An Integration Framework for Process Automation of Wind Plant Siting

Mark M. Kornfein, Vrinda Rajiv, and Jenny W. Williams

Abstract— **Many engineering processes use custom or third party applications to perform specific types of analyses and calculations. Typically, these applications are run by experts, often in localized environments on a user's desktop, remote computer or in a high-performance computing environment. These applications often rely on extensive manual intervention and expert knowledge to prepare inputs and interpret outputs.**

In this paper, we describe a novel system for wind plant siting to address these challenges, enabling a unified way to execute a disparate set of specialized analyses and programs from a common web interface. The system implements a platform agnostic architecture that runs these heterogeneous applications on distributed compute servers. It handles numerous types of input data, interfaces between applications, analyzes the output of each application, and presents a summary of the results.

Use of this integration framework reduces turnaround time, increases process consistency, reduces likelihood of errors, and captures expert knowledge to establish corporate memory. The modular architecture allows the addition of analysis modules with minimal effort. The framework described here has been deployed at GE Wind to facilitate wind plant siting. It is used to run over 900 analyses monthly and improves typical customer turnaround time by 80 percent.

Index Terms— **Integration framework, knowledge engineering, knowledge management, process automation, wind energy**

I. Introduction

The siting of a wind plant involves multiple analysis steps to determine the feasibility of installing wind turbines in specific locations. In addition to considering government, environmental and legal policies, it is essential to determine the engineering feasibility of installing wind turbines of a selected make and model. For example, siting a wind plant may require a disparate set of applications to perform analyses such as computational fluid dynamics, wind resource analysis, loads analysis and energy output calculation. The engineering feasibility of a specific wind turbine is based on ambient wind conditions, turbulence and geographical data in the selected location, as well as the capabilities of the selected wind turbine model [1].

This engineering evaluation typically involves an extensive set of site-specific analyses, executed successively, with the final results typically passed on to a customer. In the previously existing process, geographically dispersed engineers used separate programs on their local computers to perform these calculations. Each individual analysis step involved manual touch points for providing initial and intermediate inputs. Engineers exchanged necessary inputs and outputs to each other electronically by email or file transfer, as needed.

This process was time consuming, often resulting in multi-week turnaround time to return even preliminary results to a potential customer. It required significant amounts of training for analysts to come up to speed on the process and necessitated detailed reviews to ensure accurate and consistent results. In addition, it was time consuming and difficult to track the results or audit past analyses. The existing process needed an infusion of technology to improve accuracy, consistency and speed.

II. Related Work

This work augments the idea in 'Wind turbine fulfillment center' [2] which describes a knowledge management and automation process for wind loads analysis. Process knowledge capture is required to instantiate a process automation integration framework. The knowledge capture performed for this system is described in [3]. This type of framework is very similar to multi-disciplinary design optimization (MDO) frameworks built for designing large-scale, complex products such as those found in aerospace or ship building. The challenge in siting of a wind plant is in many ways analogous to designing a new product for each site. MDO was initially described by [4]. A number of MDO concepts are incorporated into our work:

1. Design-oriented Analysis - the ability to try "what-ifs"
2. Approximation Concepts - the ability to make meta-models to substitute for a more complex complete analysis
3. Decomposition - the ability to partition an engineering problem
4. Optimization procedures - the ability to execute code in an efficient manner
5. Computer human interface - the ability of an engineer to interact with the system and see the final and intermediate results

In addition [5] includes a workflow component into the MDO framework concept. The workflow component is specifically added to support collaborative design with

Manuscript received June 25, 2012; revised July 30, 2012.

M. M. Kornfein is with GE Global Research, Niskayuna, NY 12309 USA (518-387-5843; e-mail: kornfein@ge.com).

V. Rajiv is with GE Global Research, Niskayuna, NY 12309 USA (e-mail: vrinda.rajiv @ge.com)

J. W. Williams is with GE Global Research, Niskayuna, NY 12309 USA (e-mail: jenny.williams @crd.ge.com).

geographically distributed teams such as those involved in wind plant siting.

