Paper JMP01

When Little Objective Data Are Available, Find Root Causes and Effects with Interrelationship Digraphs and JMP®

Melvin Alexander, Social Security Administration, Baltimore, MD

ABSTRACT

The Interrelationship Digraph (ID) is one of seven new Quality Management and Planning tools described by Shigeru Mizuno (*Management for Quality Improvement: The New 7 QC Tools*, Cambridge, MA, Productivity Press, Inc., 1988).

IDs show cause-and-effect relationships between several items, ideas, factors, or issues. IDs are useful in prioritizing choices when decision makers find it difficult to reach consensus; and sorting out issues involved in project planning, especially when credible data may not exist.

IDs provide a means of evaluating ways in which disparate ideas influence one another. IDs: (1) make it easy to spot leading factors that affect other factors; (2) blend Cause-and-Effect thinking from Ishikawa diagrams and the creative logic of brainstorming; and (3) respond to frequent complaints made about Cause-and-Effect diagrams: “What are the most important causes among many choices available?” and “How do they interact or connect to each other?”

This presentation will show how JMP® can construct IDs to identify and process ideas that drive process improvement efforts. JSL scripts will create customized reports combining traditional graphic and matrix ID formats.

INTRODUCTION

This presentation is about the Interrelationship Digraph (ID) or Relations Diagram tool. The ID is an important quality management and planning (MP) tool designed to clarify intertwined, causal relationships among a group of items, issues, problems, or opportunities. This tool helps analysts gain insights into potential complex relationships of root causes that may underlie recurring problems despite efforts to resolve them.

The ID is a special network visualization that consists of a set of nodes (or vertices) connected by arrows. Arrows show directional relationships between “source” (sender) nodes into “target” (receiver) nodes. This representation turns the ID into a form of Social Network Analysis, where connections and interactions between items, objects, and systems are made.

This paper includes background information about IDs; a description of methods used in constructing two standard forms of IDs using JMP; and conclusions.

BACKGROUND

The ID is one of the seven Management & Planning Tools (7-MP) originally developed in 1976 by the collaboration between the Union of Japanese Scientists and Engineers (JUSE) and the Society of Quality Control Technique Development. Shigeru Mizuno [1] published information about IDs and six other tools in 1988 as a collection of methods for engineering, economic planning, and management. The other New QC tools are: Affinity Diagrams (a.k.a. KJ Method), Systematic (tree) diagrams, Matrix diagrams, Matrix Data Analysis, Process Decision Program Charts (PDPC), and Arrow diagrams.

IDs belong to a class of knowledge-based quality tools used in identifying and processing ideas that lead to a better understanding of causal relationships within the “Define” phase of the Six Sigma Define, Measure, Analyze, Improve, Control (DMAIC) framework.

IDs outdo the capability of Ishikawa (Fishbone, Cause-and-Effect) Diagrams. Ishikawa Diagrams assume that causal relationships between items are totally independent of each other, which usually is not the case. Analysts use IDs to determine which set of causes has the most influence on effects and how causes interact (or interrelate) with each other.

Since IDs are knowledge-based, they provide a preliminary step before data collection, experimentation, and root-cause verification process occurs in later MAIC Six Sigma stages

IDs graphically show the logical, causal relationships between several factors, issues, or ideas. They are useful in prioritizing choices when executives and organizations have difficulty reaching consensus, especially when credible data are not available.

The steps in creating IDs are:

(1) Knowledgeable experts from cross-functional departments agree on the issue to be resolved or question to be answered.
(2) Each member writes down the choices, factors, ideas, or items on index cards or sticky notes that affect the issue or question.
(3) The team arranges the items from step (2) in an octagonal or circular shape.
(4) Members pit each choice, or item against each other, individually or collectively. Next, they consider all combinations of the choices or items.
(5) For each pair of connected items, each member is asked which item influences (is stronger or causes) the other item and by how much on a degree of strength scale of 1 for weak or equal strength; 3 if the relationship has medium strength; or 9 if the relationship is strong or very strong.
(6) Members draw outbound one-headed arrows from "source" items to "target" items where a relationship between them exists or where one item influences the other item.
(7) Finally, the team adds the number of items that have the most inbound arrows (as key outcomes or results) and most outgoing arrows (as root causes or drivers).

IDs are presented in two basic formats: the Traditional Directed Graph (Digraph) and the Matrix format.

- The Traditional Directed Graph format shows how each item is connected to every item by arrows, either going out to (or coming in from) other items. The Traditional graphic format is visually more familiar to most people.
- The Matrix Format ID is a tabular form of the Traditional ID that uses the numerical values of “-1", "0", or “+1" to join adjacent nodes. JMP's JSL matrix operators express the Traditional ID into a format that is more systematic and manageable.

