Measures for ORA (Organizational Risk Analyzer)

Jeffrey Reminga
Kathleen M. Carley
Carnegie Mellon University
March 22, 2003

ORA is the organizational risk analyzer. Its purpose is to assess the level of possible organizational risk and the factors that contribute to this risk. All measures are based on the meta-matrix and take in to account the relations among personnel, knowledge, resources and tasks. These measures are based on work in social networks, operations research, organization theory, knowledge management, and task management. As ORA is a product in development, additional measures will be added.

ORA runs on a PC running windows 2000 or XP operating system. The system interface is in JAVA and the measures are a combination of C and C++.

ORA takes as input one or more matrices in the meta-matrix for an organization and then calculates the measures herein.

Acknowledgement.
This work has benefited from research and metric development by Carter Butts, Mike Ashworth, Craig Schreiber, LiChiou Chen and David Krackhardt.

The research reported herein was supported in part by the National Science Foundation NSF IRI9633 662, NSF IGERT 9972762, the Office of Naval Research (ONR), Office of Naval Research Grant No. N00014-97-1-0037 and 9620.1.1140071, and NASA.

Additional support was provided by CASOS - the center for Computational Analysis of Social and Organizational Systems at Carnegie Mellon University (http://www.casos.ece.cmu.edu).

Suggestions for additional measures should be sent to Kathleen M. Carley, kathleen.carley@cmu.edu

A **network** N consists of two sets of nodes, called U and V, and a set $E \subset UxV$. An element e = (i,j) in E indicates there exists a relationship or tie between nodes $i \in U$ and $j \in V$. A network where U=V and therefore $E \subset VxV$, is called a **square network**; otherwise the network is a **rectangular network**. In square networks, $(i,i) \notin E$ for $i \in V$, that is, there are no self-loops.

An **organization** is a collection of networks. A **measure** is a function that maps one or more networks to R^n. Measures are often either scalar valued (real or binary) or vector valued (real or binary with dimension |U| or |V|).

When defining or implementing measures, a network can be represented as (1) a graph or as (2) an adjacency matrix. To represent a *square* network as a graph, let G=(V,E), where V is the network's nodes, and E are the ties; *rectangular* networks will not be represented as graphs. Both square and rectangular networks are represented as adjacency matrices. Given a network N=((U,V),E), define a matrix M of dimension |U|x|V|, and let M(i,j) = 1 iff $(i,j) \in E$. Then M is the adjacency matrix representation of N. Note that since a square network has no self-loops, its adjacency matrix representation has a zero diagonal.

The adjacency matrices of an organization's networks is called the MetaMatrix for the organization. The following adjacency matrices for the most common networks are used throughout the measures documentation:

A = *Communication Network*: element (i,j) is the degree to which agent i communicates with agent j
AK = *Knowledge Network*: element (i,j) is the degree to which agent i knows knowledge j
AR = *Capabilities Network*: element (i,j) is the degree to which agent i owns resource j
AT = *Assignment Network*: element (i,j) is the degree to which agent i is assigned to task j
K = *Information Network*: element (i,j) is the degree to which knowledge i is connected to knowledge j
KR = *Training Network*: element (i,j) is the degree to which knowledge i is needed to use resource j
KT = *Knowledge Requirement Network*: element (i,j) is the degree to which knowledge i is needed to do task j
R = *Resource Substitute Network*: element (i,j) is the degree to which resource i can be substituted for resource j
RT = *Resource Requirement Network*: element (i,j) is the degree to which resource i is needed to do task j
T = *Precedence Network:* element (i,j) is the degree to which task i must be done before task j

The matrices **A**,**K**,**R**,**T** are square networks; the others are rectangular networks.

