

[CC BY-ND 4.0 DEED Attribution-NoDerivs 4.0 International](#)

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.



A Smart Manufacturing Use Case: Furnace Temperature Balancing in Steam Methane Reforming Process via Kepler Workflows

Prakashan Korambath^{1*}, Jianwu Wang², Ankur Kumar³, Jim Davis¹,
Robert Graybill⁴, Brian Schott⁴, and Michael Baldea³

¹Institute for Digital Research and Education, University of California, Los Angeles, CA, USA,

²Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD, USA, ³Department of Chemical Engineering, University of Texas, Austin, TX, USA,

⁴Nimbis Services, Inc. McLean, VA, USA

ppk@idre.ucla.edu, jianwu@umbc.edu, ankur_kr@utexas.edu, jdavis@oit.ucla.edu,
robert.graybill@nimbisservices.com, brian.schott@nimbisservices.com, mbaldea@che.utexas.edu

Abstract

The industrial scale production of hydrogen gas through steam methane reforming (SMR) process requires an optimum furnace temperature distribution to not only maximize the hydrogen yield but also increase the longevity of the furnace infrastructure which usually operates around 1300 degree Kelvin (K). Kepler workflows are used in temperature homogenization, termed as balancing of this furnace through Reduced Order Model (ROM) based Matlab calculations using the dynamic temperature inputs from an array of infrared sensors. The outputs of the computation are used to regulate the flow rate of fuel gases which in turn optimizes the temperature distribution across the furnace. The input and output values are stored in a data Historian which is a database for real-time data and events. Computations are carried out on an OpenStack based cloud environment running Windows and Linux virtual machines. Additionally, *ab initio* computational fluid dynamics (CFD) calculation using Ansys Fluent software is performed to update the ROM periodically. ROM calculations complete in few minutes whereas CFD calculations usually take a few hours to complete. The Workflow uses an appropriate combination of the ROM and CFD models. The ROM only workflow currently runs every 30 minutes to process the real-time data from the furnace, while the ROM CFD workflow runs on demand. ROM only workflow can also be triggered by an operator of the furnace on demand.

Keywords: workflow, smart manufacturing, real-time data, Matlab, Ansys, Tableau, ROM, CFD.

* Corresponding Author

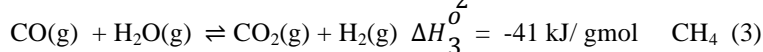
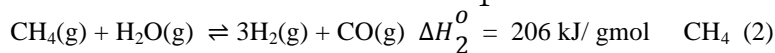
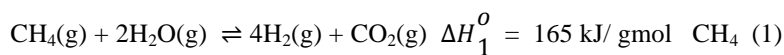
1 Introduction

In this paper we describe how Kepler Scientific workflows are used in the context of smart manufacturing [1] in optimizing temperatures across a commercial industrial scale furnace in steam methane reforming process used to manufacture hydrogen gas. Steam Methane Reforming (SMR) process involves a large furnace with over 300 catalyst-filled tubes with an operating tube temperature of around 1300 K and has several fuel-valves to control the flow of natural-gas fuel gases through a fleet of burners which in turn control the temperature distribution in the furnace. A uniform temperature distribution across the furnace leads to higher overall energy efficiency of the plant [2]. However, maldistribution of flue gases in the furnace along with other disturbances in the furnace operating conditions cause spatial temperature non-uniformity leading to the plant running sub-optimally. We developed several workflows based on the open-source Kepler scientific workflow system [3] to assist the plant operators in homogenizing the temperature distribution by providing appropriate control move recommendations. Infrared (IR) temperature sensors are installed on selected locations across the furnace to measure tube temperatures accurately in a non contact way. The output of the IR readings per minute is stored in a Historian database at the plant site. The readings are then transmitted to an OSI/PI Historian database [4] deployed for this work at an OpenStack based cloud environment operated by Rackspace Inc. The workflows are configured to run at a regular interval of 30 minutes on our cloud resource which consists of virtual machines running Linux and Windows operating systems and push the computation outputs back to the Historian. The interval of the workflow running period may be decreased to a slower pace later on as furnace condition does not alter much within 30 minutes. The execution of the workflows can also be initiated on-demand by the plant operator through a Tableau-based user interface as described later [5]. Workflows access Historian data as input through a CSV file populated based on certain preconfigured data-tags. The outputs of Reduced Order Model (ROM) calculation recommend operators at the plant how much the valve openings need to be changed to balance the temperature of the tubes to avoid the unfavorable hot or cold spots. The calculation outputs are automatically stored into the PI Historian. The plant operators see these results through the Tableau interface which is connected to the PI Historian through an ODBC connection.

