A Graph Theoretic Approach to Problem Formulation for Multidisciplinary Design Analysis and Optimization (Reviewer comments appear below in red text and the authors' responses appear in black text)

I. Reviewer 1

This idea of PSG was mentioned but was not covered in the paper. For completeness and value proposition sake, I would recommend to add at least half section to show how it will help and look like. You can take the simplest formulation of your choice and show how MCG and FPG will translate and look as a PSG. A brief discussion of advantages vs disadvantages will be useful.

We agree that the PSG was too quickly addressed in the paper. Although we feel that discussion of the PSG in significant depth would be too lengthy for the present paper, we agree that the topic does warrant a more complete discussion than that provided in the original draft.

We have added subsections 3.5 and 4.4 to address the the driver nodes and driven edges which together distinguish a PSG from an FPG. We have also added subsection 5.5, which provides an example of a simple PSG and compares and contrasts it with an XDSM diagram.

The idea of holes and collision is explained a few times in the paper. It can be reduced.

In reviewing the paper, the authors agree that the concepts of holes and collisions are discussed perhaps too often. The terms are formally defined only once in section 4.1. Section 5.3 provides a mathematical description of the set of nodes that meet the criteria for holes and edges in the context of the FPG algorithm, but it is our opinion that this discussion should stand separately from the definitions.

We have checked sections 4.1 and 5.3 and removed many instances of redundancy, including informal definitions repeated several times in the text. Nonetheless, we feel that preserving the separate mathematical discussions in sections 4.1 and 5.3 makes for a clearer message in the paper. The concept is somewhat subtle since, in our definitions, a node can be either a design variable or a hole, but not both. Similarly, a collision can occur when a variable has two incomming edges if the problem is single fidelity but not if the problem is multi-fidelity. The additional discussion is intended to clarify theses subtleties.

The review of other graph based techniques is good and highlights their limitations, but I would like to see a discussion of what challenges a practitioner may face while trying to use the approach suggested in this paper. The MCG and FPG will cover lot of information like objectives and constraints, but at what expense. Comparing XDSM with MCG or FPG, one might find XDSM as more easy to read and comprehend. For a similar size problem, FPG looks far more complex and bigger.

We agree that XDSM is more visually informative and concise. However, an easily human-readable format was not the goal of the new syntax. Instead, the goal is to make a graph structure that was easier to operate on algorithmically to achieve feasible MDO data flows. As a consequence of this goal, the graph grows larger and becomes less human readable. We believe that this tradeoff is acceptable for two reasons:

1. *It should be possible to convert between the gray lines of the XDSM format and the proposed syntax in a lossless manner. It is our conclusion that the PSG in our syntax represents a subset of the information*

in the XDSM diagram; namely, the data connectivity elements of XDSM and PSG are equivalent graphs. The implication is that it would be possible to write an algorithm and associated code to "abstract" the FPG into a partial XDSM diagram that could then be elaborated by the user to add the solution-specific elements. It may then be possible to write a PSG from an XDSM, if the additional information about all of the individual variable connections from the FPG graph were preserved in the background. (This is true for the gray lines of XDSM but not the black ones. Only data connections are shown in new syntax.)

2. *Algorithmic methods are a useful way for analyzing MDO problem structure, so it seems appropriate to have a graph structure that works well for this use, even at the expense of human readability.*

Please see the revised end of the introduction (Section 1) in which the algorithmic purpose of the new graph syntax is now stated more clearly.

If PSG is covered even to a small extent (like explained in first comment) I would suggest or like to see the PSG equivalent of problem formulation shown in Figure 2 with XDSM.

We have added Fig. 8 and Fig. 13 and associated discussion to address this request.

Is it really adding a lot of value to have graphs like MCG or FPG which are very explicit in nature and can become difficult to read

As discussed above, we believe that the value of the MCG and FPG is not derived from their ability to represent the data in a human readable form, but in their straightforward translation to machine readable formats. Section 5.3 introduces an algorithm for testing whether MDO data connectivity graphs are FPGs and for reducing graphs to FPGs. As problems grow larger and more complex, this type of algorithm is expected to become ever more valuable.

Such an algorithm could be implemented into a framework or stand-alone tool to help teams to develop the specifics of a design problem before they begin detailed development work. Section 6 shows that for even simple problems, the graph (as viewed by an algorithm) is non-trivial. It also shows that there are multiple ways to solve problems. As problems grow even a small amount from the size of the example becomes invaluable. In sum, it is our belief that the graph syntax derives its usefulness from its effectiveness in describing MDO problem graphs in computational syntax.

Can the graph be compressed in a logical way to reduce the number of nodes and their interconnections while still conveying important and critical information for that stage (MCG stage or FPG stage) ? Reducing cycles is discussed but the graph may still look complex.

Looking at these graphs (for example Fig. 8), a node for example Z2 appears several times. It may be useful to figure out holes and collision, but once it is resolved, can those nodes be consolidated to provide simpler visualization ?