The following matrix notation is used:

|Matrix| = dimension of a *square* Matrix (i.e. if Matrix has dimension r x r, then |Matrix| = r)
Matrix(i,j) = the entry in the i^{th} row and j^{th} column of Matrix
Matrix(i,:) = i^{th} row vector of Matrix
Matrix(:,j) = j^{th} column vector of Matrix
sum(Matrix) = sum of the elements in Matrix (also, Matrix can be a row or column vector of Matrix)
Matrix' = the transpose of Matrix
~Matrix = for binary Matrix, ~Matrix(i,j) = 1 iff Matrix(i,j) = 0.
Matrix@Matrix = element-wise multiplication of two matrices (e.g. C=A@B => C(i,j) = A(i,j)*B(i,j))

These mathematical terms and symbols are used:

card(Set) = |Set| = the cardinality of Set
sgn(x) = 1 if x >= 0, and -1 otherwise
$\Re$ denotes a real number
Z denotes an integer

These graph theoretic terms are used:

$d_G(i,j)$ is the length of the shortest directed path in G from node i to node j. Note that if there is a path from i to j in G, then $1 \leq d_G(i,j) < |V|$.
Therefore, let $d_G(i,j)$ = |V| if there is no path in G from i to j. Also, let $d_G(i,i) = 0$ for each i$\in$ V.

The **Reachability Graph** for a square network N=(V,E) is defined as follows: let G=(V,E) be the graph representation for N. The Reachability Graph for N is the graph G'=(V,E') where E'= {(i,j)$\in$ VxV | $\exists$directed path from i to j in G}.

The **Underlying Network** for a network N=(V,E) is defined as follows: N'=(V,E') where E'= {(i,j) | (i,j)$\in$ E $\vee$ (j,i)$\in$ E }. That is, an symmetric version of N.

Measure Name	Description	Reference	Formula
Access Index, Knowledge Based	Boolean value which is true if an agent is the only agent who knows a piece of knowledge and who is known by exactly one other agent. The one agent known also has its KAI set to one. **Type** Node Level **Input** AK:binary; A:binary **Output** Binary	Ashworth	The Knowledge Access Index (KAI) for agent i is defined as follows: let $S_i = \{s \mid AK(i,s) \wedge (sum(AK(:,s)) = 1) \wedge (sum(A(i,:)) = 1)\}$ Then $KAI_i = ((S_i \neq \emptyset) \vee (\exists j \mid S_j \neq \emptyset \wedge A(j,i) = 1))$
Access Index, Resource Based	Boolean value which is true if an agent is the only agent with access to a resource and who is known by exactly one other agent. The one agent known also has its RAI set to one. **Type** Node Level **Input** AR:binary; A:binary **Output** Binary	Ashworth	The Resource Access Index (RAI) for agent i is defined identically as Knowledge Access Index, with the matrix AK replaced by AR.
Actual Workload, Knowledge	The knowledge an agent uses to perform the tasks to which it is assigned. **Type** Node Level **Input** AK:binary; KT:binary; AT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Actual Workload for agent i is defined as follows: (AK*KT*AT')(i,i)/sum(KT) Note how Potential Workload is the first matrix product.
Actual Workload, Resource	The resources an agent uses to perform the tasks to which it is assigned. **Type** Node Level **Input** AR:binary; RT:binary; AT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Actual Resource Workload for agent i is identical to Actual Knowledge Workload, replacing AK with AR and KT with RT.
Cut Point Vertices	A node who if removed from a network N creates one or more new weak components is a Cut Point Vertice. **Type** Node Level **Input** N:square, symmetric **Output** Binary	Cormen, Leiserson, Riverest, Stein, 2001 p.558	A Cut Point Vertice is an *articulation point* of N, as defined in the referenced book.

Centrality, Betweenness	The Betweenness Centrality of node v in a network N is defined as: across all node pairs that have a shortest path containing v, the percentage that pass through v. This is defined for directed networks.
	Type Node Level
	Input N: square
	Output $\Re \in [0,1]$
		Freeman, 1979	Let G=(V,E) be the graph representation for the network. Let n=|V|, and fix a node v$\in$ V.
			For (u,w)$\in$ VxV, let $n_G(u,w)$ be the number of geodesics in G from u to w.
			If (u,w)$\in$ E, then set $n_G(u,w)$ =1.
			Define the following:
			let $S = \{(u,w) \in VxV \mid d_G(u,w) = d_G(u,v) + d_G(v,w)\}$
			let between = $\sum_{(u,w)\in S} (n_G(u,v) * n_G(v,w)) / n_G(u,w)$
			Then Betweenness Centrality of node v = between / ((n-1)(n-2)/2).
			
			Note: if G is not symmetric, then between is normalized by (n-1)(n-2).