The rest of the paper is organized as follows. Section 2 discusses chemical reactions involved in steam methane reforming process. Section 3 describes computational resources and architecture on which the calculations were performed. Section 4 outlines ROM techniques that were employed in the computation. Section 5 gives a detailed description of the Kepler workflows that we configured to do the data transfer and computations. In section 6 we preview a related work which is called “modeling the modeler” that is being designed but not fully implemented yet. Finally, Section 7 concludes the paper and gives an outlook of the future research.

2 Steam Methane Reforming Process

SMR is a process where natural gas and steam are converted into hydrogen, carbon dioxide and carbon monoxide. The predominant reactions are [6]



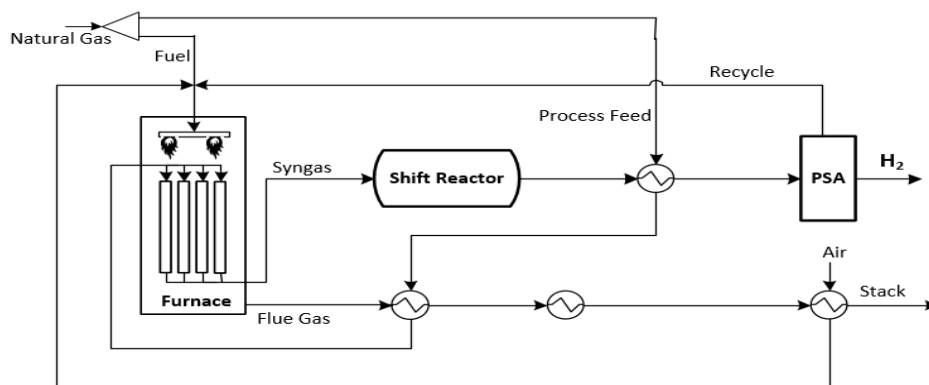


Figure 1: Process diagram of a typical hydrogen manufacturing plant.

The first two reactions being endothermic need energy to be supplied in order for the reactions to take place. These reactions take place in hundreds of catalyst-filled tubes placed in a top-fired furnace as shown in Figure 1. The tubes are heated through radiation-heat resulting from air-assisted combustion of natural-gas based fuel flowing through close to a hundred burners at the top of the furnace. Methane and steam flow continuously through the tube. The temperature distribution of the tubes (equivalently, the furnace) strongly depends on several factors such as plant production rate, air distribution among the burners, design of the burners, etc. In practice, the tube temperatures can differ by more than 100 K. Such a high temperature non-uniformity prevents the furnace from being operated at a higher average temperature, i.e., the average temperatures of the tubes is lower than the optimal value, due to which the plant efficiency decreases. The reader is referred to the work by Kumar et. al [2] for a detailed discussion on the impact of temperature non-uniformity. However, the fuel distribution through the burners can be manipulated to minimize this non-uniformity. Since the temperature distribution get affected by disturbances such as changes in ambient temperatures, adjustments in the optimal fuel distribution needs to be estimated periodically. In this regard, a computationally parsimonious data-driven reduced order furnace model (ROM) [7] was developed to run in real time and predict tube temperature distribution as a function of fuel distribution with reasonable accuracy. Provision for validation of ROM based optimization calculations through a high-fidelity CFD model of the furnace is provided as well in the workflow.