There are a number of ways the graph could be simplified to make it more human readable. Variables could be combined, or they could be removed altogether such that the graph represents only component to component connections like a traditional DSM.

As discussed in the response above, it should be possible to convert the entire PSG graph into the data-connectivity part of an XDSM. This would present all of the information in a more compact and visually insightful form, e.g. an XDSM. By preserving the variable connectivity information, one could always convert back into the more verbose PSG syntax if needed. We would argue that the XDSM is more human readable, but the new syntax is a representation of how a framework tool, e.g. OpenMDAO or ModelCenter, could view the problem.

Given our goals, the verbose nature of the graphs is a necessary side effect.

As the problem size increases, how will it affect the size and complexity of such graphs.

The answer to this question is not trivial, as it depends on a number of factors such as the number of analyses and the number of variables in those analyses. However, roughly speaking, one could measure the complexity of a graph by the total number of nodes and edges. The number of nodes grows roughly linearly *with the number of analysis tools and the number of variables. The number of fixed edges grows linearly with the number of variables as well. The number of connection edges depends entirely on the specifics of the problem, but, since any variable will have only a small finite number of incoming edges (usually 1, but maybe 2 or 3 in multi-fidelity case) one can avoid an unbounded growth. Therefore, very roughly speaking, the graph will grow as O(N+M) where N is the number of variables and M is the number of analyses.*

If one assumes that large problems have hundreds of components and millions of variables, then graphs with a few million nodes and edges are possible. These numbers are very large, but are not outside of the realm of what is readily manageable and rapidly evaluated by today's graph libraries in common programming languages.

I will assume that creation of such a graph and interconnections will be expected to be done by some kind of software tool or wizard. A brief discussion about why this MCG, FPG and PSG idea is better to lead to such a system can be useful. Although this itself can be a separate research and paper, but why this idea can provide a better foundation can be valuable.

See the new section 2.4 for a short section describing our envisioned usage of the graph language. We have also mentioned in Section 5.3 that the algorithm can be directly implemented in any programming language capable of managing lists, such as Python.

Minor comments: Section 1 Line 46: I think other commercial tools such as iSight, modeFRONTIER should also be mentioned since their usage is more widespread compared to OpenMDAO.

Agreed. We have now mentioned this tools alongside OpenMDAO in Section 1.

Page 13 line 33 column 2: Typo : Step (D) now PROCEDEDS

Fixed.

Page 16 line 38 column 2: Typo : formulations involving MULTIPLES analysis .

((let's discuss))

II. Reviewer 2

Page 1, Line 46: 2nd column...check spacing after "ModelCenter".

Fixed

Page 2, Line 38: 2nd column...as your motivation was really done in the Introduction section, consider changing the heading on section 2 to "Background" or something similar. You don't really provide motivation in this section anyhow.

Section title changed to "Background", as suggested.

Page 3, Line 19, 1st column: change "weather" to "whether"

Fixed

Page 3, Line 21, 2nd column: change "descrition" to "description"

Fixed

Page 4, Line 37, 1st column: what does "REMS" stand for?

"Reconfigurability in MDO Problem Synthesis". The full name has been added to the paper.

Page 4, Line 45, 1st column: change "REMS address" to "REMS addresses"
Fixed

Page 4, Lines 35 to 55, 1st column: This paragraph is a bit awkward in structure. In the second to last sentence you state, "REMS provides syntax for variables and functions but does not facilitate inclusion of solvers or optimizers in the graph." The reader then expects your method to address these shortcomings in the next sentence but instead you state, "...the goal of allowing graphs to describe analysis couplings and more closely interact with existing design processes and MDAO frameworks," which you never really called out as a shortcoming of REMS in the first place. I think you could improve the flow/wording here.

The authors agree that this paragraph was poorly worded. We have re-worded it to attempt to make the issue with REMS more clear. The text now states (last two sentences, first column of p. 4): "... REMS does not provide a mechanism for inclusion of solvers or optimizers in the graph. This fundamentally limits REMS from describing the specifics of different solution strategies as applied to a particular problem." We have removed the oddly juxtaposed statement about the goal of our graph syntax that was noted by the reviewer from the paragraph. However, based on the comments by Reviewer 1 noted above, we have now added an additional description of how our method can capture optimizers and solvers in the context of the problem solution graph (PSG).

Line 10, 1st column, Page 10: This equation (missing a number) has a LOT of unions, variables, etc...in it. It would help tremendously to provide a verbal description of at least one of these equations to help with translation into the real world. This would be similar to what you did following the steps (1) to (4) in the next subsection.

The authors agree. We have added equation numbers (now Eqns. 8 and 9). We have also provided verbal descriptions to accompany the set notations.

Lines 5 - 12, 2nd column, Page 12: Just a general question spurred by the text here. What criteria would be applied to determine which FPG is better than another?