Centrality, Closeness	The average closeness of a node to the other nodes in a network N. Loosely, Closeness is the inverse of the average distance in the network between the node and all other nodes. This is defined for directed networks. **Type** Node Level **Input** N:square **Output** $\Re \in [0,1]$	Freeman, 1979	Let G=(V,E) be the graph representation of the square network. Fix v$\in$ V. let dist = $\sum_{i\in V} d_G(v,i)$, if every node is reachable from v Then Closeness Centrality of node v = (|V|-1)/dist. If some node is not reachable from v then the Closeness Centrality of v is |V|.
Centrality, Degree	The Degree Centrality of a node in a square network N is its normalized out-degree. This is defined the same for directed networks. **Type** Node Level **Input** N:square **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994 (pg 199)	Let G=(V,E) be the graph representation of a square network and fix a node x. let deg = $card\{u \in V \mid (x,u) \in E\}$, this is the out-degree of node x. The Degree Centrality of node x is deg / (|V|-1)

Clustering Coefficient, 1998	Measures the degree of clustering in a network N. **Type** Graph Level **Input** N:symmetric(?), square **Output** $\Re \in [0,1]$	Watts and Strogatz, 1998	let G=(V,E) be the graph representation of a square network. For each node i∈ V define the following: let $in_i = \{u \in V \mid (u,i) \in V\}$ let $out_i = \{u \in V \mid (i,u) \in V\}$ let $inconnect_i = \{(u,v) \in E \mid u,v \in in_i\}$ let $outconnect_i = \{(u,v) \in E \mid u,v \in out_i\}$ Then compute for each node i∈ V its Node Clustering Coefficient ncc_i. There are three ways to do this: based on (1) in-degree, (2) out-degree, or (3) freeman degree: If $\lvert in_i \rvert = 0$ or $\lvert out_i \rvert = 0$, then $ncc_i = 0$. Otherwise, compute ncc_i in one of the following three ways: (1) let $ncc_i = \frac{\lvert inconnect_i \rvert}{\lvert in_i \rvert^2 - \lvert in_i \rvert}$ (2) let $ncc_i = \frac{\lvert outconnect_i \rvert}{\lvert out_i \rvert^2 - \lvert out_i \rvert}$ (3) let $ncc_i = \frac{1}{2}\left(\frac{\lvert inconnect_i \rvert}{\lvert in_i \rvert^2 - \lvert in_i \rvert} + \frac{\lvert outconnect_i \rvert}{\lvert out_i \rvert^2 - \lvert out_i \rvert}\right)$ Then Clustering Coefficient = $\left(\sum_{i \in V} ncc_i\right) / \lvert V \rvert$.

Cognitive Load	A complex measure taking into account the number of other agents, resources, and tasks an agent needs to manage and the communication needed to engage in such activity. Note: Cognitive Load is defined if one or both of the following pairs of networks exists: {AR,RT}, {AK,KT}. **Type** Node Level **Input** A:binary; AT:binary; [AR:binary; RT:binary]; [AK:binary; KT:binary] **Output** $\Re \in [0,1]$	Carley, 2002	The Cognitive Load for agent i is defined as follows: let ATR = AT*RT' let ATA = AT*AT' let x_1 = # of agents that agent i interacts with / total # of agents $= \left(\sum_{j \neq i} A(i,j)\right) / (\lvert A \rvert - 1)$ let x_2 = # of tasks agent i is assigned to / total # of tasks = sum(AT(i,:))/|T| let x_3 = sum of # agents who do the same tasks as agent i / (total # tasks * total # agents) $= \left(\sum_{j \neq i} ATA(i,j)\right) / (\lvert A \rvert - 1)(\lvert T \rvert)$ Note that x_4, x_5, x_6 depend upon networks AR and RT; if the networks AK and KT exist, then three analogous terms for knowledge are computed and averaged. If only AK and KT exist, then only they are used. let x_4 = # of resources agent i manages / total # of resources = sum(AR(i,:))/|R| let x_5 = sum of # resources agent i needs to do all its tasks / (total # tasks * total # resources) = sum(ATR(i,:))/(|T|*|R|) let x_6 = sum of negotiation needs agent i must do for each task / total possible negotiations $= \left(\sum_{j} (AR(i,j) > 0 \neq ATR(i,j) > 0)\right) / (\lvert R \rvert \lvert T \rvert)$ Then Cognitive Load for agent i = $(x_1 + x_2 + x_3 + x_4 + x_5 + x_6)/6$
Communicative Need	**Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Carley, 2002	Let G = (V,E) represent a square network: Then the Communicative Need = (Reciprocal Edge Count of G) / |E|
Component Count	The number of weakly connected components in a network N. **Type** Graph Level **Input** N:square, symmetric **Output** $Z \in [0, \lvert V \rvert]$	Wasserman and Faust, 1994 (pg 109)	Given a square, symmetric network represented by a graph G=(V,E), the Component Count is the number of connected components in G. Such components are often called "weak" because the graph G is undirected.

