

Determining the Order of Design Processes

Dr Norbert Doerry
September 18, 2013

PROBLEM:

Determine the order that one should conduct design activities within a design iteration to speed the overall convergence of the design. Design activities in general can be defined as the effort needed to produce design data. Design activities can incorporate the use of computer programs, manual calculations and defining assumptions. Each design activity can strongly or weakly depend on the design data of other design activities, or can depend on an Interface Control Document (ICD) defining the dependency. The goal of determining the order is to ensure that any design activity upon which another design activity strongly depends on is performed first. Unfortunately, some sets of design activities strongly depend on each other and must be solved as a block. Weak dependencies are treated less strictly. For now, the ordering of design activities does not account for weak dependencies. Rather, if data that a design activity has a weak dependency on has not been calculated in a previous block of the design iteration, then the data from the previous design iteration is used. ICDs enable each of the two design activities controlled by the ICD to proceed without knowing the design data of the other.

As an example of the methodology, the program `dtorder.exe` has been created. This program implements a Design Structure Matrix (DSM) as described in www.dsmweb.org. This program should be considered “academic” code serving as a demonstration of the concept.

Definitions:

$$x = f(x,y)$$

x = a vector of vectors. Each element of x , or x_i , is a vector describing the design data, for a design activity.

y = a vector of vectors. Each element of y , or y_i , is a vector describing a constraint. Design activities and outputs can depend on a constraint, but a constraint remains fixed during the design process.

$f(x,y)$ = a vector of design activities (tools and processes) that generate x .

$f_i(x,y)$ = a design activity or tool that generate x_i .

P = the participation matrix.

Each row of P , or P_i , corresponds to a design activity or tool to produce the i element of x , or $f_i(x,y)$.

Each element of P , or P_{ij} is an element of the set $\{0,I,W,S\}$. where

- 0 means that the design activity $f_i(x,y)$ does not depend on design data x_j
- I means that the design activity $f_i(x,y)$ relationship to design data x_j is defined through an ICD
- W means that the design activity $f_i(x,y)$ only weakly depends on design data x_j
- S means that the design activity $f_i(x,y)$ strongly depends on design data x_j

Assumption: P_{ii} does not equal S. In general, P_{ii} should be 0, but if the design activity is based on modifying the results of the previous iteration, then a W would be appropriate.

Q = the constraints matrix.

Each element of Q , or Q_{ij} is an element of the set $\{0,I,W,S\}$. where

- 0 means that the design activity $f_i(x,y)$ does not depend on constraint y_j
- I means that the design activity $f_i(x,y)$ relationship to constraint y_j is defined through an ICD
- W means that the design activity $f_i(x,y)$ only weakly depends on constraint y_j
- S means that the design activity $f_i(x,y)$ strongly depends on constraint y_j

$$z = g(x,y)$$

z = a vector of vectors. Each element of z , or z_i is a vector describing the data for an output product. Output products are a function of design data and constraints, but nothing is a function of Output products.

R = the output participation matrix. Each element of R , or R_{ij} is an element of the set $\{0,I,W,S\}$ which describes the dependency of z_i on x .

T = the output constraints matrix. Each element of T , or T_{ij} is an element of the set $\{0,I,W,S\}$ which describes the dependency of z_i on y .

The goal of the process is the development of the vector d . Each element of d consists of two elements:

d_{i0} = the block number – design activities are performed in the order of their block number

d_{i1} = the sub-block number – if 0, then all the design activities of the sub-block are all strongly dependent on each other and must be performed as a set (generally as a sub-iteration within the larger design iteration). If non-zero, then while design activities can be performed independently of each other, the preferred order, taking weak dependencies into account, is to perform them in the order of the sub-block number. (Currently in `dtorder.exe` sub-block numbers do not reflect the preferred order)

Methodology:

1. Initialize

Set the block number index counter to 1

Set all elements of d to 0

2. Find order 1 blocks

Search rows of P , only look at rows and columns for which $d_{i0} = 0$, but ignore the diagonal.