The inputs for the calculations are sensor data (images) from infrared cameras as well as total fuel flow-rates, total air flow-rates etc. The inputs are preprocessed using Matlab software followed by ROM analysis using the same software. The computational fluid dynamics (CFD) calculations are performed using Ansys Fluent software. ROM calculations are fast (in the order of minutes) and they can be computed using Matlab whereas CFD calculations, being of very high quality, takes a lot of time, in the order of 10 hours on 32 CPU processors. Such high run-time of the CFD model precludes its usage directly during the optimization calculations and thus it is used only for validation as alluded to earlier. However, the results from CFD are further used to refine ROM to better predict the temperatures from flow rates. Updated ROM results are used to adjust the flow rate of the fuel, hence the temperatures of the tubes. In a factory environment ROM calculation which takes only few minutes are always preferred over accurate CFD calculations to make adjustments. The reason being that even though ROM calculations may give 60% accuracy the operators can run multiple iterations to reach the optimum value to improve the accuracy of the prediction. Also, the CFD calculations do not generally scale very well beyond 64 processors in many cases due to increase in inter-processor communication overhead. So we cannot arbitrarily increase the number of processors to run CFD calculations faster. So running ROM model is very appropriate in this scenario.

All computations on the test bed are automated through workflows for efficient reliable operation of furnace, which is the most expensive, and energy intensive equipment in the SMR process. 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. The workflows leverage data from plants in real time to predict, manage and optimize performance and control the temperature for higher efficiency and prolonged equipment life.

3 Computational Resources and Architecture

The computation of the ROM and CFD model is carried out on an OpenStack based cloud computing and cloud storage environment at Rackspace. The resources are deployed and maintained by Nimbis Inc [8]. This operation requires two Windows OS based virtual instances to support OSI/PI Historian and a Tableau visualization software. For running ROM and CFD calculations additional Linux based VMs may be required. Currently, we deployed one virtual instance each for managing Kepler workflows, running Matlab calculations and running CFD calculations. Additional instances for CFD calculations may be deployed when they are run in parallel using MPI message passing scheme in a distributed computing manner. Each virtual instances have at least 6 GB of memory and four virtual CPUs. Additionally, Nimbis runs a license manager instance for Matlab and Ansys. A schematic of the resources deployed for running the furnace-optimization jobs at Rackspace infrastructure is given in Figure 2. It was decided to deploy functionality for each operation in separate VMs as same image can be reused in other workflows without much change.

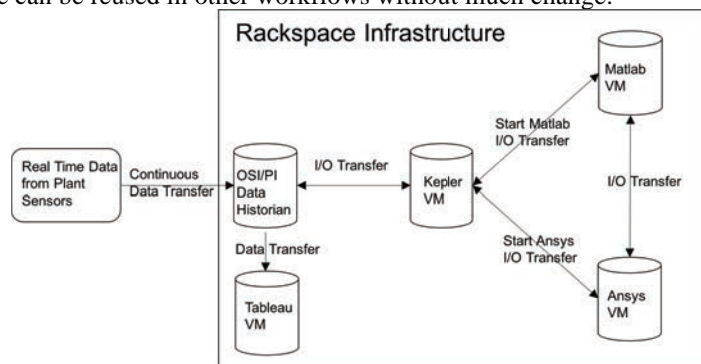


Figure 2: Architecture for running SMR jobs at Rackspace infrastructure.

Additionally, this project also makes use of an OpenStack based cloud resources deployed at IDRE, UCLA to develop and test Kepler workflows. The current test bed hardware at UCLA consists of 16 IBM iDataPlex 2U units with Intel Nehalem E5530 CPU having 8 cores per node and 24 GB RAM per node. There is one controller node with 12 cores and 32 GB RAM and three Gigabit network switches. The workflow we developed is portable in such a way that the same workflow can be run either on Nimbis resources or UCLA resource by just changing the IP addresses of the Matlab images. Since ROM computations take only few minutes there are no considerable time differences for computation between the two resources. We have already published a REST based workflow as a service (WfaaS) for task orchestration to facilitate workflow services on this environment [9].

