4:th CEAS Air & Space Conference
FTF Congress: Flygteknik 2013

Ad hoc Collaborative Design with Focus on Iterative Multidisciplinary Process Chain Development applied to Thermal Management of Spacecraft

Doreen Seider, Achim Basermann, Robert Mischke, Martin Siggel, Anke Tröltzsch and Sascha Zur
Simulation and Software Technology
German Aerospace Center (DLR)
51147 Cologne, Germany

Keywords: Collaborative Design, Integration Framework, MDAO, RCE

Abstract

In multidisciplinary design optimization, different experts from different disciplines located often at different sites solve a problem in common. Efficient and effective collaboration is essential. This paper describes an approach how software technologies enable ad hoc collaboration during the development of process chains used in multidisciplinary design optimization. The German Aerospace Center develops a framework that supports this kind of ad hoc collaboration. It is used for optimizing the thermal management of spacecraft. As a result, the development phase of the process chain gets more efficient, because the technical effort of collaboration is reduced.

1 Introduction

The design study of complex objects such as aircraft or spacecraft often involves many different disciplines. The involved experts work in different organisations, which are often spread across multiple sites. Collaborative work requires lots of communication via e-mail or phone and data and tools have to be shared and exchanged. Multidisciplinary collaboration can be cumbersome due to the communication overhead.

In the initial design phase, the tools of the different disciplines are developed and an automated design process chain to couple them is set up. Tools include simulation codes, analysis applications, or optimization algorithms. At this stage, the tools are still evolving and subject to frequent changes because of bug fixes and new features. Keeping the tools of all disciplines at all sites always up to date is a challenge. The conventional approach in multidisciplinary design is collecting all required tools and running them on one dedicated compute machine. This approach can be a limitation due to

- Software licensing issues, as research tools are often based on commercial software for numerical computation and simulation, whose licenses are too expensive for an additional installation on the compute machine.
- Platform specific tools, which require specific operating systems or "exotic" hardware platforms such as compute clusters, graphic cards, accelerators etc.
- Impractical centralized installation of frequent software updates.

This paper describes an approach to support collaboration in initial design phases within distributed environments, where tools can remain at each developer's machine. The approach ensures that every involved discipline can always

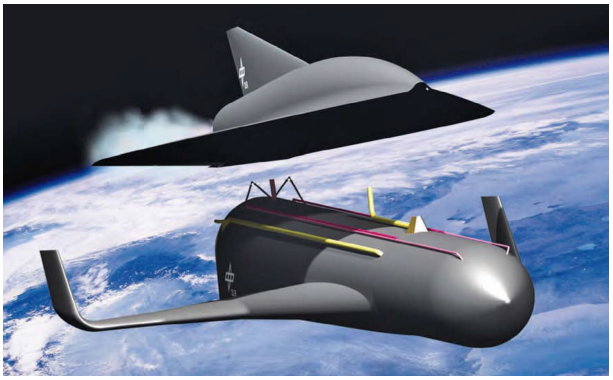

Figure 1: The SpaceLiner while separating from its main tank.

use the latest version of tools from all other disciplines. It reduces the mentioned coordination overhead between project members and allows an efficient and iterative process chain development.

In Section 2 the design optimization of the SpaceLiner [1] is introduced – a revolutionary concept between aviation travel and space travel for ultra fast passenger transport (see Fig. 1). Requirements for ad hoc collaboration are elaborated in Section 3. In Section 4 the implementation of the introduced ad hoc collaboration approach is presented.

2 Multidisciplinary Design Optimization in Thermal Management of Spacecraft

When designing spacecraft, one of the major issues is to devise its thermal management. This is particularly true for thermal protection systems (TPS) during atmospheric re-entry. One part of current DLR research focuses on the development of new hybrid structures with integrated thermal control units for future space transportation systems and hypersonic planes like the SpaceLiner [1] (see Fig. 1).

Beside passive thermal management techniques through heat sink elements consisting of materials with extremely high heat conductivity and forced radiation cooling, magnetohydrodynamic (MHD) effects with cooled magnets for heat reduction [2] are in the focus of DLR research.