Set the subblock index counter to 0

For each row, count the number of S are in columns where $d_{i0} = 0$

For any row without any S's,

 set d_{i0} to the block number index counter

 increment the subblock index counter

 set d_{i1} to the subblock number index counter

If any rows were found without any S's

 Increment the block number index counter

 Repeat step 2.

3. Find order 2 blocks

Search rows of P , only look at rows and columns for which $d_{i0} = 0$, but ignore the diagonal.

Keep a list of rows that only have one S in any columns for which $d_{i0} = 0$

If the list has fewer than 2 rows, then there are no order 2 blocks, go on to step 4.

Check to see if the list is self contained. Eliminate any row that has an S in a column that corresponds to a row that is not in the list. Each time you eliminate a row, all the previous rows have to be rechecked.

If the list has fewer than 2 rows, then there are no order 2 blocks, go on to step 4.

If the list has exactly 2 rows, then you have a solution, mark them as the next block and go back to step 1 to start looking for order 1 blocks again.

3a. If the list has more than 2 rows, then have to start trial solutions.

 Start at the first row in the list, mark that row, and all the rows it depends on.

 For each marked row, ensure all the rows that it depends on are also marked.

 If you have more than 2 marked rows, then the first row can not be part of the solution.

 Mark it for deletion and all the rows that depend on it for deletion.

 For each marked rows, ensure all rows that depend on it are also marked for deletion.

 Delete from the list all the rows marked for deletion.

If there are fewer than 2 rows remaining, then there are no order 2 blocks, go to step 4.

If there are exactly 2 rows remaining, you have a solution, mark them as the next block and go back to step 1 to start looking for order 1 blocks again

If there are still more than 2 rows remaining, Go back to step 3a.

4. Continue looking for blocks of increasing orders (using the general algorithm of step 3) until all elements of d_{i0} are non-zero. Once a block of larger order is found, must go back and look for blocks of size 1 and progress incrementally to larger numbers.

Notes:

Under certain conditions, blocks with a higher block index may be started earlier in parallel with blocks with lower block indexes. For the rows of P corresponding to the block, the variables corresponding to columns that contain an S must have been previously defined or part of the block. To find the actual predecessor block, look in all the rows of P . Search all the columns not in the block for an “S”. Find the column with an S and the largest block number. This block number corresponds to the predecessor activity for the block.

It may be beneficial to reorder P so the rows and columns are in the order of the block number / sub-block number. Doing so will result in the “S” elements being predominately in the lower left diagonal of the reordered P matrix. The only “S” elements in the upper right diagonal of the reordered P matrix will be part of a block.

To solve the blocks of order > 1 , it may be beneficial to have each of the involved design activities produce a response surface for the involved variables. The solution to the block would be the intersection points of the multiple response surfaces.

Tool Considerations

Data file structure

`dtorder.exe` uses the following XML Syntax as an input filename

```
<design_process>
  <name> design_process_name </name>
  <version> version number </version>
  <version_date> version date </version_date>
  <study_name> study name </study_name>

  <comment> </comment> or <c> </c>

  <design_activity>
    <name> design_activity_name </name>
    <data> design_data_name </data>
    <strong> design_data_name</strong>           : can have multiple
    <weak> design_data_name</weak>             : can have multiple
```

<ICD> design_data_name</ICD> : can have multiple

<comment> </comment> or <c> </c>

</design_activity>

: can have multiple design activity blocks

<output>

<name> </name>

<weak> </weak>

<icd> </icd>

<comment> </comment> or <c> </c>

</output>

<constraint>

<data> </data>

<comment> </comment> or <c> </c>

</constraint>

</design_process>

Design Tool Flowchart

- Initialize
- Read Command Line
- Read XML file
- Process XML file and Construct P and d arrays
- Identify the blocks and subblocks
- Sort P
- Identify the predecessor blocks
- Produce the output file