4 Computational Models

4.1 Reduced Order Model (ROM) Creation Methodology

Availability of furnace-wide measurements through advanced IR-sensors allows the possibility of using data-driven methodologies for creation of process models. A modeler can leverage a number of well-established modeling techniques such as ordinary least square (OLS) regression or partial least square (PLS) regression [10] or can create their own custom model depending on the specific application. Figure 3 shows one such application where historical plant data upon post-processing is used for OLS-based model generation. The associated process consists of appropriate Matlab scripts for the OLS module and post-processing (moving average filter). The output is essentially a Matlab data-file with the generated model parameters to be used in a later process to predict the manipulation of valves to balance the temperature.

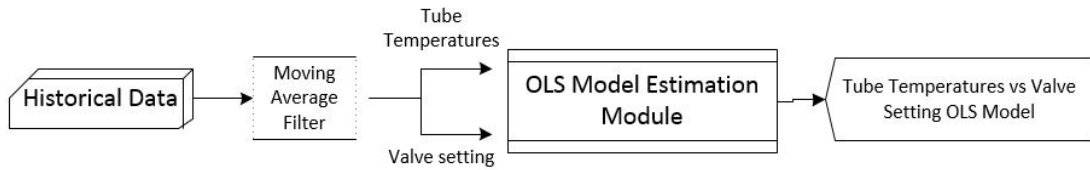


Figure 3: OLS Model generation scheme.

4.2 Model Deployment in Optimizer Module

Figure 4 shows the process when a model created by the modeler in Section 4.1 is employed in an optimizer process. The optimizer may be a quadratic program or a linear program that uses the OLS or PLS model generated in the previous step. It uses the current valve settings and tube temperatures as inputs and outputs the new valve settings for the optimum temperature distribution.

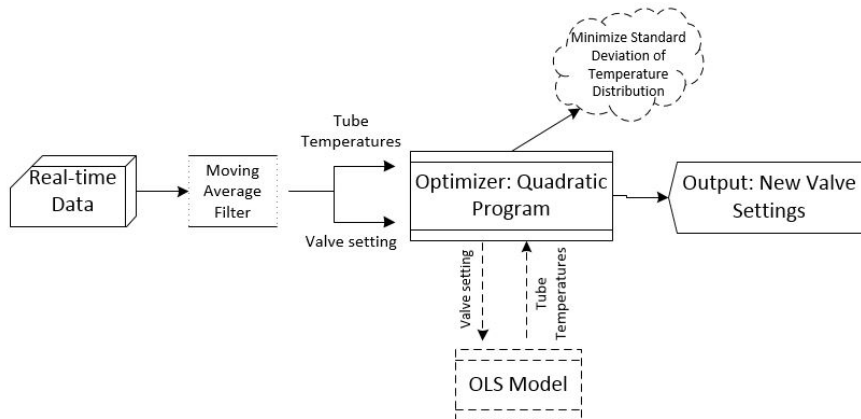


Figure 4: Model deployment in optimizer module.

4.3 Model and Optimizer Integration

Figure 5 shows the integration of the above model and optimizer modules that can process real-time sensor data from the plant and generate plant equipment adjustment suggestions. The integrated process requires a CFD model provided for the targeted plant to check the accuracy of the ROM-based optimizer calculations. If the accuracy is good, the process will finish and generate plant equipment

adjustment suggestions. Otherwise, it will update ROM model using the CFD data and iterate the whole process execution by using the latest data from sensors.