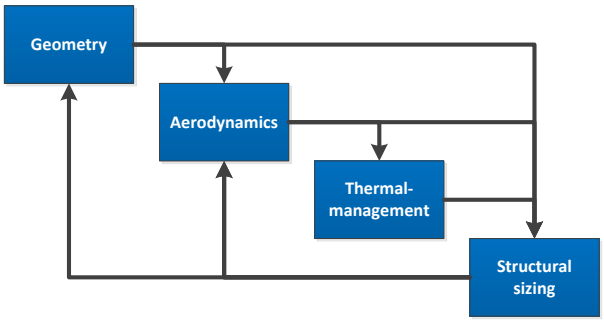

Figure 2: Involved disciplines in the MDO problem.

The goal of the overall project is to find the optimal preliminary SpaceLiner design which considers all disciplines together (see Fig. 2). The involved disciplines are:

- Geometry (e.g., maximum length or nose radius of the spacecraft),
- Aerodynamics (e.g., a solution in compliance with minimum lift and maximum drag requirements),
- Thermal management (e.g., the choice or combination of the cooling system and its parameters),
- Structural sizing (e.g., the computation of structural masses).

The objective of the optimization problem is to locate the design with the maximum glide ratio subject to an upper mass constraint and to several geometry constraints for a fixed trajectory. As the considered disciplines together build a highly coupled system and a change of input in one simulation tool affects also the output of several other ones, multidisciplinary design optimization (MDO) techniques have to be applied [3, 4]. The application of the appropriate MDO technique for the addressed optimization problem is a research topic of its own and out of the scope of this paper.

The design optimization of the SpaceLiner requires the close collaboration between experts of all disciplines, especially in the initial design phase. As most of the involved physical phenomena cannot be computed exactly but rather

have to be approximated by numerical simulation tools, tools of the mentioned disciplines are integrated in an appropriate design process chain. Especially in the initial design phase, the tools are subject to constant development and improvements, which results in challenges during the development of the process chain. Another problem arises from the fact that the involved experts do not only belong to different disciplines, but also work in different institutes located on different sites in Germany.

3 Ad hoc Collaboration in Multidisciplinary Process Chain Development

As in the thermal management of spacecraft, multidisciplinary design is often performed using automated process chains. Process chains consist of multiple tools. The tools belong to different disciplines, and therefore have dependencies between each other that must be considered during execution. To set up those process chains, integration frameworks are widely used [5]. As described in Section 2, process chains with multidisciplinary character have specific requirements regarding collaboration. Especially, this affects the phase when involved tools are being developed, and the process chain is being set up. To allow collaboration in this phase, the used framework must enable the user to use tools of other disciplines easily, in the latest version, from remote, and ad hoc. Based on these demands, requirements are derived that an integration framework must fulfill.

A user should be always able to use the latest version of a certain tool. On the other hand, the tool developer wants to provide frequent updates of his tool during the development phase. Thus, the tool developer must be able to integrate and provide his tool to others in an effective and efficient way. This can be ensured, if the integration process itself is simple and robust and if the distribution of the tool is as transparent as possible for the developer.

A user must be allowed to use tools on demand which can run remotely on different sites. This implies different requirements to the integration framework: first, a local installation of the framework must be able to connect on demand to the rest of the distributed process chain. Second, the framework must be able to execute the tools. Third, the local configuration of the framework to e.g. setup the network should be easy and transparent for the user.

In summary, a tool developer must be able to easily integrate his tool in a single local integration framework. He must be enabled to configure his framework in a way that it can connect on demand to an overall network of frameworks from experts of other involved disciplines. It must be possible for a tool user to access and execute tools as long as the associated integration framework is part of the network.

The benefits of collaboration in multidisciplinary design with the approach outlined above are:

- Control over tool under development: The tool stays all the time on the machine controlled by the developer and will not be distributed in an unmanageable way.
- Frequently centralized updates of tool: Newer versions of the tool can be provided by the developer by updating it on his local machine without additional overhead.
- Tool users can directly make use of new versions and can give feedback instantly (e.g., if dependencies to tools of other disciplines are violated or not considered properly).

For the multidisciplinary optimization of thermal management the integration framework RCE (Remote Component Environment) [6] is used from the beginning of the project. As the demand for collaboration during the iterative multidisciplinary process chain development increases, RCE was extended with the support for collaboration in that phase by considering the requirements from above.

Other integration frameworks were evaluated concerning support for multidisciplinary optimization and collaboration. is an open-source integration framework developed at NASA Glenn Research Center (GRC). It focuses on the support for multidisciplinary design analysis and optimization (MDAO) [7]. Compared to RCE it provides more powerful

and more multi-purpose optimization methods. OpenMDAO allows for execution of tools on remote machines like compute clusters. Centralized tool distribution is not addressed.