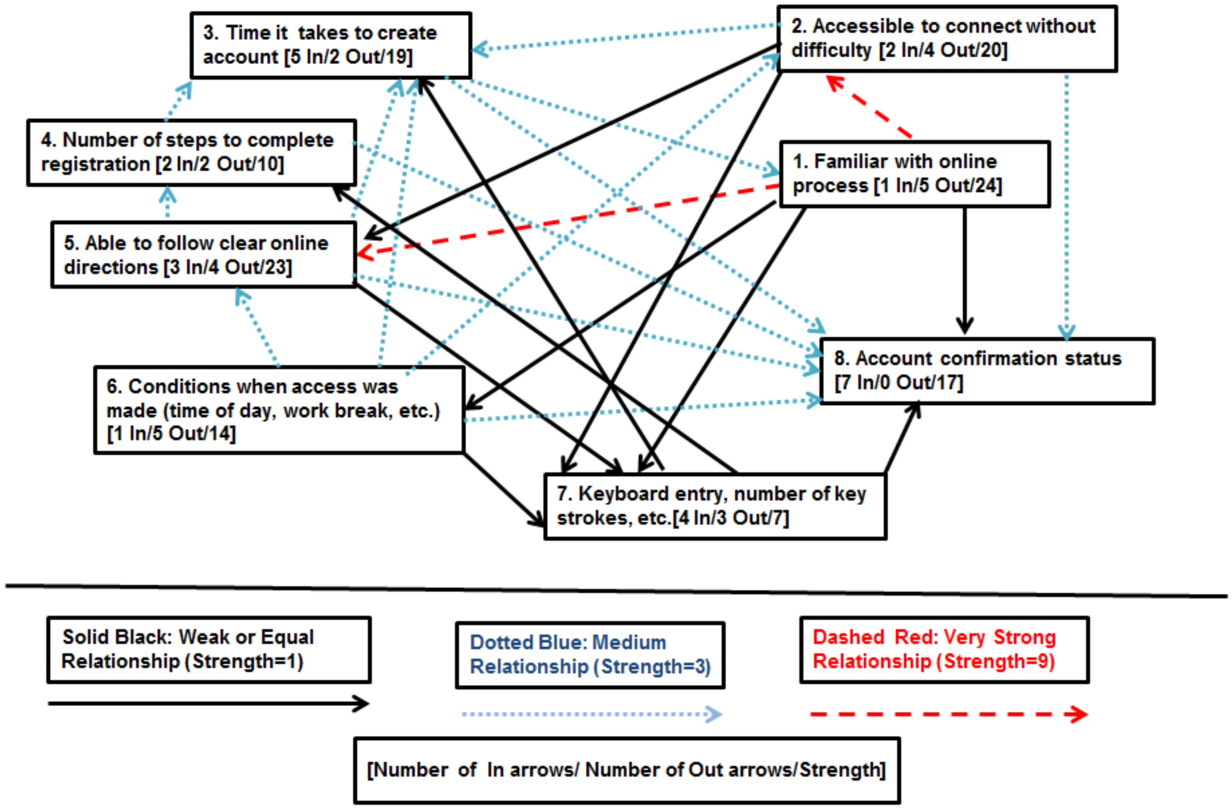
IDs in JMP use JSL operators, functions, and other utilities to create both formats. The Traditional graphic format uses Associative Arrays, Depth-Search-First functions, and graphic box operations. The Matrix format uses the matrix algebraic operators in JSL.

EXAMPLE

An example where IDs was applied was the following: A public service organization wanted customers to minimize office visits by getting them to set up and use online services. A Quality improvement team produced an Interrelationship Digraph to identify and understand the factors thought to be important in the successful creation of online accounts. Figure 1 shows the Traditional Format ID the team produced.

Paper JMP01

FIGURE 1: INTERRELATIONSHIP DIGRAPH OF FACTORS AFFECTING SUCCESSFUL CREATION OF ONLINE ACCOUNTS (TRADITIONAL FORMAT)



The team listed eight factors as important for successful online account creation:
Customers being familiar with the online process (Factor 1), Customers Access to connect online without difficulty (Factor 2), The time it takes to create online accounts (Factor 3), Number of steps to complete registration (Factor 4), Customers ability to follow clear online directions (Factor 5), Conditions when access was made, e.g., during workbreak, time of day, etc. (Factor 6), Keyboard entry, number of keystrokes, etc. (Factor 7), and getting Account confirmation status (Factor 8).

The team determined that Familiarity with the online process (Factor 1) had a very strong, causal influence on accessibility to connect without difficulty (Factor 2) and the ability to follow clear online directions (Factor 5) – indicated by the dashed red arrows. Thirteen medium strength relationships (dotted blue arrows) were established between factors. Ten equal strength relationships (solid black arrows) were displayed.

Table 1 shows the Relations diagram in a matrix format. Incoming and outgoing arrows are represented as “-1” and “+1” in the adjacent node pairs (connected factors) for row item i and column item j. No connection between nodes is represented as “0”. Factors with the sum of the incoming arrows and outgoing arrows are listed in the “Total In(-1)” and “Total Out(+1)” columns, respectively. The Strength column vector was computed using JMP’s JSL matrix product operations of the interrelationship and strength-of-relationship matrices.