Syntax of dtorder.exe

```
dtorder input.xml -ooutfile -fXXX -d
```

input.xml is the input filename

outfile is the output filename (or stdout if not specified)

format is CSV, XML, or TXT (TXT is default)

d = set debug mode

Sample Input file

```
<design_process>
<name> Brown-DDG51-Revisit </name>
<version> 1.0 </version>
<version_date> May 8, 2008 </version_date>

<comment> NEJ 2007 Vol 199 No. 3 Revisiting DDGX/DDG-51 Concept Exploration by
Justin Stepanchick and Alan Brown </comment>

<study_name> Revisit DDGX-DDG-51 Concept Exploration </study_name>

<design_activity>
  <name> SCInput-Mod</name>
  <data> SCInput </data>
  <weak> MOGO </weak>
</design_activity>

<design_activity>
  <name>SCCombat-Mod </name>
  <data> SCCombat </data>
  <strong> SCInput</strong>
</design_activity>

<design_activity>
  <name> SCPropulsion-Mod </name>
  <data> SCPropulsion </data>
  <strong> SCInput </strong>
</design_activity>

<design_activity>
  <name> SCHull-Mod</name>
  <data> SCHull </data>
  <strong> SCInput</strong>
</design_activity>

<design_activity>
  <name>SCSpaceA-Mod </name>
  <data>SCSpaceA </data>
  <strong> SCInput</strong>
  <strong> SCPropulsion</strong>
  <strong> SCHull</strong>
  <strong> SElectric </strong>
</design_activity>

<design_activity>
  <name> SElectric-Mod </name>
  <data> SElectric </data>
  <strong>SCInput </strong>
  <strong>SCCombat </strong>
  <strong>SCPropulsion </strong>
  <strong>SCHull </strong>
  <strong>SCSpaceA </strong>
</design_activity>

<design_activity>
  <name> Resistance-Mod </name>
  <data> Resistance </data>
  <strong> SCInput </strong>
  <strong> SCPropulsion </strong>
  <strong> SCHull </strong>
  <strong> SElectric </strong>
</design_activity>

<design_activity>
  <name> SCWeight-Mod </name>
  <data> SCWeight </data>
  <strong>SCInput </strong>
  <strong>SCCombat </strong>
```

```

        <strong>SCPropulsion </strong>
        <strong>SCHull </strong>
        <strong>SCSpaceA </strong>
        <strong>SCElectric</strong>
        <strong>Resistance </strong>
</design_activity>
<design_activity>
  <name> SCTankage-Mod</name>
  <data> SCTankage </data>
  <strong>SCInput </strong>
  <strong>SCCombat </strong>
  <strong>SCPropulsion </strong>
  <strong>SCElectric</strong>
  <strong>Resistance </strong>
  <strong>SCWeight </strong>
</design_activity>
<design_activity>
  <name> SCSpaceR-Mod</name>
  <data> SCSpaceR </data>
  <strong>SCInput </strong>
  <strong>SCCombat </strong>
  <strong>SCPropulsion </strong>
  <strong>SCSpaceA </strong>
  <strong>SCElectric</strong>
  <strong>SCTankage </strong>
  <strong> SCCost</strong>
  <strong> Feasibility</strong>
</design_activity>
<design_activity>
  <name> Feasibility-Mod</name>
  <data> Feasibility </data>
  <strong>SCInput </strong>
  <strong>SCPropulsion </strong>
  <strong>SCSpaceA </strong>
  <strong>SCElectric</strong>
  <strong>Resistance </strong>
  <strong>SCWeight </strong>
  <strong>SCTankage </strong>
  <strong>SCSpaceR </strong>
  <strong> SCOMOE</strong>
</design_activity>
<design_activity>
  <name> SCCost-Mod</name>
  <data> SCCost </data>
  <strong>SCInput </strong>
  <strong>SCCombat </strong>
  <strong>SCPropulsion </strong>
  <strong>SCElectric</strong>
  <strong>SCWeight </strong>
  <strong>SCTankage </strong>
</design_activity>
<design_activity>
  <name>SCOMOE-Mod </name>
  <data> SCOMOE </data>
  <strong>SCInput </strong>
  <strong>SCCombat </strong>
  <strong>SCPropulsion </strong>
  <strong>SCHull </strong>
  <strong>Resistance </strong>
  <strong>SCWeight </strong>
  <strong>SCTankage </strong>
  <strong> Feasibility</strong>