<table>
<tr>
<td>Congruence, Communication</td>
<td>Measures to what extent agents communicate when and only when it is needful to complete tasks. Hence, higher congruence occurs when agents don't communicate if the tasks don't require it, and do when tasks require it. Communication needs to be reciprocal.
Type Graph Level
Input AT:binary; AR:binary; RT:binary, T:binary
Output $\Re \in [0,1]$</td>
<td>Carley, 2002</td>
<td>Communication Congruence = 1 iff agents communicate when and only when it is needful to complete their tasks. Agents i and j must reciprocally communicate iff one of the following is true:
(a) if i is assigned to a task s and j is assigned to a task t and s directly precedes task t (handoff)
(b) if i is assigned to a task s and j is also assigned to s (co-assignment)
(c) if i is assigned to a task s and j is not, and there is a resource r to which agents assigned to s have no access but j does (negotiation to get needed resource).
The three cases are computed as follows:
(a) let H = AT*T*AT'
(b) let C = AT*AT'
(c) let N = AT*Z*AR', where Z(t,r) = (AT'*AR - RT')(t,r)<0

Then let Q(i,j) = [(H+C+N) + (H+C+N)'](i,j) > 0, and note that reciprocal communication is required - indicated by adding the transpose.

let d = card{ (i,j) | A(i,j) != Q(i,j) }, which measures the degree to which communication differs from that which is needed to do tasks.

Finally, d /= (|A|*(|A|-1)), normalizes d to be in [0,1]
Then, Communication Congruence = 1 - d</td>
</tr>
<tr>
<td>Congruence, Knowledge</td>
<td>Measures the similarity between what knowledge is assigned to tasks via agents, and what knowledge is required to do tasks. Perfect congruence occurs when agents have knowledge when and only when it is needful to complete tasks.
Type Graph Level
Input AK:binary; AT:binary; KT:binary
Output $\Re \in [0,1]$</td>
<td>Carley, 2002</td>
<td>Knowledge Congruence = 1 iff agents have knowledge when and only when it is needful to complete their tasks. Thus, we compute the knowledge assigned to tasks via agents, and compare it with the knowledge needed for tasks.
let KAT = (AK'*AT)
let d = card{ (i,j) | (KAT(i,j)>0) != (KT(i,j)>0)}
let d = d / (|K|*|T|), which normalizes d to be in [0,1]
Then Knowledge Congruence = 1 - d</td>
</tr>
<tr>
<td>Congruence, Resource</td>
<td>Measures the similarity between what resources are assigned to tasks via agents, and what resources are required to do tasks. Perfect congruence occurs when agents have access to resources when and only when it is needful to complete tasks.
Type Graph Level
Input AR:binary; AT:binary; RT:binary
Output $\Re \in [0,1]$</td>
<td>Carley, 2002</td>
<td>Identical to Knowledge Congruence with AR replaced by AK and KT replaced by RT.</td>
</tr>
</table>