Paper JMP01

TABLE 1: INTERRELATIONSHIP DIGRAPH OF FACTORS AFFECTING THE SUCCESSFUL CREATION OF ONLINE ACCOUNTS (MATRIX FORMAT)

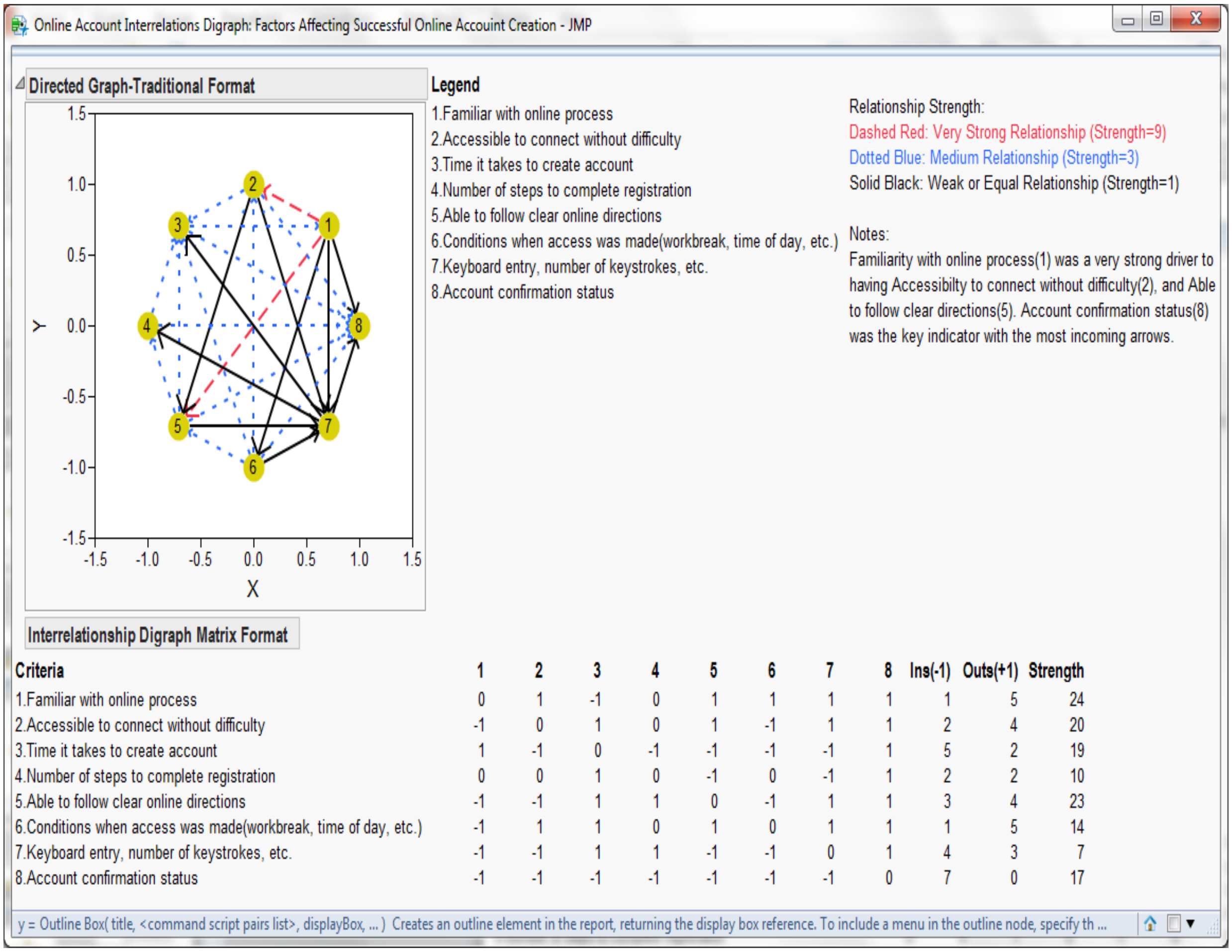
Online Account Interrelations Digraph: Factors Affecting Successful Online Account Creation											
	1.Familiar with online process	2.Accessible to connect without difficulty	3.Time it takes to create account	4. Number of steps to complete registration	5.Able to follow clear online directions	6.Conditions when access was made (workbreak, time of day, etc.)	7.Keyboard entry, number of keystrokes, etc.	8.Account confirmation status	Total In(-1)	Total Out(+1)	Strength
1.Familiar with online process	0	1	-1	0	1	1	1	1	1	5	24
2.Accessible to connect without difficulty	-1	0	1	0	1	-1	1	1	2	4	20
3.Time it takes to create account	1	-1	0	-1	-1	-1	-1	1	5	2	19
4. Number of steps to complete registration	0	0	1	0	-1	0	-1	1	2	2	10
5.Able to follow clear online directions	-1	-1	1	1	0	-1	1	1	3	4	23
6.Conditions when access was made (workbreak, time of day, etc.)	-1	1	1	0	1	0	1	1	1	5	14
7.Keyboard entry, number of keystrokes, etc.	-1	-1	1	1	-1	-1	0	1	4	3	7
8.Account confirmation status	-1	-1	-1	-1	-1	-1	-1	0	7	0	17

The matrix format of Table 1 shows that Familiarity with the online process (Factor 1) was a very strong driver to Accessibility to connect without difficulty (Factor 2) and being Able to follow clear directions (Factor 5). Familiarity with the online process and the Conditions when access was made (Factor 6) had the most outgoing arrows ("Total Out(+1)" = 5). The key indicator or effect was Account confirmation status (Factor 8) having the most incoming arrows ("Total In(-1)" = 7). The "key driver" among all factors was Familiarity with the online process (Factor 1) since it had the most outgoing arrows ("Total Out(+1)" = 5) and the maximum "Strength" value of 24.