```

```
</design_activity>

<design_activity>
  <name> SCRisk-Mod</name>
  <data> SCRisk  </data>
  <strong>SCInput </strong>
</design_activity>

<design_activity>
  <name> MOGO-Mod</name>
  <data>MOGO  </data>
  <strong> Feasibility</strong>
  <strong> SCCost</strong>
  <strong> SCOMOE</strong>
</design_activity>

</design_process>
```


Sample TXT output file

```

dtorder version      = 1.0
dtorder version date = May 17 2008
XML Input Filename  = test1.xml
Design Process Name = Brown-DDG51-Revisit
Version Number      = 1.0
Version Date        = May 8, 2008
Study Name          = Revisit DDGX-DDG-51 Concept Exploration
File Creation Time  = Sat May 17 20:55:56 2008

```

0	SCInput-Mod	SCInput	1	1	0	000000000000000W
1	SCCombat-Mod	SCCombat	2	1	1	S000000000000000
2	SCPropulsion-Mod	SCPropulsion	2	2	1	S000000000000000
3	SCHull-Mod	SCHull	2	3	1	S000000000000000
4	SCRisk-Mod	SCRisk	2	4	1	S000000000000000
5	SCElectric-Mod	SCElectric	3	0	2	SSSS00S00000000
6	SCSpaceA-Mod	SCSpaceA	3	0	2	S0SS0S000000000
7	Resistance-Mod	Resistance	4	1	3	S0SS0S000000000
8	SCWeight-Mod	SCWeight	5	1	4	SSSS0SSS0000000
9	SCTankage-Mod	SCTankage	6	1	5	SSS00S0SS000000
10	SCCost-Mod	SCCost	7	1	6	SSS00S00SS00000
11	Feasibility-Mod	Feasibility	8	0	7	S0S00SSSSS00SS0
12	SCMOE-Mod	SCMOE	8	0	7	SSSS000SSS0S000
13	SCSpaceR-Mod	SCSpaceR	8	0	7	SSS00SS00SSS000
14	MOGO-Mod	MOGO	9	1	8	0000000000SSS00

BLOCK Nbr 1 : Depends on Predecessor BLOCK Nbr 0

Each design activity of the block can be performed independently

Design Activity SCInput-Mod Produces Data SCInput and depends on:
 (WEAK last iteration) MOGO (from design_activity MOGO-Mod)

BLOCK Nbr 2 : Depends on Predecessor BLOCK Nbr 1

Each design activity of the block can be performed independently

Design Activity SCCombat-Mod Produces Data SCCombat and depends on:
 (STRONG) SCInput (from design_activity SCInput-Mod)

Design Activity SCPropulsion-Mod Produces Data SCPropulsion and depends on:
 (STRONG) SCInput (from design_activity SCInput-Mod)

Design Activity SCHull-Mod Produces Data SCHull and depends on:
 (STRONG) SCInput (from design_activity SCInput-Mod)

Design Activity SCRisk-Mod Produces Data SCRisk and depends on:
 (STRONG) SCInput (from design_activity SCInput-Mod)

BLOCK Nbr 3 : Depends on Predecessor BLOCK Nbr 2

All design activities of the block must be solved together

Design Activity SCElectric-Mod Produces Data SCElectric and depends on:
 (STRONG) SCInput (from design_activity SCInput-Mod)
 (STRONG) SCCombat (from design_activity SCCombat-Mod)
 (STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
 (STRONG) SCHull (from design_activity SCHull-Mod)
 (STRONG) SCSpaceA (from design_activity SCSpaceA-Mod)

Design Activity SCSpaceA-Mod Produces Data SCSpaceA and depends on:

(STRONG) SCInput (from design_activity SCInput-Mod)
(STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
(STRONG) SCHull (from design_activity SCHull-Mod)
(STRONG) SCElectric (from design_activity SCElectric-Mod)

BLOCK Nbr 4 : Depends on Predecessor BLOCK Nbr 3

Each design activity of the block can be performed independently

Design Activity Resistance-Mod Produces Data Resistance and depends on:
(STRONG) SCInput (from design_activity SCInput-Mod)
(STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
(STRONG) SCHull (from design_activity SCHull-Mod)
(STRONG) SCElectric (from design_activity SCElectric-Mod)