Our framework is also an example of an Enterprise Application Integration (EAI) for engineering tying together legacy systems, databases and data [6] for the engineering domain.

These mentioned systems, however, are strongly related to the specific task of engineering design and are not directly suitable for process automation of wind plant siting. They tend to require extensive code development for each tool integrated. In addition, integration of the decomposed parts requires a high level of expertise among the user community.

III. Approach

To address the challenges discussed in section I above, multiple approaches were considered. The first approach considered involved enabling users to manually run individual analysis applications on a remote server using scheduling software such as AppWorx [7]. This approach was relatively easy to implement and ensured a consistent code base, but still required extensive manual steps and analyst training. A second approach considered, involved building a single large fixed application to handle all analyses. This approach removed manual intervention and simplified engineering tasks, but would have resulted in a system that would be difficult to maintain and not easily extensible [8].

The integration framework approach described here provides a collaborative knowledge management platform for engineers, sales representatives and customers to easily share data, run engineering analyses and access results. It provides a mechanism to rapidly incorporate new and updated analysis modules into the system, with minimal development effort. The analyses are performed under a unified global management and control process.

IV. System Architecture

The integration framework architecture is shown in Figure 1. In building the integration framework, several goals were paramount. First, the framework must provide a single access point to a heterogeneous set of applications. Second, it must support a diverse set of applications that run on multiple platforms (Windows, UNIX, Linux). Third, it must be able to quickly plug-in engineering or business tools that interact with tools already in the framework. Fourth, it must offer the ability to rapidly prototype new functionality. Fifth, it must offer the ability to add capabilities rapidly without the typical procedural code deployment bottlenecks found in many large organizations.

To meet these requirements, a number of commercial tools were considered to provide the basic infrastructure for the framework. After a thorough search of this space, EASA software [9] was chosen as a tool that could best meet these needs. EASA provides capabilities to queue processing jobs, monitor progress and view results on remote or local servers. It also has an authoring environment to create applications, and a Java API that can be used to extend authoring capabilities.

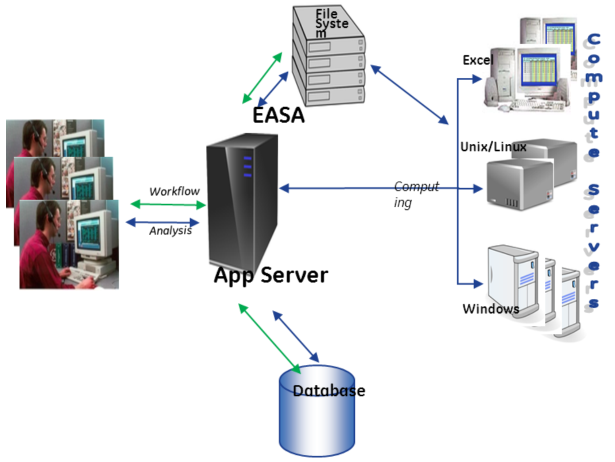

Fig 1. Integration Framework

The EASA application server provides an interface for users to run jobs, interact with a database and process the inputs and outputs of a job.

A unified common file system is utilized to support all data and file storage. This includes files related to each type of analysis in the system, such as executables, scripts, and batch files, as well as project data such as customer data, input and results files, and reports generated by the system.

Dedicated compute servers are utilized for the sole purpose of running analysis computations. Due to the system architecture, these compute servers require only a minimal footprint, with only a few basic utilities installed on the servers. Thus, deployment of a new compute server is fast and its maintenance is minimal. In addition, virtual machines or cloud computing capabilities may be seamlessly leveraged.