PHX ModelCenter® is an commercial integration framework developed and purchased by Phoenix® Integration. It supports distributed computing. Specific server can be set up where tools can be integrated and distributed to ModelCenter instances. This approach supports centralized tool distribution, but it is limited to distribution from server-side. From a user's machine tools can only be consumed from remote and not distributed to others.

4 Ad hoc Collaboration using RCE

Multidisciplinary design and optimization is a complex task. From a specialist's perspective, dependencies between disciplines must be understood. Based on these dependencies, the interfaces of tools must be defined and tools must be coupled correctly.

From a technical perspective, the tools, which are often located at different sites, must be executed and data must be transferred. Both the tools and the data are heterogeneous: Tools are written in different languages, or run on different operating systems. Data is represented in different formats, or values may use different measurement units.

Tools and data are continuously changing during the development phase of a multidisciplinary process chain. That leads to a demand for ad hoc collaboration (see Section 3).

In multidisciplinary optimization of thermal management, engineers prefer to focus on the specialist side of the optimization task. To reduce the technical effort to a minimum, the integration framework RCE (Remote Component Environment) is used. The following subsections introduce RCE and explain how it enables ad hoc collaboration from a technical point of view.

4.1 Distributed Multidisciplinary Process Chains

In multidisciplinary process chains, each involved discipline is represented by at least one tool. RCE allows the integration of tools, the composition of tools to process chains, and their distributed execution from one single site. RCE is a distributed system. Instances run of different machines. Each running instance is called a node in the RCE network.

RCE instances can be started with a graphical user interface (GUI), which also provides a graphical editor for defining process chains. All available tools are listed in the editor, an can be simply dragged into the process chain editor. A tool is called available if it is integrated on at least one node. When a process chain is executed, RCE executes each tool on the node where it is integrated. Tool execution ordering and data transfer are handled automatically.

4.2 Networking and Communication

A typical approach to networked computing is the client-server model, in which a central instance (the "server") controls all interactions within the network. While this approach is easy to implement, the resulting network layouts are quite static, which is not ideally suited for ad hoc collaboration. To provide more flexibility, the structure of a RCE network is based on the peer-to-peer approach. In this concept, any RCE node can choose to accept network connections from other nodes. All nodes that are directly or indirectly connected are combined into a virtual network which is independent of the underlying physical network. As communication requests between nodes without a direct connection are forwarded ("routed") by intermediate nodes, each pair of nodes can transparently communicate with each other. With the flexibility that this approach offers, both ad hoc and long-term networks can be set up easily.

One important design criterion of the current RCE network system was that as long as a physical network connection can be established in either direction, communication within the virtual network can flow in both directions. This property allows simple solutions to common networking problems. For example, a typical scenario is where members of a working group are located in different networks, each of which is protected by a firewall that allows no incoming connec-

tions. This situation can be easily handled by setting up an RCE node (called a "relay") in a public network that is reachable from each protected network. As working group members only need to make connections from their protected network to a public one, problems associated with network management are greatly reduced or even completely eliminated.

While this simple solution would be possible with a client-server approach as well, the peer-to-peer approach allows to extend this setup in many ways. For example, multiple relays can be connected to each other, which is often useful to link users or resources located at different institutes into the same virtual network.

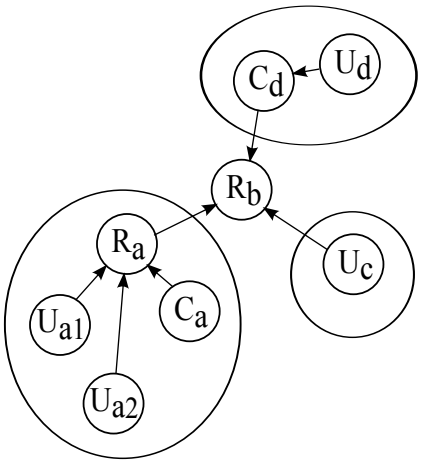

Figure 3: Network of RCE nodes configured as relays (R), compute nodes (C) and user frontend machines (U).

Fig. 3 shows such a setup, where a relay node R_a is located within the local network of an institution, and another relay R_b is set up on a publicly available machine (for example, in a DMZ).