BLOCK Nbr 5 : Depends on Predecessor BLOCK Nbr 4

Each design activity of the block can be performed independently

Design Activity SCWeight-Mod Produces Data SCWeight and depends on:
(STRONG) SCInput (from design_activity SCInput-Mod)
(STRONG) SCCombat (from design_activity SCCombat-Mod)
(STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
(STRONG) SCHull (from design_activity SCHull-Mod)
(STRONG) SCElectric (from design_activity SCElectric-Mod)
(STRONG) SCSpaceA (from design_activity SCSpaceA-Mod)
(STRONG) Resistance (from design_activity Resistance-Mod)

BLOCK Nbr 6 : Depends on Predecessor BLOCK Nbr 5

Each design activity of the block can be performed independently

Design Activity SCTankage-Mod Produces Data SCTankage and depends on:
(STRONG) SCInput (from design_activity SCInput-Mod)
(STRONG) SCCombat (from design_activity SCCombat-Mod)
(STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
(STRONG) SCElectric (from design_activity SCElectric-Mod)
(STRONG) Resistance (from design_activity Resistance-Mod)
(STRONG) SCWeight (from design_activity SCWeight-Mod)

BLOCK Nbr 7 : Depends on Predecessor BLOCK Nbr 6

Each design activity of the block can be performed independently

Design Activity SCCost-Mod Produces Data SCCost and depends on:
(STRONG) SCInput (from design_activity SCInput-Mod)
(STRONG) SCCombat (from design_activity SCCombat-Mod)
(STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
(STRONG) SCElectric (from design_activity SCElectric-Mod)
(STRONG) SCWeight (from design_activity SCWeight-Mod)
(STRONG) SCTankage (from design_activity SCTankage-Mod)

BLOCK Nbr 8 : Depends on Predecessor BLOCK Nbr 7

All design activities of the block must be solved together

Design Activity Feasibility-Mod Produces Data Feasibility and depends on:
(STRONG) SCInput (from design_activity SCInput-Mod)
(STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
(STRONG) SCElectric (from design_activity SCElectric-Mod)
(STRONG) SCSpaceA (from design_activity SCSpaceA-Mod)

(STRONG) Resistance (from design_activity Resistance-Mod)
 (STRONG) SCWeight (from design_activity SCWeight-Mod)
 (STRONG) SCTankage (from design_activity SCTankage-Mod)
 (STRONG) SCOMOE (from design_activity SCOMOE-Mod)
 (STRONG) SCSpaceR (from design_activity SCSpaceR-Mod)

Design Activity SCOMOE-Mod Produces Data SCOMOE and depends on:

(STRONG) SCInput (from design_activity SCInput-Mod)
 (STRONG) SCCombat (from design_activity SCCombat-Mod)
 (STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
 (STRONG) SCHull (from design_activity SCHull-Mod)
 (STRONG) Resistance (from design_activity Resistance-Mod)
 (STRONG) SCWeight (from design_activity SCWeight-Mod)
 (STRONG) SCTankage (from design_activity SCTankage-Mod)
 (STRONG) Feasibility (from design_activity Feasibility-Mod)

Design Activity SCSpaceR-Mod Produces Data SCSpaceR and depends on:

(STRONG) SCInput (from design_activity SCInput-Mod)
 (STRONG) SCCombat (from design_activity SCCombat-Mod)
 (STRONG) SCPropulsion (from design_activity SCPropulsion-Mod)
 (STRONG) SCElectric (from design_activity SCElectric-Mod)
 (STRONG) SCSpaceA (from design_activity SCSpaceA-Mod)
 (STRONG) SCTankage (from design_activity SCTankage-Mod)
 (STRONG) SCCost (from design_activity SCCost-Mod)
 (STRONG) Feasibility (from design_activity Feasibility-Mod)

BLOCK Nbr 9 : Depends on Predecessor BLOCK Nbr 8

Each design activity of the block can be performed independently

Design Activity MOGO-Mod Produces Data MOGO and depends on:

(STRONG) SCCost (from design_activity SCCost-Mod)
 (STRONG) Feasibility (from design_activity Feasibility-Mod)
 (STRONG) SCOMOE (from design_activity SCOMOE-Mod)