A relational database stores the metadata to the common file store. The database contains information about the available executables, for example which executable to use for an analysis type, its location on the file store, and what supporting data is needed. The database also holds metadata indicating the locations of input data and results for an individual project. Further, the database contains general information about each project, such an environmental, meteorological, and turbine layout data. Finally, the database also stores result data, which is used in reporting and also facilitates use of the system as a knowledge repository by GE Wind. In most cases, the database schema is generalized, such that adding a new analysis to the system requires only the addition of data rather than a schema change.

The system architecture also features a user workflow, enabling users to assign projects, upload customer data and other files, send email notifications and participate in discussion forums. In addition, it allows engineers to input turbine model information including specifications, analyses

ISBN: 978-988-19252-4-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

to run, turbine component definitions and turbine controller simulation software. These can all be added to the system by the engineering team without support from information technology team.

For some types of analysis, framework extensions are needed, requiring analysis-specific coding. Two types of extensions are used in the integration framework. First, a Java API is used to extend EASA software. The API allows for classes to be used to extend the user interface or process results from an analysis. A second extension to the framework is software that runs on the compute servers. These include scripts, Visual Basic code, or batch files that are needed to start an analysis or parse analysis data.

V. Current Implementation

The wind plant siting domain provides a novel implementation of the integration framework. It combines workflow elements, engineering analyses, customer reporting and knowledge retention.

Figure 2 describes the typical activity flow for performing wind plant siting analysis. A project leader assigns an engineering team to the project using the system workflow. Several engineers are generally assigned to a given project, each with a different role, also captured in the system.

Inputs are entered into the system by the project team. Input types include customer data, data from engineering work performed outside the system, and turbine model information. The system parses and validates the inputs, performs necessary transformations and generates information for the various analyses.

Engineers can then initiate one or all of the available analyses – for example wind resource, power curve, relative mechanical loads, and turbine specific mechanical loads. Based on dependency rules, the system allows for parallel or sequential execution of these analyses. Relative and turbine specific mechanical loads analyses are usually required on most projects. Wind resource and power curve analyses are optional and are performed based on customer requests. In both these cases, multiple sub-types of analyses are supported in the system. In Figure 2, analysis tools that are in dotted boxes are run within the framework and can be rapidly replaced using the integration framework. The prototyping and implementation of a new analysis types is done with minimal effort. Inputs, outputs and dependencies are mapped and the analysis is paired with the appropriate compute environment. Customer reports are created to document the various analyses performed. These are automatically generated by the system in a format suitable for distribution.

At every step in this activity flow, information on inputs, processes and results is stored in the database. Enough audit information is also stored to enable the recreation and tracking of a past analysis instance.

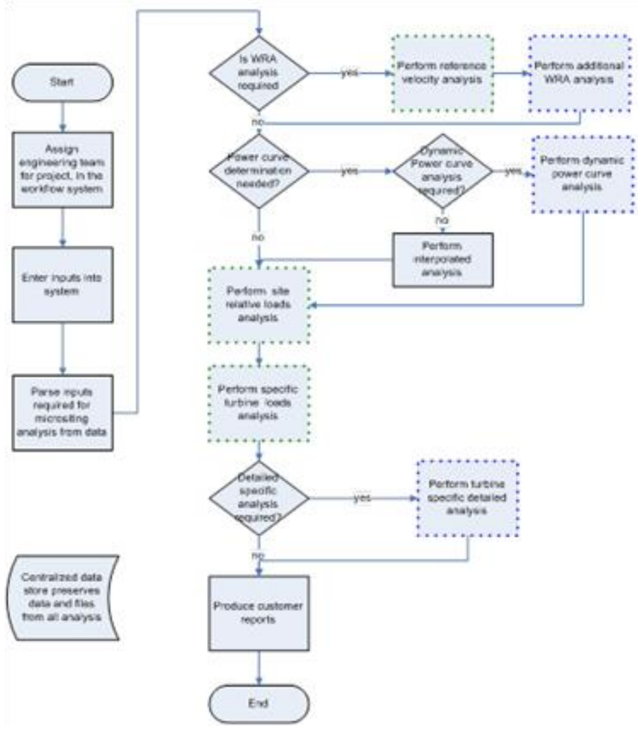

Fig 2. Typical wind plant siting workflow

VI. Benefits

The integration framework described here provides a number of advantages. It automates multiple sequential or parallel wind plant siting analyses, where each analysis may itself execute multiple steps. It provides the flexibility to define the set of steps that form the analysis, as well as the dependencies between them. It tracks all the steps in an analysis, serving as a knowledge repository while providing automated reporting and searching of previous analysis. It provides global internal access and, where desired, external access for customers.