Figure 2 shows the combined ID in both graphical and matrix formats in JMP.

Paper JMP01

FIGURE 2: COMBINED INTERRELATIONSHIP DIGRAPH IN TRADITIONAL AND MATRIX FORMATS IN JMP



Criteria	1	2	3	4	5	6	7	8	Ins(-1)	Outs(+1)	Strength
1.Familiar with online process	0	1	-1	0	1	1	1	1	1	5	24
2.Accessible to connect without difficulty	-1	0	1	0	1	-1	1	1	2	4	20
3.Time it takes to create account	1	-1	0	-1	-1	-1	-1	1	5	2	19
4.Number of steps to complete registration	0	0	1	0	-1	0	-1	1	2	2	10
5.Able to follow clear online directions	-1	-1	1	1	0	-1	1	1	3	4	23
6.Conditions when access was made(workbreak, time of day, etc.)	-1	1	1	0	1	0	1	1	1	5	14
7.Keyboard entry, number of keystrokes, etc.	-1	-1	1	1	-1	-1	0	1	4	3	7
8.Account confirmation status	-1	-1	-1	-1	-1	-1	-1	0	7	0	17

The next section will describe how Figure 2 was produced.

CREATING INTERRELATIONSHIP DIGRAPHS USING JMP®

TRADITIONAL DIGRAPH FORMAT

First, an Associative Array h, called the containing array, maps a set of keys (source nodes) to the set of values (target nodes). The Associative Array defines the data structure for the Directed Graph (Digraph). The numbers inside the brackets ([ ]) on the left-side of the equal sign are the source nodes. These source nodes connect to other target nodes on the right-side of the equal sign, inside the braces ({ }) of the Associative Array function list.
For more information about Associative Arrays, see the Section on "Associative Arrays in Graph Theory" in the JMP Scripting Guide [2], version 10.0.2.

The node keys of h[ ] that have the most outgoing arrows into the Associative Array list, { }, denote the key causes or “driver” nodes. Target nodes with the fewest values in the Associative Array { } list are factors with the most incoming arrows or leading effects.

Below are The Associative Array statements in JMP JSL

```
h = Associative Array();
h[1] = Associative Array ({2, 5, 6, 7, 8}) ;
h[2] = Associative Array({3, 5, 7, 8});
h[3] = Associative Array({1, 8});
h[4] = Associative Array({3, 8});
h[5] = Associative Array({3, 4, 7, 8} );
h[6] = Associative Array({2, 3, 5, 7, 8 });
h[7] = Associative Array({3, 4, 8});
h[8] = Associative Array({} );
```

Second, the depth-first search function (dfs) statements, below, sets the mapping structure between nodes and edges from which the digraph is drawn. Arguments in the dfs function tell which Associative Array is used (`ref`); the starting-point node (`node`); and a vector that tracks the number of nodes visited (`visited`) as the digraph is traversed when the graph is drawn.

```
dfs = Function( {ref, node, visited},
Local( {chnode, tmp},
Write( "Node: " || Char( node ) || ", " || Char( ref[node] << get
contents ) || "\!N" );
visited[node] = 1;
tmp = ref[node];
chnode = tmp << first;
While( !Is Missing( chnode ),
If( !visited[chnode],
visited = Recurse( ref, chnode, visited )
);
chnode = tmp << next( chnode );
);
visited; ) );

dfs( h, 1, J( N Items( h << get keys ), 1, 0 ) );
```

The Local () function lists the local variables used throughout the open JMP session.