Sample CSV output file

```
"dtorder version"," 1.0"
"dtorder version date"," May 17 2008"
"XML Input Filename"," test1.xml"
"Design Process Name"," Brown-DDG51-Revisit"
"Version Number"," 1.0"
"Version Date"," May 8, 2008"
"Study Name"," Revisit DDGX-DDG-51 Concept Exploration"
"File Creation Time"," Sat May 17 13:42:29 2008"

" Design Activity Number"," Design Activity"," Design Data"," Block Number"," Subblock
Number"," Predecessor Block Number"," SCInput"," SCCombat"," SCPropulsion"," SCHull","
SCRisk"," SCElectric"," SCSpaceA"," Resistance"," SCWeight"," SCTankage"," SCCost","
Feasibility"," SCOMOE"," SCSpaceR"," MOGO"
0," SCInput-Mod" ," SCInput",1,1,0 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,W
1," SCCombat-Mod" ," SCCombat",2,1,1 ,S,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2," SCPropulsion-Mod" ," SCPropulsion",2,2,1 ,S,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3," SCHull-Mod" ," SCHull",2,3,1 ,S,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4," SCRisk-Mod" ," SCRisk",2,4,1 ,S,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
5," SCElectric-Mod" ," SCElectric",3,0,2 ,S,S,S,S,0,0,S,0,0,0,0,0,0,0,0,0,0,0
6," SCSpaceA-Mod" ," SCSpaceA",3,0,2 ,S,0,S,S,0,S,0,0,0,0,0,0,0,0,0,0,0,0,0
7," Resistance-Mod" ," Resistance",4,1,3 ,S,0,S,S,0,S,0,0,0,0,0,0,0,0,0,0,0,0,0
8," SCWeight-Mod" ," SCWeight",5,1,4 ,S,S,S,S,0,S,S,S,0,0,0,0,0,0,0,0,0,0,0
9," SCTankage-Mod" ," SCTankage",6,1,5 ,S,S,S,0,0,S,0,S,S,0,0,0,0,0,0,0,0,0,0,0
10," SCCost-Mod" ," SCCost",7,1,6 ,S,S,S,0,0,S,0,0,S,S,0,0,0,0,0,0,0,0,0,0,0
11," Feasibility-Mod" ," Feasibility",8,0,7 ,S,0,S,0,0,S,S,S,S,S,0,0,S,S,0
12," SCOMOE-Mod" ," SCOMOE",8,0,7 ,S,S,S,S,0,0,0,S,S,S,0,S,0,0,0
13," SCSpaceR-Mod" ," SCSpaceR",8,0,7 ,S,S,S,0,0,S,S,0,0,S,S,S,0,0,0
14," MOGO-Mod" ," MOGO",9,1,8 ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,S,S,S,0,0
```

