. The use of any models and simulations can be architected for computational tractability in a data driven workflow framework designed to address specific performance objectives. Workflow as a foundational construct is ideally suited to bridge seams such as vendor, department, company, complexity, or time. WfaaS can offer access to an appropriate mix of technology and infrastructure, applied at the time required for manufacturers building on regular cloud technology foundations. Extensive deployment of architecturally consistent workflow applications creates the enterprise environment for manufacturing intelligence. The workflow foundation defines responses to changes and readily extends to include intelligent process monitoring, fault diagnosis, procedural synthesis, and resilient systems as more sophisticated tasks in an orchestrated workflow. Thus, workflow allows the level of sophistication that can be readily established for the needs of specific business objectives and organization readiness. Workflow is foundational to enterprise comprehensiveness and is the general construct to accommodate time and action across the enterprise.

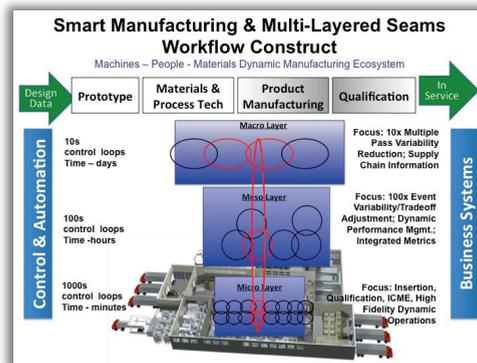


Figure 1: Smart Manufacturing and Seams of Opportunity

## 2 Orchestrating Workflow in a Shared Platform

The SM Platform in SMLC is an open architecture, and pre-competitive WfaaS software that facilitates a service-oriented and management environment to integrate the components using apps and toolkits. At its technical core the SM Platform defines how data is collected, shared and related, how computationally generated results interface with operating equipment and automation infrastructures, and how results are displayed in an actionable form to operators, engineers, and managers through dashboards. In terms of modeling, simulation, computation, and analytics, the SM Platform's code

management architecture is designed for time integrated sensor, data driven workflows constructed from Apps, each of which is software code that accommodates workflow interface specifications.

Workflow is defined as the orchestration of discrete tasks needed to get the necessary data, contextualize it, analyze it, and put it into actionable forms within a needed time window. As a foundational, collaborative technique in the SM Platform, workflow offers the following capabilities to manufacturing that are currently not available.

1. Workflow supports data-driven applications. Many workflows systems can describe both control and data dependencies among tasks in an application.
2. Workflow allows widely varying requirements on applying time constraints in a broadly multi-layered SM application environment to be managed by the needs of the workflow objectives.
3. Composition capability of workflow creates the possibility of a rich apps store for SM that contains not only component workflow apps but also a library of workflow sequences as well as entire workflow templates. Workflows can stitch data, information, models, metrics and other workflows together in an optimized way.
4. Workflow is an equally supportive foundation for product and process design, as well as engineering. Infrastructure and toolkits for managing product and process design, engineering, and rapid evaluation include integrated metric definitions, and rapid evaluation schema as well as work processes for integrating diverse expertise in feasibility testing and option selection.

### 3 SMLC Test Bed Use Case

We deployed a SMLC test bed to compute the optimum flow of combustible gases through Steam-Methane reforming furnaces to produce commercial gas ( $H_2$ ). The test bed makes use of Reduced Order Models (ROMs) derived from available Computational Fluid Dynamics models (CFD). For real-time analysis ROM is preferred over CFD calculations because CFD calculations are CPU intensive and take long time to compute. The inputs for the calculations are sensor data (images) from infrared cameras. They are analyzed using ROM scripts that runs using either Octave or Matlab software followed by CFD simulation for validation of results obtained from ROM analysis. Thus, the assembled workflow can be used for process control to prevent occurrences of unfavorable temperature distribution inside the furnace by suitably manipulating fuel flow rates or to find new operating regime for the furnace corresponding to higher energy efficiency while maintaining the reformer-tube wall temperature within its upper limit.

#### 3.1 SMLC Architecture

The SMLC architecture requires four principal components: 1) a user interface in the form of web portal, 2) REST service based WfaaS, 3) a scientific workflow software, 4) a cloud computing and storage infrastructure. In addition to these four components, SMLC also calls for an application store to share the workflows and other software packages among the users, the details of which are outside the scope of this paper. We have chosen Kepler software for scientific workflow mainly because its support for web services, grid, and cloud technologies [3,4]. The workflow is an essential tool in the implementation of automation, control, business management and dynamic actionable decision-making process through contextualization and analysis of the real-time data. WfaaS facilitates the service and real-time management by integrating all the components from user interface to running the application on the compute infrastructure, transferring the data among the resources and displaying the results to the user.