Third, the nodecolor, linesty, and fillclr expressions are external expressions used to assign attributes to arrows and nodes (e.g., pen colors line styles, and group nodes) that draw the graphs produced by the idplot expression. Fillclr will be described later in the **ONE MORE THING** section.

```
/* nodecolor assigns pen colors based on strength of relationships
   connecting source nodes (i) and and target nodes (edge) where
   "red"=very strong, "blue"=medium, "black"=weak or equal
*/

nodecolor = expr(if ( i == 1 & edge == 2,"Red",   i == 1 & edge == 5,"Red",
                    i == 1 & edge == 6,"Black", i == 1 & edge == 7,"Black",
                    i == 1 & edge == 8,"Black", i == 2 & edge == 3,"Blue",
                    i == 2 & edge == 5,"Black", i == 2 & edge == 7,"Black",
                    i == 2 & edge == 8,"Blue",  i == 3 & edge == 1,"Blue",
                    i == 3 & edge == 8,"Blue",  i == 4 & edge == 3,"Blue",
                    i == 4 & edge == 8,"Blue",  i == 5 & edge == 3,"Blue",
                    i == 5 & edge == 4,"Blue",  i == 5 & edge == 7,"Black",
                    i == 5 & edge == 8,"Blue",  i == 6 & edge == 2,"Blue",
                    i == 6 & edge == 3,"Blue",  i == 6 & edge == 5,"Blue",
                    i == 6 & edge == 7,"Black", i == 6 & edge == 8,"Blue",
                    i == 7 & edge == 3,"Black", i == 7 & edge == 4,"Black",
                    i == 7 & edge == 8,"Black", "Black") );
```

```
/* linesty assigns line styles based on strength of relationships
   connecting source nodes (i) and and target nodes (edge) where
   "Dashed"=very strong, "Dotted"=medium, "Solid"=weak or equal
*/
linesty = expr(if (i == 1 & edge == 2,"Dashed", i == 1 & edge == 5,"Dashed",
                 i == 1 & edge == 6,"Solid",  i == 1 & edge == 7,"Solid",
                 i == 1 & edge == 8,"Solid",  i == 2 & edge == 3,"Dotted",
                 i == 2 & edge == 5,"Solid",  i == 2 & edge == 7,"Solid",
                 i == 2 & edge == 8,"Dotted", i == 3 & edge == 1,"Dotted",
                 i == 3 & edge == 8,"Dotted", i == 4 & edge == 3,"Dotted",
                 i == 4 & edge == 8,"Dotted", i == 5 & edge == 3,"Dotted",
                 i == 5 & edge == 4,"Dotted", i == 5 & edge == 7,"Solid",
                 i == 5 & edge == 8,"Dotted", i == 6 & edge == 2,"Dotted",
                 i == 6 & edge == 3,"Dotted", i == 6 & edge == 5,"Dotted",
                 i == 6 & edge == 7,"Solid" , i == 6 & edge == 8,"Dotted",
                 i == 7 & edge == 3,"Solid",  i == 7 & edge == 4,"Solid",
                 i == 7 & edge == 8,"Solid",  "Solid") );

fillclr = expr("Yellow") ;
```

Fourth, The idplot expression statements plot the Interrelationship Digraph to the display box window. Different colors and line styles represent the strength of relationship between nodes to form a "weighted" digraph.

```
idplot= expr(Graph Box(
        Frame Size( 300, 300 ),
        X Scale( -1.5, 1.5 ),
        Y Scale( -1.5, 1.5 ),
        Local( {n = N Items( h ), k = 2 * Pi() / n, r, i, pt, from, to, edge, v,
d},
       Line Style(0);
      Pen Size( 3 );
      r = 1 / (n + 2);
      For( i = 1, i <= n, i++,
          pt = Eval List( {Cos( k * i ), Sin( k * i )} );
          edges = h[i];
      For( edge = edges << first, !Is Empty( edge ), edge = edges <<
Next( edge ),
      to = Eval List( {Cos( k * edge ), Sin( k * edge )} );
      If( i == edge,
      Circle( Eval List( 1.2 * pt ), 0.9 * r ), // else
      v = pt - to;
      d = Sqrt( Sum( v * v ) );
    {from, to} = Eval List(
    {pt * (d - r) / d + to * r / d, pt * r / d + to * (d - r) /
d}
);

Pen Color(nodecolor);
Line Style(linesty);
     Arrow( from, to );
)
);
Fill Color (fillclr) ;
     Circle( pt, r, "fill" );
```

```
     Text( Center Justified, pt - {0, 0.05}, Char( i ) );
)
)));
```

The nodecolor and linesty Conditional if statements assign pen colors and line styles based on the strength of relationships connecting source nodes (i) and target nodes (edge).

Red-dashed arrows denote very strong relationships (Strength=9). Blue-dotted arrows denote medium strong relationships (Strength=3). Black-solid arrows denote weak or equally strong relationships (Strength=1) between nodes.

Additional colors and line styles may be used to represent ordinal scales of strength. The 9, 3, 1 scales are standard scales used by most quality and six-sigma belt practitioners.

"Unweighted" digraphs use the same solid-black pen color and line style. The “weighted” arrows are used when the graph is drawn. The Pencolor and Line Style statements draw the arrows connecting the nodes based on the conditional if statements defined earlier above.

MATRIX FORMAT

The interrelations_digraph (ID) expression takes the Adjacency matrix, k=[A(i,j)], and strength of relationship matrix, w, to form a full matrix (fullmatrx). Fullmatrx concatenates matrix k with the sum of incoming arrows (sumin), outgoing arrows (sumout), and the strength vector (c). The strength vector, c, is determined by computing the diagonal elements of the product of two matrices, absolute value of matrix k and the strength of relationship matrix w using the JSL Vecdiag () function.

Note: The matrix k has similarities to directed path effects of structural equation models (SEMs). The strength relationship matrix w would represent latent effects variables in SEMs.