After considerable deliberation, we chose not to address the issue of how one should select one FPG in preference to another in the context of the present paper. The answer to this question depends heavily on the specific situation. If one is limited by run-time of the MDO workflow, a smaller FPG with fewer cycles and variables may be preferable. If one is trying to resolve a detailed physical question, e.g. subtle design changes associated with nacelle-wing interference, then it may be preferable to select a more high fidelity and computationally expensive FPG. These notions are introduced in the example problem, but further development was beyond the scope of the paper.

The MCG provides a good foundation for finding the only FPG or indicate multiple possible FPGs. The user can then select the best FPG for the present stage of the design process, but this decision can be revisited later in the process as the design goals change.

Line 57, 1st column, Page 13: change "as a input" to "as an input"
Fixed

Now for the request of more significant importance. This is coming from someone with significant experience in MDO and who uses a lot of the analysis tools you specified in your case study on a daily basis. I really liked this article but felt it was lacking some punch in the results section. This was namely in the form of quantitative results. You vaguely give some in the form of the different resulting connectivity graphs and this was good but to really drive home the value of your research to a potential user I think you could do a bit better with very minimal work. The potential user of this methodology will be concerned with one of two things: 1) minimizing computational time or 2) increasing the accuracy of the results. I think you've given examples of these two scenarios already in figures 15 and 16, so why not provide a table of the CPU times (1) and/or deviation from higher order analyses (2) for each of these graphs. Doesn't have to necessarily be these two metrics but something to really tie the output of your methodology to something tangible for comparison purposes. You could even include the metrics for a randomly obtained FPG as a baseline. Then the person *reading the article can say, "Wow, given a set of analysis tools and a general problem, I can employ this method to obtain an FPG that significantly reduces my overall computational time." or "Wow, given a set of analysis tools and a general problem, I can employ this method to obtain an FPG that significantly improves my overall computational accuracy while keeping my computational efficiency as high as possible."*

We agree that this sort of quantitative comparison would be very valuable. However, we believe that, in order not to be misleading, this comparison needs to be larger in scope than just that which could be obtained by executing the example problem given currently presented in the paper. We believe that one would need to address a wide range of problems with different graph structural characteristics and different inherent nonlinear features in the analyses in order to make statements that are more generalizable. The challenge is that quantitative measures are inherently dependent on particular details of a problem. For example, for coupled analyses, total CPU time is highly dependent on the number of iterations required to converge the system of nonlinear functions to a consistent solution within a specified tolerance. This convergence is tied very closely to the specifics of the coupling and the nonlinearity. We had previously considered the idea of selecting "optimal FPGs" in terms of aggregate metrics that could be specified a priori as meta-data for a particular analysis, but defining these metrics without considering problem coupling and nonlinearity proved especially challenging. To address this issue well would therefore necessitate the evaluation of many example problems with different characteristics. This work, while very valuable, would likely require an entire paper in its own right. We thank the reviewer for this suggestion, and we consider it to be a great topic for future work in this area.

If you are worried about length, I think the conclusions section may be cut in half and still accomplish its intended purpose.

We agree that the conclusion section could be more concise. We have shortened it by two paragraphs and improved the wording.

III. Reviewer 3

As the authors point out the design of a major product, such as a civil airliner, is complex and the level of complexity increases as design requirements change i.e. the need to reduce carbon emissions, the need to reduce maintenance downtime etc. and the number of design teams involved in creating a new product increases in number and distribution. Their paper is, I think, attempting to address the need to support design teams in confronting this situation but have not clearly defined where their work fits into the design process

Broadly speaking there are three major tasks or steps that have to be worked through when a globally distributed team is setting up a design system that allows it to effectively deploy an MDO methodology to support the design of a product such as a new civil airliner:

1. The team must decide which design tools are required at each stage of the design process as the design proceeds down the time-line this includes the data flow required between the various design centers and the design history preservation requirements,

2. This combination of requirements must be converted into a set of interacting tools together with data flow and control programs that constitute a template for a Computational Design Engine (CDE),

3. Finally this template has to be instantiated into a usable CDE software system which might take advantage of commercially available software such as that found in Simula etc.

The paper is clearly focused on step 2 above and it advances a sound argument for the use of graph theory to establish an appropriate set of outcomes from such a step and, whilst it implies the existence of the other two steps, it does not state this early enough in the paper and with sufficient detail to allow a practicing engineer to understand the role being played by the authors contribution. I would recommend that the authors clearly define the role that their graph theoretic approach plays in this triple set of steps leading to *the creation of effective CDE.*

The reviewer has correctly identified the major motivation for this work. The goal is indeed to assist engineers and teams of engineers in the process of building a model out of a set of interconnected tools. We believe that the distinction between phases 1 and 2 in the reviewer's comments are covered adequately in the the introduction and especially in the notional problem described in the indented text on p. 2. However, to clarify the distinction between phases 2 and 3 indicated by the reviewer, we modified the text following the indented text at the bottom of p. 2 to read as follows:

((copy and paste the last two paragraphs of Section 2.3))