SMLC users interact with the run-time environment through a web portal. All the development work for running an application is done and tested in advance by SMLC software developers who have expertise in writing REST services, configuring workflows, deploying high performance

computing applications, and configuring and setting up a private cloud infrastructure. Additionally, the private cloud computing software package that we used, called OpenStack [5], will increase efficiency and productivity with low operating, maintenance and support costs, reduced downtime in terms of upgrading and applying patches and has a strong industry academia partnership. The capital investment on hardware is the only significant expenditure in the private cloud deployment used in this study.

### 3.2 Furnace Temperature Modeling using Kepler Scientific Workflow

The overall goal of this application is to improve the productivity through environmentally friendly technologies in the commercial manufacturing sector. In order to optimize real time production of combustible gases we run various established mathematical models using the real time parameters from the furnace. The computation process is done in on-demand basis on resources that are made available in a private cloud-computing environment built through Infrastructure as a Service (IaaS) model as well as storage as a service (SaaS) model. The movement of input and output data at various stages during the workflow is accomplished by configuring the Kepler actors. The private cloud that we built using the open source software OpenStack is deployed on the hardware running Linux based operating system (OS) and all the virtual images are based on Linux OS as well. We deployed the Kepler workflow system [6] for Praxair furnace modeling with a combination of REST based workflow service, web-based GUI at the front end and a Kepler workflow engine at the back end.

### 3.3 Kepler Workflow, Web Portal and REST based Web Services

The model we used is similar to the traditional workflow usage in high-performance computing where users can start the workflow from their laptop or desktop and the jobs are submitted to a batch scheduler on a remote cluster and the results are brought back to the laptop and displayed. The difference here is that the workflow is called by REST web services written using Django API. The web service needs to pass the workflow shown in Figure 2, the names of the input files and user defined scripts before it calls the workflow. The workflow’s inputs and outputs are listed in JSON format. Note that the input and output items might be different for different user inputs. The workflow will list all output files and list them into output.json file. An example input.json file and output.json file are shown in Figures 3 and 4 below. This model also involves a cloud storage built with OpenStack Object Storage service called swift. The details of the execution of this service are given in Section 4.

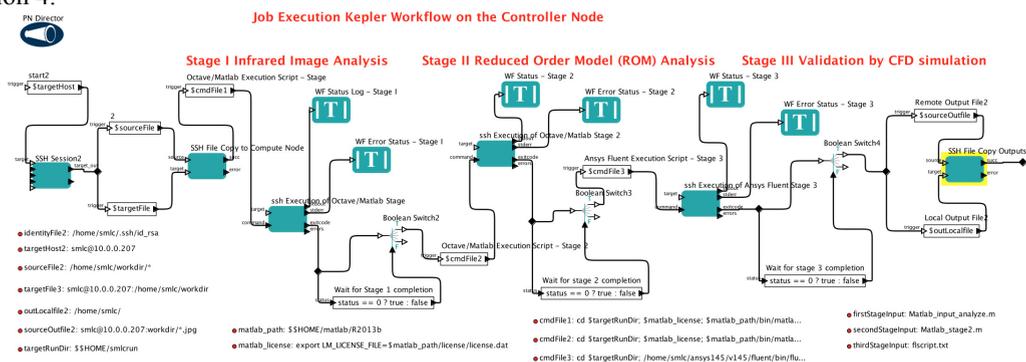


Figure 2: The Kepler workflow to do Image analysis, ROM validation and CFD simulation.

The Kepler workflow and all the computation that runs on the cloud resources use statically provisioned compute cloud instances with pre-deployed application software packages namely Matlab, Octave and ANSYS Fluent. The REST services will transfer files to the cloud computing resources and trigger workflow on the controller node where the workflow resides and wait for workflow to

complete so that it can transfer the results of computation to the cloud storage resource. The workflow when called will initiate the transfer of the input files from controller node to appropriate compute instances and successively run the jobs in various stages and copy the outputs back to the controller node. The computation itself goes through three successive stages involving either Matlab or Octave (first two stages) and ANSYS Fluent (third stage).

```

"endpoints": {
  "status": "",
  "results": "https://198.51.100.10:8080/auth/v1.0/AUTH_a332e6d7/workflows/instances/1/outputs"
},
"parameters": {
  "StageThreeScriptName", "flscript.txt"
},
"files": {
  "CFDModel": "octave-fluent/instances/1/inputs/CFDmodel.cas",
  "InputImage": "octave-fluent/instances/1/inputs/Input_T_image.jpg",
}
}

```

Figure 3. Example input JSON file for workflow service

```

{
  "files": {
    "Flame.jpg": "octave-fluent/instances/1/outputs/Flame.jpg",
    "T_proposed.jpg": "octave-fluent/instances/1/outputs/T_proposed.jpg"
  },
  "success": "true"
}