IDs has graphical similarities to SEMs used in Path Analysis and Directed Acyclic Graphs (DAGs) of Social Network Analysis.

```
// Interrelationship Digraph
// Matrix Format
// input matrix of in arrows(-1),
//out arrows(+1), and no arrows(0)
interrelations_digraph=expr(
// initialize sumin and sumout vectors
sumin = J(1, nrow(k), 0); sumout = J(1, nrow(k), 0);
// compute sumin(-1) and sumout (+1) vectors
sumin= (V sum((k < 0)`))` ;
sumout= (V sum((k > 0)`))` ;
// compute strength vector c
c = vecdiag(abs(k)*w) ;
// concatenate ins, outs, and strength vectors
//in idmatrix
idmatrix = sumin||sumout||c ;
//concatenate k and idmatrix into fullmatrx
fullmatrx = k||idmatrix ; );
```

INPUT MATRICES AND INTERRELATIONS_DIGRAPH EXPRESSION CALL

The statements below forms the Matrix ID and Strength-of-Relationship matrices in JSL. The matrix k expresses the adjacency matrix, k = [A(i,j)], where cell A(i,j) = -1 means that row item i has an incoming arrow from column item j; A(i,j) = 0 means no connection exists between row item i and column item j; A(i,j) = +1 means that row item i has an outgoing arrow into column item j.

The strength of relationship matrix, w, has values of 0 (no relationship), 1 (equal relationship), 3 (medium strength relationship), or 9 (very strong relationship) between the row item i-column item j node pairs. The rnames and cnames lists assign the row and column names for the nodes before calling the interrleations_diagram expression.

```
k= [ 0  1 -1  0  1  1  1  1,  -1  0  1  0  1 -1  1  1,
     1 -1  0 -1 -1 -1 -1  1,   0  0  1  0 -1  0 -1  1,
    -1 -1  1  1  0 -1  1  1,  -1  1  1  0  1  0  1  1,
    -1 -1  1  1 -1 -1  0  1,  -1 -1 -1 -1 -1 -1 -1  0];
// create strength of relationship matrix
//(0 = no relationship), (1= weak or equal relationship),
// (3= medium relationship),(9= strong relationship)
w = [0 9 3 0 9 1 1 1, 9 0 3 0 1 3 1 3,
     3 3 0 3 3 3 1 3, 0 0 3 0 3 0 1 3,
     9 1 3 3 0 3 1 3, 1 3 3 0 3 0 1 3,
     1 1 1 1 1 1 0 1, 1 3 3 3 3 3 1 0];

rnames = {"1.Familiar with online process",
          "2.Accessible to connect without difficulty",
          "3.Time it takes to create account",
          "4.Number of steps to complete registration",
          "5.Able to follow clear online directions",
          "6.Conditions when access was made(workbreak, time of day, etc.)",
          "7.Keyboard entry, number of keystrokes, etc.",
          "8.Account confirmation status" };

cnames = {"1", "2", "3", "4", "5", "6", "7","8", "Ins(-1)", "Outs(+1)","Strength"};

interrelations_digraph ;
```

SENDING MATRIX FORMAT TO THE REPORT WINDOW

The dt2rptb expression sends the results matrix from the interrelations_digraph expression to a report window..

This expression also works if one wanted to send a JMP data table to a report window. One would not want to send a large data table to a report window. A large data table would best be sent to a JMP journal.

```
//copy data table or matrix to the report window as a display box
dt2rptb=Expr(
// ncbx is the expression to interatively create number column boxes
// for the output display
ncbx=Expr(Number Col Box (cnames[i],fullmatrx[0,i]));
n=n items(cnames); /* count the number of matrix items */

nwdw=New Window("Table Matrix",
        OutlineBox("Interrelationship Digraph Matrix Format"),
           TableBox(
              String Col Box("Criteria",rnames),
                   hb = HlistBox(textbox("")),
                     for (i=1, i<=n, i++,hb<<append(ncbx)); ) ) );
```

COMBINE TRADITIONAL AND MATRIX FORMATS TO A NEW WINDOW

The script below attaches the original data, graphs, and analyses into the single report shown in Figure 2.

```
/*******************************************************
Append the Digraph, Legend, and Matrix objects into the
Combination ID/Matrix format window
******************************************************/
```

```
idwdw=New Window( "Online Account Interrelations Digraph: Factors Affecting Successful
Online Accouint Creation",
   HListBox( OutlineBox("Directed Graph-Traditional Format", Idplot  ),
   OutlineBox("",
   TableBox(
         String Col Box("Legend",rnames),
         VListBox(
          Text Box(" "),
          Text Box("  Relationship Strength:",
          << Font Color("Black")),
          Text Box("  Dashed Red: Very Strong Relationship (Strength=9)",
          << Font Color("Red")),
          Text Box("  Dotted Blue: Medium Relationship (Strength=3)",
          << Font Color("Blue")),
          Text Box("  Solid Black: Weak or Equal Relationship (Strength=1)",
          << Font Color("Black")),
          Text Box(" "),
          Text Box("  Notes:"),
          Text Box("  Familiarity with online process(1) was a very strong driver to",
          << Font Color("Black")),
          Text Box("  having Accessibilty to connect without difficulty(2), and Able",
          << Font Color("Black")),
          Text Box("  to follow clear directions(5). Account confirmation status(8)",
          << Font Color("Black")),
          Text Box("  was the key indicator with the most incoming arrows.",
          << Font Color("Black"))))) ), dt2rptb) ;
/* adjust the frame box line width scales so that arrows look more like dashes and
dots in the idwdw object */

 idwdw[framebox(1)] <<  Line Width Scale( 0.67 );
```

Figure 2 shows the sum of the incoming arrows ("Ins(-1)"), outgoing arrows ("Outs(+1)"), and Strength vector were computed from the input matrices and matrix operators of the interrelations_digraph expression.