U_{a1} and U_{a2} represent RCE installations on team member's machines in the same network as R_a. These installations only need to be configured with the information how to connect to R_a to access the whole virtual network with all connected resources, which greatly simplifies the handling of ad hoc network changes.

U_c represents a user at a different institute who chooses to connect to R_b directly, without an intermediate relay.

C_a represents a compute node located in the same network as relay R_a. By simply connecting to R_a, it can make its resources available to all members in the virtual network, including U_c. This kind of resource sharing would be difficult without RCE, as the network containing C_a does not allow incoming connections from other institutes.

C_d represents a compute node that is directly connected to the central relay as well, making its resources available in a similar way as C_a.

While this example has used the typical roles of relay, compute node and user machine so far, any combination of these can be configured as well. For example, the compute node C_d has been set up to allow incoming connections from user machines like U_d, which combines the roles of compute node and relay into one. In the same manner, user machines can act as compute nodes or as relays for colleagues as well.

Adding new remote sites or compute resources to the virtual network (for example, by connecting them to R_b as well) requires no action on part of the working group members. From their perspective, the new resources simply appear and are immediately ready to use. Similarly, new members connecting to any part of the virtual network are immediately able to consume all published resources of other members, and publish their own resources (including tools) as well.

4.3 Ad hoc Distribution of Tools

As described in Section 3, from a technical point of view a prerequisite of ad hoc collaboration in multidisciplinary process chains is a framework which allows the integration of tools. To integrate a tool into RCE following requirements must be fulfilled:

- The tool must be executable via a single command line call, and without any user interaction during execution,
- the tool's input must only be command line parameters and files,
- all input files must be located in a specific folder, and
- all output files generated by the tool must be written in a specific folder as well.

In the integration concept of RCE, an integrated tool is treated as a black box. I.e., it is seen in terms of its inputs and outputs without any knowledge of its internal working. Fig. 4 summarizes RCE's tool integration concept.

RCE executes a tool as it would be executed to run standalone. It invokes a pre-defined command line call, which usually includes the executable and optional parameters. RCE waits asynchronously for the tool's termination. During execution, it continuously fetches the standard console output generated by the tool. It provides it immediately to the user (GUI and log file) even if the tool execution is done on a remote RCE node.

Tools are developed to run standalone. If they become a part of an automated process chain extra tasks must be usually performed to fit into the chain. This includes e.g. the conversion of input files into a different file format or writing output data calculated by the tool into a shared database. As these tasks are only needed if the tool is executed within a process chain, they must not be part of the tool's logic. Therefore, RCE supports pre and post processing steps: It provides the possibility to execute two given scripts in conjunction with the tool execution itself. One script is executed right before the execution of the tool, and the other one right after the tool is finished.

An integrated tool is configured via RCE's graphical user interface (GUI). Tool-specific configuration GUIs can be defined by tool developers in specified configuration description files. Out of these files, the GUIs for the tool configuration are automatically generated during RCE's runtime.

For each tool integrated into RCE, a description file must be provided by the tool developer. All relevant information like paths to the tool's executable or to the pre/post processing scripts, the icon a tool shall be shown within the GUI, etc. must be defined there. At startup time, RCE is processing these files and then integrates the tool. If the integration of a tool was successful, a new item is added to the list of tools that are available for building process chains.

Once a tool has been integrated into one RCE instance, e.g., the tool developer's instance, it is instantly being distributed in the RCE network the correspondent node is part of. As only a descriptive information about the tool is distributed, the distribution step is fast and not expensive in terms of communication. As soon as another node has received the information, this node's user can immediately start using the tool in its process chains as if it is installed on its local machine. The actual execution of the tool is performed on the node's machine that it is integrated on. This distinction ensures that the actual tool does not need to be transferred to other machines, and that no conflicts with outdated versions of the tool can occur.

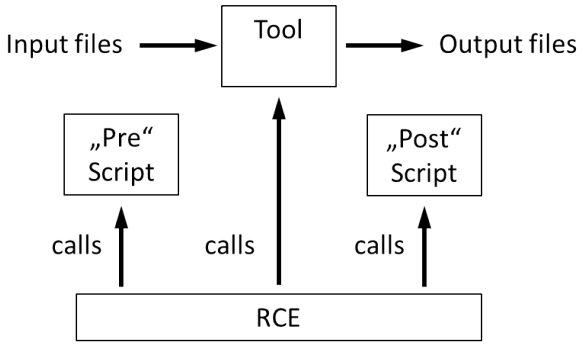

Figure 4: Tool integration concept of RCE.