Sample XML output file

```
<design_tool_order>
<dtorder_version>1.0</dtorder_version>
<dtorder_version_date>May 17 2008</dtorder_version_date>
<xml_filename>test1.xml</xml_filename>
<name>Brown-DDG51-Revisit</name>
<version>1.0</version>
<version_date>May 8, 2008</version_date>
<study_name>Revisit DDGX-DDG-51 Concept Exploration</study_name>
<file_creation_time>Sat May 17 13:39:12 2008</file_creation_time>
<block>
<block_number>1</block_number>
<predecessor_block>0</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>SCInput-Mod</name>
<Data>SCInput</Data>
<weak>MOGO</weak>
</design_activity>
<block_size>1</block_size>
</block>
<block>
<block_number>2</block_number>
<predecessor_block>1</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>SCCombat-Mod</name>
<Data>SCCombat</Data>
<strong>SCInput</strong>
</design_activity>
<design_activity>
<name>SCPropulsion-Mod</name>
<Data>SCPropulsion</Data>
<strong>SCInput</strong>
</design_activity>
<design_activity>
<name>SCHull-Mod</name>
<Data>SCHull</Data>
<strong>SCInput</strong>
</design_activity>
<design_activity>
<name>SCRisk-Mod</name>
<Data>SCRisk</Data>
<strong>SCInput</strong>
</design_activity>
<block_size>4</block_size>
</block>
<block>
<block_number>3</block_number>
<predecessor_block>2</predecessor_block>
<block_type>dependent</block_type>
<design_activity>
<name>SCElectric-Mod</name>
<Data>SCElectric</Data>
<strong>SCInput</strong>
<strong>SCCombat</strong>
<strong>SCPropulsion</strong>
<strong>SCHull</strong>
<strong>SCSpaceA</strong>
</design_activity>
<design_activity>
<name>SCSpaceA-Mod</name>
<Data>SCSpaceA</Data>
<strong>SCInput</strong>
<strong>SCPropulsion</strong>
<strong>SCHull</strong>
<strong>SCElectric</strong>
</design_activity>
<block_size>2</block_size>
```

```

</block>
<block>
<block_number>4</block_number>
<predecessor_block>3</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>Resistance-Mod</name>
<Data>Resistance</Data>
<strong>SCInput</strong>
<strong>SCPropulsion</strong>
<strong>SCHull</strong>
<strong>SCelectric</strong>
</design_activity>
<block_size>1</block_size>
</block>
<block>
<block_number>5</block_number>
<predecessor_block>4</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>SCWeight-Mod</name>
<Data>SCWeight</Data>
<strong>SCInput</strong>
<strong>SCCombat</strong>
<strong>SCPropulsion</strong>
<strong>SCHull</strong>
<strong>SCelectric</strong>
<strong>SCSpaceA</strong>
<strong>Resistance</strong>
</design_activity>
<block_size>1</block_size>
</block>
<block>
<block_number>6</block_number>
<predecessor_block>5</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>SCTankage-Mod</name>
<Data>SCTankage</Data>
<strong>SCInput</strong>
<strong>SCCombat</strong>
<strong>SCPropulsion</strong>
<strong>SCelectric</strong>
<strong>Resistance</strong>
<strong>SCWeight</strong>
</design_activity>
<block_size>1</block_size>
</block>
<block>
<block_number>7</block_number>
<predecessor_block>6</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>SCCost-Mod</name>
<Data>SCCost</Data>
<strong>SCInput</strong>
<strong>SCCombat</strong>
<strong>SCPropulsion</strong>
<strong>SCelectric</strong>
<strong>SCWeight</strong>
<strong>SCTankage</strong>
</design_activity>
<block_size>1</block_size>
</block>
<block>
<block_number>8</block_number>
<predecessor_block>7</predecessor_block>
<block_type>dependent</block_type>
<design_activity>
<name>Feasibility-Mod</name>
<Data>Feasibility</Data>

```

```

<strong>SCInput</strong>
<strong>SCPropulsion</strong>
<strong>SCElectric</strong>
<strong>SCSpaceA</strong>
<strong>Resistance</strong>
<strong>SCWeight</strong>
<strong>SCTankage</strong>
<strong>SCOMOE</strong>
<strong>SCSpaceR</strong>
</design_activity>
<design_activity>
<name>SCOMOE-Mod</name>
<Data>SCOMOE</Data>
<strong>SCInput</strong>
<strong>SCCombat</strong>
<strong>SCPropulsion</strong>
<strong>SCHull</strong>
<strong>Resistance</strong>
<strong>SCWeight</strong>
<strong>SCTankage</strong>
<strong>Feasibility</strong>
</design_activity>
<design_activity>
<name>SCSpaceR-Mod</name>
<Data>SCSpaceR</Data>
<strong>SCInput</strong>
<strong>SCCombat</strong>
<strong>SCPropulsion</strong>
<strong>SCElectric</strong>
<strong>SCSpaceA</strong>
<strong>SCTankage</strong>
<strong>SCCost</strong>
<strong>Feasibility</strong>
</design_activity>
<block_size>3</block_size>
</block>
<block>
<block_number>9</block_number>
<predecessor_block>8</predecessor_block>
<block_type>independent</block_type>
<design_activity>
<name>MOGO-Mod</name>
<Data>MOGO</Data>
<strong>SCCost</strong>
<strong>Feasibility</strong>
<strong>SCOMOE</strong>
</design_activity>
<block_size>1</block_size>
</block>
</design_tool_order>

```