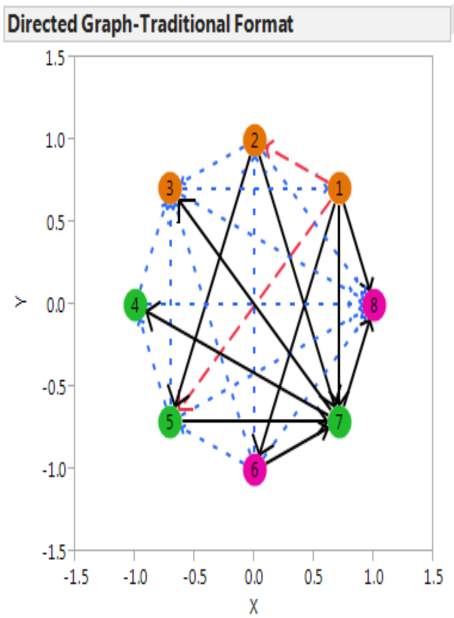
The report window identifies the key driver, "Familiarity with the online process", as the main focus of the QI team before pilot testing or developing the online system. The key effect or outcome, "Online Account confirmation status", appears to be the major indicator of online account setup success.

The benefit of the ID here is in identifying and processing ideas after brainstorming or affinity diagraming which groups the ideas as the first step in the ID "Define" process.

ONE MORE THING

If one wanted to group items according to different account types the following example set-up could be used. For example, Factors 1, 2, and 3 could represent an "Account Access" Group. Factors 4, 5, and 7 could refer to an "Account Set-up" Group. Factors 6 and 8 could be classified as the "Account Conditions and Status" Group. The group nodes could be assigned different colors to indicate the three group memberships shown in Figure 3.

FIGURE 3: COMBINED INTERRELATIONSHIP DIGRAPH IN TRADITIONAL AND MATRIX FORMATS IN JMP



Legend

1.Familiar with online process
2.Accessible to connect without difficulty
3.Time it takes to create account
4.Number of steps to complete registration
5.Able to follow clear online directions
6.Conditions when access was made(workbreak, time of day, etc.
7.Keyboard entry, number of keystrokes, etc.
8.Account confirmation status

Relationship Strength:
Dashed Red: Very Strong Relationship (Strength=9)
Dotted Blue: Medium Relationship (Strength=3)
Solid Black: Weak or Equal Relationship (Strength=1)

Notes:
Familiarity with online process(1) was a very strong driver to having Accessibilty to connect without difficulty(2), and Able to follow clear directions(5). Account confirmation status(8) was the key indicator with the most incoming arrows.

Circle Color Groups:
Orange referred to Account Access
Green denoted Account Set up
Fuchsia represented Account Conditions and Status

Interrelationship Digraph Matrix Format

Criteria	1	2	3	4	5	6	7	8	Ins(-1)	Outs(+1)	Strength
1.Familiar with online process	0	1	-1	0	1	1	1	1	1	5	24
2.Accessible to connect without difficulty	-1	0	1	0	1	-1	1	1	2	4	20
3.Time it takes to create account	1	-1	0	-1	-1	-1	-1	1	5	2	19
4.Number of steps to complete registration	0	0	1	0	-1	0	-1	1	2	2	10
5.Able to follow clear online directions	-1	-1	1	1	0	-1	1	1	3	4	23
6.Conditions when access was made(workbreak, time of day, etc.	-1	1	1	0	1	0	1	1	1	5	14
7.Keyboard entry, number of keystrokes, etc.	-1	-1	1	1	-1	-1	0	1	4	3	7
8.Account confirmation status	-1	-1	-1	-1	-1	-1	-1	0	7	0	17

Two changes need to be made to produce Figure 3. Replace the fillclr expression on page 7

```
fillclr = expr("Yellow") ;
```

with the expression:

```
/* fillclr assigns the circle fill colors for source nodes based on
   the groups like nodes belong to Orange-Account Access; Green-Account
   Set up; Fuchsia-Account Conditions and Status
*/

fillclr = expr( if (i == 1, "Orange", i == 2, "Orange", i == 3, "Orange",
 i == 4, "Green", i == 5, "Green", i == 6, "Fuchsia",   i == 7, "Green",
 i == 8, "Fuchsia", "Yellow" ) );
```