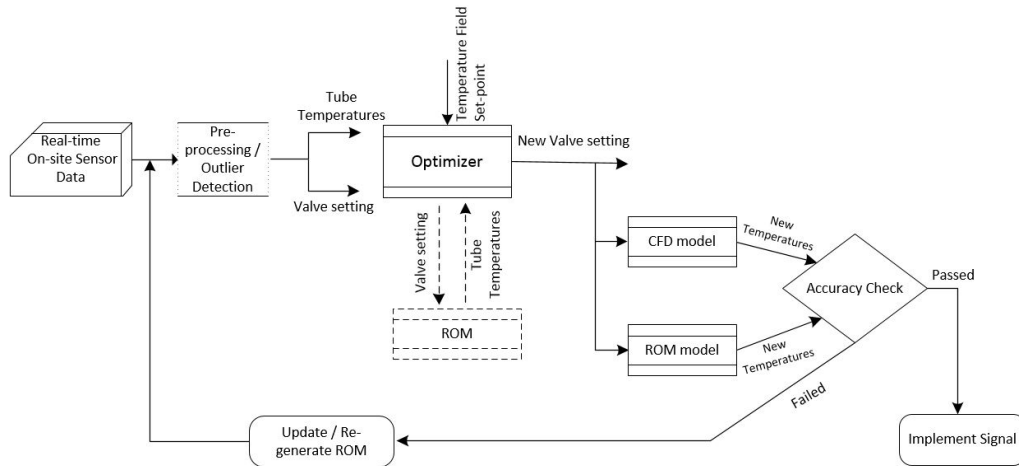


Figure 5: Model and optimizer integration.

5 Kepler Workflows

To support the computation models and processes in Section 4, we built several Kepler workflows to automate computation using Matlab [11] and Ansys Fluent [12]. Two of those workflows are described in this paper. The majority of the actors in this workflow are SSH-Execution actors, which performs either Matlab or Ansys Fluent computations on the virtual machines (VM). The workflows are usually triggered from one of the VMs where Kepler is installed. All VMs are configured for passwordless SSH and SCP access for the user account on which the workflow is running. The workflow execution will initiate the transfer of the input files from the Historian VM to appropriate compute instances and successively run the jobs in various stages and copy the outputs back to the Kepler VM as well as to the Historian. The computation itself goes through three successive stages involving Matlab, and one stage involving ANSYS Fluent. A similar workflow using WfaaS is explained in one of our earlier publications [9] but no dynamic data was used in that workflow. In the current workflow, all data are dynamic and the calculated results are instantaneous.

The input data for the workflow is retrieved from the local Historian. The user of the workflow never interacts with the factory directly for the inputs because the data Historian is continuously receiving the data from the plant and the workflow is preconfigured to copy the latest data from the Historian. We plan to create the input data based on user defined tags such as average temperatures every hour or maximum temperatures of various tubes etc.

5.1 ROM Only Workflow

ROM only workflow shown in Figure 6 is the Kepler workflow based on the model deployment in the optimizer module shown in Figure 4. The first stage is the transfer of the inputs from a directory in Kepler VM to the Matlab VM. This inputs consist of the Matlab scripts that will be used in the calculation. The workflow then retrieves the latest input data from the OSI/PI Historian which could be an Excel spreadsheet or a CSV file. Then it does two stages of Matlab calculation with output from first stage being used as an input for the second stage and so on. The computation is followed by a transfer of the output file to the OSI/PI Historian. The workflows are completely parameterized so that

they can be deployed on any cloud resources. Typical variables are the IP address or hostname of various VMs, home directory of the user and name of the compute directory etc. These variables can be changed at the time of submission. The Kepler workflow itself is stored as an XML file and the arguments are given on the command line. If no arguments are given to the workflow, default argument values will be used. This workflow has been tested at least on two test-beds.

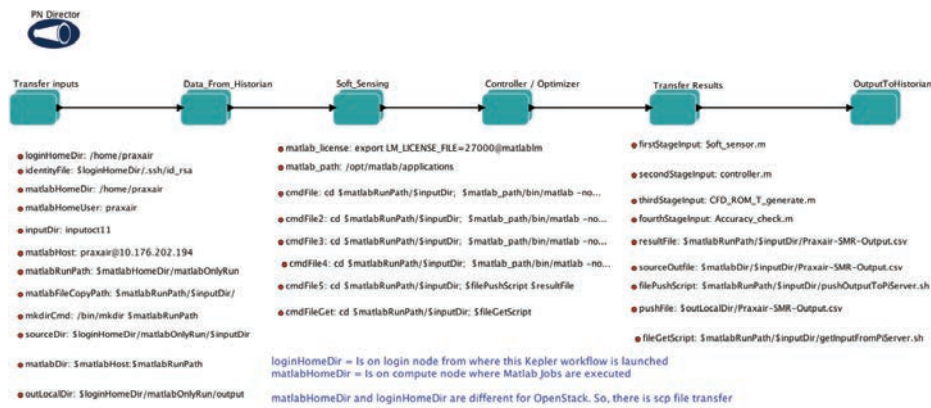


Figure 6: ROM only workflow.