The modular nature of the framework allows analysis tools to be swapped out rapidly when new versions are available. The system also facilitates addition of new types of analysis, often with no or little programming. The ability to deploy changes with little programming means that the information technology organization often does not need to be involved in functionality rollouts, reducing the cost and increasing the speed of deployment. The system it supports at GE Wind saw a ten-fold increase in productivity of wind plant siting, while improving quality and serving as a repository of corporate knowledge.

VII. Limitations

The framework has two limitations worth noting. First, the underlying analysis tools must conform to the system architecture; for example, they must be able to be run with limited interaction once initiated. The analysis output must be in a machine readable format so results can be parsed, stored and used to automatically generate inputs to subsequent steps. Second, when 3rd party software is a part of an analysis that runs on remote servers, licensing issues arise occasionally. In at least one case, extensive negotiation with a vendor was required to modify the license terms to allow the application to be used in this environment.

VIII. Future Work

Several extensions to the framework are possible. The first, currently under design, is providing a web service interface to access the various analyses within the system. This would allow other systems, and potentially customers, to execute analysis within the framework. Another area of interest is expanding the analytical analysis capabilities to the system. Finally, as the system continues to grow and store more wind plant siting data, there are increased opportunities to build semantic models from documents and design practices, and analyze the data for possible use in future sales, remote monitoring and diagnosis.

IX. Conclusion

The integration framework described here solves a number of challenges common to many engineering organizations. It provides a unified way to execute a disparate set of specialized analyses and programs, reducing turnaround time, improving consistency and maintainability and allowing for what-if types of analysis. The modular architecture allows the addition of analysis modules with minimal effort. It has been deployed for use by over 600 globally distributed engineers, analysts and sales personnel and is now considered indispensable to GE wind plant siting operations.

References

[1] Ackermann T, Soder L. An overview of wind energy-status 2002. Renewable and Sustainable Energy Reviews 2002;6:67–128.
[2] Rajiv, V., Kornfein, M. "Wind turbine Fulfillment Center: A streamlined approach to knowledge capture, retention and communication for wind turbine suitability analysis", Abstract presented at Knowledge Generation Communication & Management. Orlando, FL, 2008.
[3] Kornfein, M., Rajiv, V. "Capturing Engineering Process Knowledge" To be published at Knowledge Generation Communication & Management. Orlando, FL, 2012.
[4] Sobieszczanski-Sobieski, J. Multidisciplinary Design Optimization: An Emerging New Engineering Discipline. In World Congress on Optimal Design of Structural Systems. Kluwer, August 1993.
[5] Shi, D., Yin, J., Chen, G. & Dong, J. 2005 A workflow-based distributed multi-disciplinary design optimization framework. In Proc. 9th Int. Conf. on Computer Supported Cooperative Work in Design, vol. 1, pp. 497–502. Piscataway, NJ.
[6] Zimmerman, K. A. "Integration is key to making the most of KM" , KM world, Vol 12 , No 2, 2003
[7] AppWorx, http://www.uc4.com/products/uc4-automation-platform/uc4-automation-engine.html
[8] Zentner, J, De Buets, P. W. G., Mavris D.N. "Formulation of an Integrating Framework for Conceptual Object-Oriented Systems Design", http://www.as~,gatech,edu/publications/pdt72002/SAE-2002-01-2955.pdf.
[9] EASA Software, http://www.easasoftware.com