The tool's executable and related files are referenced locally on the executing node's machine. If a new version of the tool is available, only updates to local files are required. From then on, only the new version of the tool will be distributed and can be used by others.

Once a tool has reached a stable state, it is usually desirable that it is available even if the tool developer's node is not connected to the RCE network. In that case the tool can be moved to a compute node, which are designed to be connected permanently (see Section 4.2).

4.4 Ad hoc Distribution of Optimization Methods

Multidisciplinary process chains are often utilized for optimization purposes. To improve the performance of the optimization, it might be required to design and implement a custom optimization method first that fits well to the specific optimization problem.

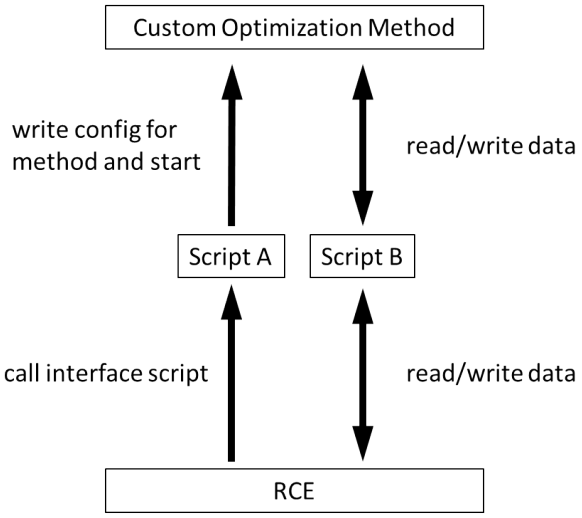

Figure 5: Optimizer integration concept of RCE.

The development of a custom optimization method is more efficient if the developer can test its method with tools the method is designed to optimize. If the developer can share the current state of the method to others, involved in the process chain, he can get feedback from them. This leads to a more effective development as well.

It is possible to integrate custom optimization methods into RCE without having to wait for a new release of RCE. The optimizer component defines a script interface which allows any user to integrate his optimization method in RCE and to distribute it to others. For doing so, there is a template folder containing two predefined scripts. The first script will be called by RCE to start the optimization method. The other script is responsible for handling the communication work, e.g. exchanging new design and response variables between RCE and the optimization method (see Fig. 5).

Using these scripts (written in Python), the user can define his own methods in any programming language that can be triggered using Python. It is also possible to define method configurations in further template files. With those, RCE will automatically generate a graphical user interface and deliver the information the user enters to the optimization method. While designing the new method, the user will be able to use it within RCE, without having to configure anything outside of the framework.

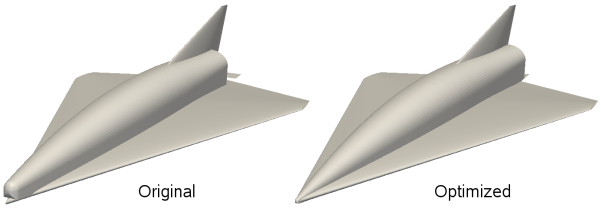

Figure 6: First results of the optimization of the SpaceLiner geometry.

The execution of the method is performed on the node's machine providing the method. Again, version conflicts are avoided. If the optimization method is available in a newer version, only files on the node's machine must be updated.

5 Multidisciplinary Design Optimization in Thermal Management of Spacecraft using RCE

In DLR, for the optimization of thermal management of the SpaceLiner RCE is used from the beginning of the project. One current setup consists of experts for the structural sizing discipline in Bremen, Germany, sharing their tools with experts from aerodynamics in Cologne, Germany, via RCE. Both groups are able to develop their tools and test them immediately within a process chain using real world data. Problems in the tools or even mistakes in the design of the process chain can be tracked down much easier and sooner than without the ad hoc collaboration support provided by RCE.

Fig. 7 illustrates the process chain we established in RCE so far. Three of the four involved disciplines mentioned in Section 2 are already integrated. First results of the optimization are available and are visualized in Fig. 6.