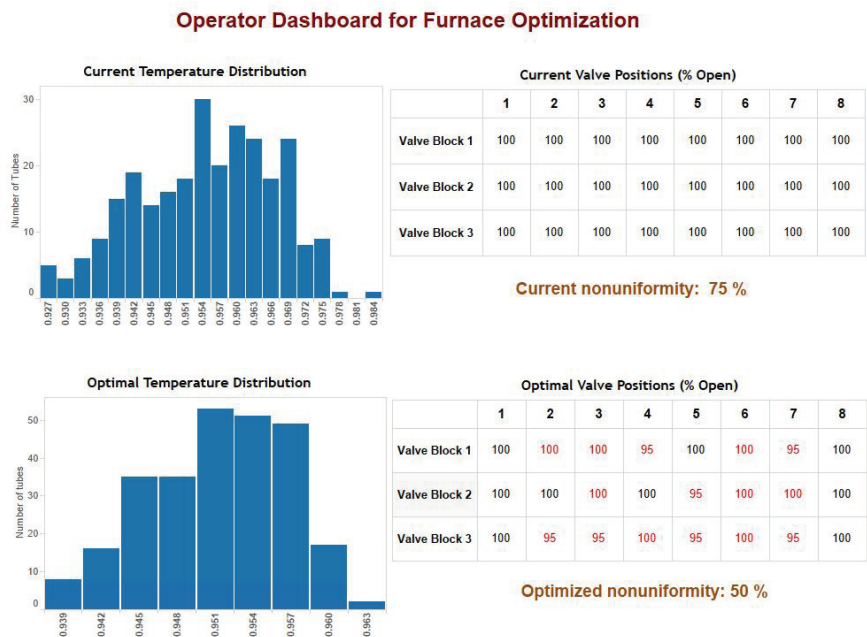


Figure 7: An operator view of the Tableau dashboard (Note that the data has been normalized or changed to conceal proprietary plant data).

We use Tableau visualization software to view the results of the calculation. An operator view of the Tableau dashboard is given in Figure 7 where operators can look at the color coded table and adjust the flow rates to balance the temperature. The x-axis in the histograms is the normalized

temperature i.e. the temperature divided by the reference temperature. We are displaying the normalized temperature instead of actual temperature because of the proprietary nature of the data, but the operators at the plant site will be able to see the actual temperature value. The same reference temperature is used in both the histograms. The y-axis is the number of tubes.

5.2 CFD Workflow to Update ROM

Figure 8 shows a complete workflow where the ROM can be updated based on the results from CFD calculation. The plant operators always need an output from ROM calculation as they can be completed in minutes. The CFD calculation however, is time consuming and also requires number of processors to complete. Like the ROM only workflow in Section 5.1, this workflow also starts with transferring the input scripts to the computational VMs. In this case the workflow needs to transfer Matlab as well as Ansys Fluent scripts and data files over to the respective images as well as the latest input from the OSI/PI Historian. At the end of the ROM and CFD calculations, this workflow does validation between the two calculations and updates the ROM on the fly to improve the ROM results.

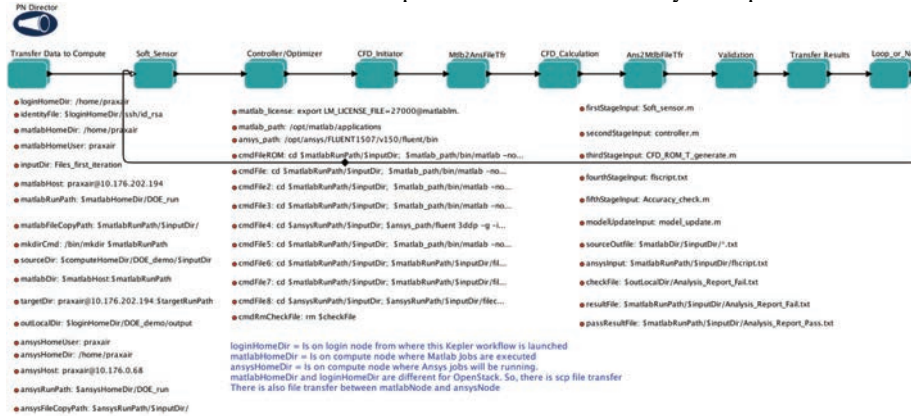


Figure 8: Model and Optimizer workflow with CFD to update ROM.