This change fills each node with different colors. Next, in the idwdw Window, add the code snippet inside the Text Box statements (shown by ❶ below) to the VlistBox display list so that the Color Group Legend appears in Figure 3.

```
idwdw=New Window( "Online Account Interrelations Digraph: Factors Affecting Successful
Online Accouint Creation",
       HListBox( OutlineBox("Directed Graph-Traditional Format",
       idplot), OutlineBox("",
       TableBox(
         String Col Box("Legend",rnames),
         VListBox(
         Text Box(" "),
         Text Box("  Relationship Strength:",
            << Font Color("Black")),
         Text Box("  Dashed Red: Very Strong Relationship (Strength=9)",
            << Font Color("Red")),
         Text Box("  Dotted Blue: Medium Relationship (Strength=3)",
            << Font Color("Blue")),
         Text Box("  Solid Black: Weak or Equal Relationship (Strength=1)",
            << Font Color("Black")),
         Text Box(" "),
         Text Box("  Notes:"),
         Text Box("  Familiarity with online process(1) was a very strong driver to",
            << Font Color("Black")),
         Text Box("  having Accessibilty to connect without difficulty(2), and Able",
            << Font Color("Black")),
         Text Box("  to follow clear directions(5). Account confirmation status(8)",
            << Font Color("Black")),
          Text Box("  was the key indicator with the most incoming arrows.",
             << Font Color("Black"))
      ❶ ,
          Text Box(" "),
          Text Box("  Circle Color Groups:",
            << Font Color("Black")),
         Text Box("  Orange referred to Account Access",
            << Font Color("Orange")),
         Text Box("  Green denoted Account Set up",
            << Font Color("Green")),
         Text Box("  Fuchsia represented Account Conditions and Status",
            << Font Color("Fuchsia"))

) ) ) ), dt2rptb) ;
```

CONCLUSIONS

IDs are special cases of Directed Acyclic Graphs (DAGs) used in Graph Theory and Social Network Analysis. IDs are powerful tools that: (1) are useful in analyzing qualitative, “idea” data (i.e., data collected from experiences, opinions, creative thoughts of a team working towards solving a problem, answering a question, or developing a goal or mission); (2) transcend cause and effect diagrams by enabling teams to systematically depict complex relationships between different issues; and (3) force teams to establish logical, sequential connections between products, processes, and related ideas.

The JMP ID Tools (Associative Arrays, DFS, matrix operators, expressions, and scripts) allow teams to produce combined Traditional Graphic and Matrix IDs in a more structured and organized manner that interrelate different ideas teams may likely face during six-sigma project planning.

The resulting report windows help teams focus on the more important causes and effects, enabling them to explore future quality improvement opportunities.

Key drawbacks of IDs are that identified items may not be complete and are dependent on the knowledge of team members creating the ID. Therefore, having team members who are most knowledgeable; representing the diverse perspectives, viewpoints, and functions of the organization; and are familiar with the needs of stakeholders and customers will help minimize the shortcomings in constructing IDs.

IDs developed in the manner described above set the stage for forming “useful” models that establish functional relationships between process input causal factors and process effects (outcomes) that can be studied in later six sigma MAIC stages.

IDs lead into tracing the known reasons of causal relationships using designed experiments and other Lean Six Sigma (LSS) methods like Failure-Mode-and-Effects Analysis (FMEA), Fault Tree Analysis (FTA), Reliability Block Diagrams (RBD), 5-Whys, Root Cause Analysis (RCA), and Is-Is Not Analysis. For situations when the reasons for root causes are unknown, IDs are ideal for finding root causes and effects. See [3-9] for more information.

REFERENCES

1. S. Mizuno, 1988, *Management for Quality Improvement: The Seven New QC Tools*, Productivity Press: Cambridge, MA.

2. SAS Institute Inc. 2012. *JMP® 10 Scripting Guide*, Cary, NC: SAS Institute Inc., pp.175-178.

3. R.W. Hoerl and R.D. Snee, 2012, *Statistical Thinking: Improving Business Performance, 2nd edition*, Wiley: Hoboken, NJ, pp. 109, 126, 185-188.

4. A,M. Doggett, 2005, “Root Cause Analysis: A Framework for Tool Selection”, *Quality Management Journal, 12:4*, pp. 34-45.

5. W.J. Michalski and D.G. King, 2005, *Six Sigma Tool Navigator © The Master Guide for Teams*, (http://flylib.com/books/en/2.890.1.227/1/, accessed 10/18/2012).

6. M. Brassard, 1996, *The Memory Jogger Plus+®: Featuring the Seven Management and Planning Tools*. GOAL/QPC: Methuen, MA, pp. 41-71.

7. Business Centered Learning, 2012. *Coach’s Guide to the Interrelationship Digraph*, (http://www.bclearning.com/resources/coach_digraph.pdf, accessed 03/04/2013).

8. L. Boisvert, 2004. “Successful Relationship Diagrams,“ *Quality Progress*, January, p. 104.

9. I.C. Allen, 2013, “EXPERT ANSWERS: Which tool to use?,” *Quality Progress*, April, p. 8.

ACKNOWLEDGMENTS

I thank Tonya Mauldin of JMP Technical Support for her contributions and Lucia Ward-Alexander for her review and editorial assistance.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
Melvin Alexander
Social Security Administration
6401 Security Blvd. ; East High Rise Building (5-A-10)
Baltimore, MD 21235
Phone: (410) 966-2155
Fax: (410) 966-4337
E-mail: Melvin.Alexander@ssa.gov