```

Figure 4. Example output JSON file for workflow service

The main actors used in these workflows are for SSH execution and secure file copy. Since the virtual instances are up throughout the execution, cloud-related actors are not used here. We plan to add cloud-related actors in the future to dynamically start virtual instances when execution requests arrive. The starting input is an infrared image file along with appropriate user scripts to run Matlab/Octave as well as ANSYS Fluent. Users are presented with a web GUI where they can upload the files, which are in turn copied over to OpenStack Object Storage system. As shown in Figure 3, the input manifest file is a JSON file that contains three dictionaries, namely *endpoints*, *parameters*, and *files*. The endpoints dictionary stores REST endpoints that the workflow service should use to communicate back to the web portal. In the example above, the workflow service would submit status updates to the status URL. The parameters dictionary element stores information about string parameters and their values. The files dictionary element stores information about input files. The keys are the file names, and the values are URLs in OpenStack Object Storage that contains the contents of the file.

## 4 Workflow Execution as a Service

The workflow service executes all stages of the workflow and manages virtual instances in a cloud computing and storage environment such as OpenStack. Each workflow is provided with a JSON input and output manifest template file. The input manifest will list the parameters, input files of the workflow and their default values. Portal can parse the input manifest for the input web page of the workflow. The output manifest will list which output files will be shown in the output web page of the workflow. The assumptions made in this services is that users of this workflow service needs to upload their own input files and run time scripts for either Matlab or Octave for the first two stages and for ANSYS Fluent in the third stage. The run time scripts can be either chosen from a repository at the web site if available or uploaded by the user. The user-uploaded scripts have to confirm to a rigid format so that the workflow does not end up with missing file names or wrong input names. The web portal provides an HTML form where users can upload the necessary files.

The workflows usually executes across multiple virtual instances. The workflow tool stages intermediate results into and out of OpenStack Object Storage or directly copy the inputs for the subsequent stages to the corresponding virtual instances. The initial input manifest contains an object

storage endpoint (e.g., results) that represents a writeable OpenStack Object Storage container that the workflow tool can use for downloading and uploading files. Additionally, as the workflow executes, it provides periodic update status. The initial input manifests file contains an endpoint where the status update messages are populated. Once the workflow has completed, it ensures that all output files have been written to OpenStack Object Storage, along with a manifest file. The workflow system then sends a final status notification that the workflow has completed its execution, with a link to the manifest. The output manifest file is a JSON-formatted file that contains URLs to all of the output files along with any other relevant metadata for the workflow.

## 5 Conclusion

We have deployed a Kepler workflow in a test bed built for SMLC to incorporate manufacturing intelligence in a cloud compute environment in real-time with data movement through public and private networks. This test bed was built entirely of open source packages such as Kepler, OpenStack and Django. Only the application software such as Matlab and ANSYS are proprietary. Our initial experience shows that WfaaS is a good way to support flexible workflow applications on cloud and how to support it using Kepler. We plan to build more such services in our test bed. We will also apply our new WfaaS architecture and workflow scheduling algorithms in [7] into the test bed.

## 6 Acknowledgements

We acknowledge the DOE grant DE-EE0005763 “Industrial Scale Demonstration of Smart Manufacturing Achieving Transformational Energy Productivity Gains”, UCLA cyberinfrastructure grant to the IDRE’s virtual computing laboratory and free software licenses from MathWorks Inc, and ANSYS Inc. to conduct this study. Julie Tran, SMLC project manager was very helpful in getting us the resources needed for this study and organizing meetings.

## References

1. Smart Manufacturing Leadership Coalition: <https://smartmanufacturingcoalition.org>
2. S. Chand and J.F. Davis, “What is smart manufacturing”, Time Magazine Wrapper, 2010, July.
3. J. Wang, P. Korambath, S. Kim, S. Johnson, K. Jin, D. Crawl, I. Altintas, S. Smallen, B. Labate, K. Houk, “Facilitating e-Science Discovery Using Scientific Workflows on the Grid” in X. Yang, L. Wang, W. Jie (eds), Guide to e-Science: Next Generation Scientific Research and Discovery. Springer, ISBN: 978-0-85729-439-5, 2011, pp. 353-382
4. J. Wang, P. Korambath, S. Kim, S. Johnson, K. Jin, D. Crawl, I. Altintas, S. Smallen, B. Labate, K. Houk, “Theoretical Enzyme Design Using the Kepler Scientific Workflows on the Grid”. In proceedings of the Fifth Workshop on Computational Chemistry and Its Applications (5th CCA) at International Conference on Computational Science (ICCS 2010). pp.1169-1178.
5. OpenStack Cloud Software: <http://www.openstack.org/>
6. Kepler Project: <https://kepler-project.org>
7. J. Wang, P. Korambath, I. Altintas, J. Davis, D. Crawl. “Workflow as a Service in the Cloud: Architecture and Scheduling Algorithms”. Accepted by International Conference on Computational Science (ICCS 2014)