Unlike the ROM-only workflow, this one has not been tested with real-time data on our test bed because our project collaborators are still improving the CFD model to predict the temperature distribution. Currently we are only testing a preliminary CFD model and the ROM so that we can go ahead with the workflow development process. This workflow also has an exit condition on how many iterations it will run in the event ROM results are not converged to the CFD level. In such event we need to restart the workflow as the conditions inside the furnace might have changed. Appropriate messages are communicated to the operators on the Tableau dashboard in such an event.

One of the differences between the two workflows described above is the computational time. The ROM only workflow takes only around 5 minutes where as the CFD workflow could take anywhere from 4 to 6 hours to complete. So, it is not always practical to run the CFD workflow, but we need to run the CFD workflow at least once a day to verify the ROM parameters. In this project the factory condition does not normally alter much in a day. So, it is perfectly reasonable to run the CFD workflow once a day and run the ROM only workflow as and when the operators need it.

6 Modeling the Modeler Concept Design

Modeling the modeler is a concept that is in the design stage where we are planning to use the workflows to facilitate the model creation process for users who are not necessarily a domain expert.

Typically, modeler has a set of data received from a factory that he or she is trying to analyze and figure out correlation among various elements in the data set so that they can predict some future outcomes or optimize the manufacturing process. The best model he or she arrived should be able to create the data they started backwards. The platform will provide ready-made toolkits in the form of user interfaces, software tools, workflows and web services that can be combined to build the model. The users of this service will be able to analyze the performance of the model using the output reports generated by the web services and make changes in the model or the workflow on the web based interface in case of unsatisfactory performance. As shown in Figure 9, this process involves various components such as data processing, model creation, model optimizer, modeling software and model deployment. The sample data is retrieved from a data Historian. The modeler never interacts with the factory directly. The data Historian is continuously receiving the data from the plant and the modeler is sending request to the Historian to send them the input data based on user defined tags such as average temperature every one hour or maximum temperatures of various tubes etc. Then as described in section 4.1, users can leverage established modeling techniques to build the model.

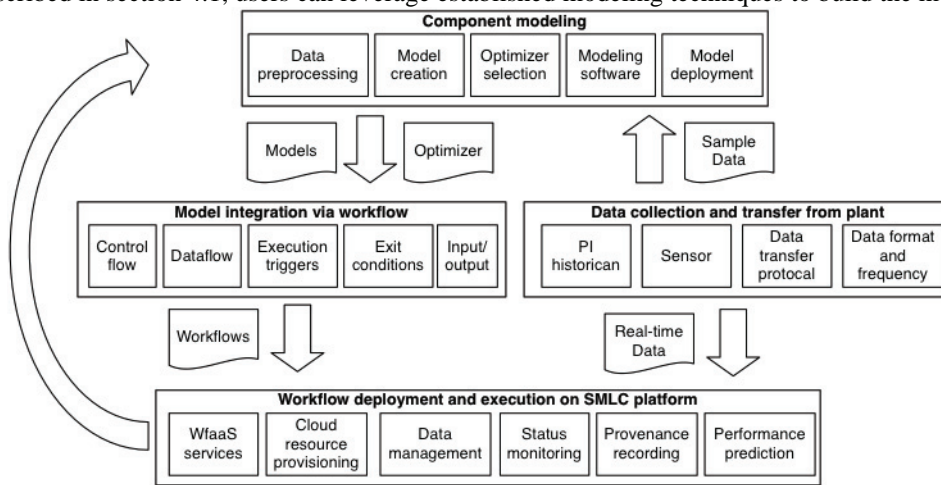


Figure 9: Process overview of the general procedure for modeling the modeler.