6 Conclusions

Using the example of optimizing the thermal management of spacecraft we demonstrated the demand of collaboration when developing tools and setting up multidisciplinary process chains. We pointed out how collaboration in that phase can be supported by integration frameworks.

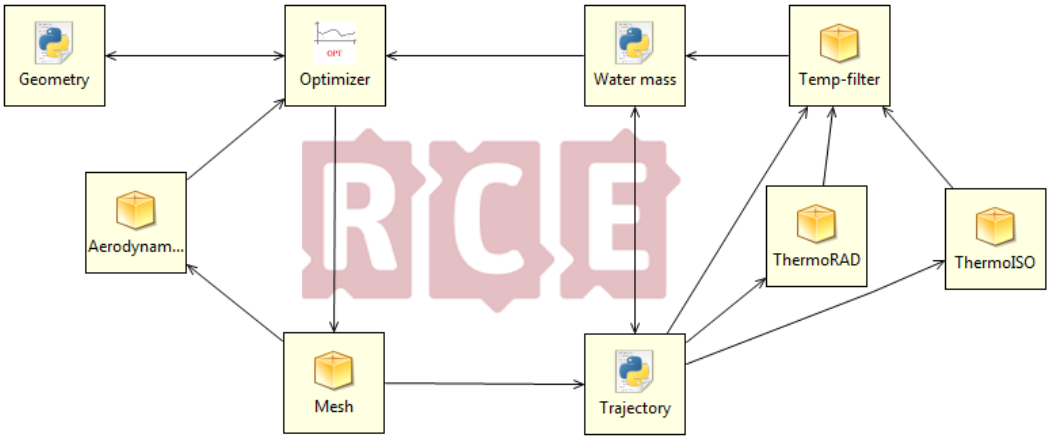

Figure 7: Current version of the process chain for optimizing the SpaceLiner's thermal management implemented in RCE.

We introduced a process chain, where we utilized the collaboration support provided by the integration framework RCE.

In the presented use case of thermal management in spacecraft design, the ad hoc collaboration approach enables efficient collaboration between experts working at different sites. The tool specialists use newly implemented optimization methods immediately. The optimization specialists develop new methods always against the latest version of the involved tools. From an end user's point of view, the introduced approach allows time savings and simplifies collaborative work.

Collaboration is always about humans. Software technologies can support collaboration but can not guarantee efficient and effective collaboration. To address the human aspect collaboration needs to be considered from a psychological point of view as well. Questions should be answered such as: Which kind of people collaborate best? Which circumstances prevent people from collaboration? What motivates people to collaborate? In the end, software technologies can be used as an enabler for collaboration again, if the answers are used to derive the right ideas for new supportive software.

Based on experiences made in multidisciplinary optimization RCE will be extended continuously. Currently, tools can be distributed from one expert to another. This does not ensure that an expert from another discipline knows how to use the tool correctly. RCE will address this issue in the future. Information about the person, who is responsible for the tool, will be distributed next to the tool itself. Documentation of the tool will be directly integrated in RCE as well as an opportunity for instant messaging.

References

[1] Sippel M. Introducing the spaceliner vision. In *7th International Symposium on Launcher Technologies, Barcelona, Spain*, 2007.

[2] Böttcher C. and Longo J. M. Simulation and analysis of magnetohydrodynamic flows using the dlr tau code – mediums air and argon. In *International ARA Days, Arcachon, France*, 2008.

[3] Martins J. R. R. and Lambe A. B. Multidisciplinary design optimization: Survey of architectures. *AIAA Journal, in press*, 2013.

[4] Tedford N. P. and Martins J. R. R. Benchmarking multidisciplinary design optimization algorithms. *Optimization and Engineering*, 11:159–83, February 2010.

[5] Bachmann A., Litz M., Kunde M., and Schreiber A. Advances in generalization and decoupling of software parts in a scientific

simulation workflow system. In *Fourth International Conference on Advanced Engineering Computing and Applications in Sciences*, 2010.

[6] Seider D., Fischer P. M., Litz M., Schreiber A., and Gerndt A. Open source software framework for applications in aeronautics and space. In *IEEE Aerospace Conference, Mountain View, MT, USA*, pages 1–11. IEEE, 2012.

[7] Justin S. Gray, Kenneth T. Moore, and Bret A. Naylor. OPENMDAO: An Open Source Framework for Multidisciplinary Analysis and Optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Fort Worth, Texas, August 2010. AIAA.