7 Conclusions and Ongoing Work

The ROM only workflow has been tested for robustness by continually running for around 60 days. Only time we interrupted the continuous 30 minutes' operation was when we needed to make some improvements to a web interface Nimbis added to the platform. We have additional cron jobs that run on the system to make sure input file time stamps concur with system time. That way we are sure that we are processing the latest input files. We will continue to monitor the stability of the whole process for several weeks to make sure the system can handle workflow that triggers every 30 minutes. For this plant operation we could decrease the frequency to hourly or even less. The workflow itself takes under 5 minutes to completion. So, every workflow is completed before the new workflow starts. The Data Historian which is receiving data every minute from the plant has been in operation for the past 6 months. As part of the future work we will be conducting experiments to expand the computation from public cloud such as Rackspace over to a private cloud such as the OpenStack cloud at UCLA. Currently, the result of the workflow is running view only mode meaning the workflow is not directly regulating the flow of the gases, but the operators can look at the suggestions and decide whether to act up on it or not. We are in the design stage of a sensor actuator so that the workflow can automatically regulate the valve operations directly from the output of the

calculations without operator interaction. The input pre-processing part of the Matlab scripts in the workflow also takes into account of the fact that sometimes the sensor camera can go offline or loose focus on the target tubes. Since all the compute components of the workflow is contained in a cloud computing resources, we can reproduce identical set ups from the image snap shots on another IaaS system within an hour in the event of any hardware failure or any such event.

We used Kepler workflow in this project mainly because it is designed to work in a high performance computing environment. It has actors and capability for decision making, looping and executing ssh commands on a remote Linux VM etc. We also use PN Director for our workflows which allows parallel executions to proceed when branching happens in the workflow. The combined ROM CFD workflow is an iterative process where we use the iteration feature of the workflows to improve the quality of our ROM model.

8 Acknowledgements

We acknowledge the DOE grant DE-EE0005763 “Industrial Scale Demonstration of Smart Manufacturing Achieving Transformational Energy Productivity Gains”, and SMLC-NIST Partnership Study, Award # 70NANB12H219. We also acknowledge the free software licenses from Mathworks Inc., Ansys Inc., and Tableau Inc. for this work.

References

1. J. Davis, T. Edgar, R. Graybill, P. Korambath, B. Schott, D. Swink, J. Wang, J. Wetzel (2015), “Smart Manufacturing”, Annual Review of Chemical and Biomolecular Engineering, 6, 141
2. A. Kumar, M. Baldea, T. Edgar, O. A. Ezekoye (2015), “Smart manufacturing approach for efficient operation of industrial steam-methane reformers”, Industrial & Engineering Chemistry Research, 54 (16), 4360
3. Kepler, [online]. Available: <https://kepler-project.org>, accessed Jan 17, 2016
4. OSIsoft Inc. [online], Available: <http://www.osisoft.com>, accessed Jan 17, 2016
5. Tableau, [online], Available: <http://www.tableau.com> accessed Jan 17, 2016
6. D. A. Latham, K. B. McAuley, B. A. Peppley, T. M. Raybold (2011). “Mathematical modeling of an industrial steam-methane reformer for on-line deployment” Fuel Processing Technology 92 1574-1586
7. A. Kumar, M. Baldea, T. Edgar (2016). “Real-time Optimization of an Industrial Steam-Methane Reformer Under Distributed Sensing”, Submitted to Journal of Control Engineering Practice.
8. Nimbis, [online], Available: <http://www.nimbisservices.com>, accessed Jan 17, 2016
9. P. Korambath, J. Wang, A. Kumar, L. Hochstein, B. Schott, R. Graybill, M. Baldea, and J. Davis (2014), “Deploying Kepler Workflows as Services on a Cloud Infrastructure for Smart Manufacturing”, Procedia Computer Science 29, 2254
10. S. Wolde, M. Sjöström, and L. Eriksson. (2001) “Some recent developments in PLS modeling”, Chemometrics and intelligent laboratory systems 58, no. 2 109-130
11. Mathworks, [online], Available: <http://www.mathworks.com>, accessed Jan 17, 2016
12. Ansys, [online], Available: <http://ansys.com>, accessed Jan 17, 2016