Connectedness	Given a square network N, the degree to which N's underlying network is connected. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Krackhardt, 1994	Let N be a given square network. The Connectedness of N is the Density of the Reachability Network for N.
Constraint	The degree to which an agent is constrained by its current communication network. **Type** Node Level **Input** A **Output** $\Re \in [0,1]$	Burt, 1992	This is the Effective Size of Network measure described by Equ. 2.4 on pg. 55 of Burt, 1992. Note that the Communication Network is used for the matrix Z.
Density	The actual number of network edges versus the maximum possible edges for a network N. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994 (pg 101)	Let M be the adjacency matrix for the network of dimension m x n. If the network is square, then M is square and has a zero diagonal, and therefore Density = sum(M)/(m*(m-1)). For rectangular networks, Density = sum(M)/(m*n).
Diameter	The maximum shortest path length between any two nodes in a square network G=(V,E). If there exist i,j in V such that j is not reachable from i, then the diameter is returned as |V|. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994 (pg 111)	The diameter of G=(V,E) is defined as: $\max\{d_G(i,j) \mid i,j \in V\}$ That is, the maximum shortest directed path between any two vertices in G. If there exists i and j such that j is not reachable from i, then |V| is returned.
Diversity	The distribution of difference in idea sharing. **Type** Graph Level **Input** AK:binary **Output**	???	Let w_k = sum(AK(:,k)), $1 \le k \le \lvert K \rvert$ Let d = $1 - \sum_{k=1}^{\lvert K \rvert} \left(w_k / \lvert A \rvert \right)^2$ Then Diversity = d / |A|
Edge Count, Lateral	Fixing a root node x, a lateral edge (i,j) is one in which the distance from x to i is the same as the distance from x to j. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Carley, 2002	Let G=(V,E) be the graph representation of a network. And fix a node x∈ V to be the root node. Let S = {(i,j)∈ E | $d_G(x,i) = d_G(x,j)$ } Then, Lateral Edge Count = |S| / |E|
Edge Count, Pooled	A pooled edge in a network N=(V,E) is an edge (i,j)∈ E such that there exists at least one other edge (i,k) ∈ E, and k ≠ j. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	Carley, 2002	Let M be the adjacency matrix representation of the network. Let S = { (i,j) | M(i,j)=1 ∧ sum(M(:,j))>1 } In other words: edge (i,j) is a pooled edge iff the indegree of node j > 1. The Pooled Edge Count = |S| / |E|

Edge Count, Reciprocal	The number of edges in a network N=(V,E) that are reciprocated; an edge (i,j)∈ E is reciprocated if (j,i)∈ E. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$		Let G=(V,E) be the graph representation of a network. Let S = card{(i,j)∈ E | i<j, (j,i)∈ E } he Reciprocal Edge Count = |S| / |E|
Edge Count, Sequential	The number of edges in network N that are neither Reciprocal Edges nor Pooled Edges. Note that an edge can be both a Pooled and a Reciprocal edge. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	Carley, 2002	Let G=(V,E) be the graph representation of a network, and let X = set of Pooled edges of G, and let Y = set of Reciprocal edges of G. Then Sequential Edge Count = | E-X-Y| / |E|
Edge Count, Skip	The number of edges in a network that skip levels. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	Carley, 2002	A skip edge in a network represented by G=(V,E) is an edge (i,j) ∈ E such that j is reachable from i in the graph G'=(V,E\(i,j)), that is, the graph G with edge (i,j) removed. Skip Count is simply the number of such edges in G normalized to be in [0,1] by dividing by |E|.
Effective Network Size	The effective size of an agent's Communication Network based on redundancy of ties. **Type** Node Level **Input** A **Output** $\Re \in [0,1]$	Burt, 1992	This is the Effective Size of Network measure described by Equ. 2.2 on pg. 52 of Burt, 1992. Note that the Communication Network is used for the matrix Z.
Exclusivity, Knowledge Based	Detects agents who have singular knowledge. **Type** Node Level **Input** AK:binary **Output** $\Re \in [0,1]$	Ashworth	The Knowledge Exclusivity Index (KEI) for agent i is defined as follows: $\sum_{j=1}^{\lvert K \rvert} AK(i,j) * \exp(1 - sum(AK(:,j)))$
Exclusivity, Resource Based	Detects agents who have singular resource access. **Type** Node Level **Input** AR:binary **Output** $\Re \in [0,1]$	Ashworth	The Resource Exclusivity Index (REI) for agent i is defined exactly as for Knowledge Based Exclusivity, but with the matrix AK replaced by AR.
Exclusivity, Task Based	Detects agents who exclusively perform tasks. **Type** Node Level **Input** AT:binary **Output** $\Re \in [0,1]$	Ashworth	The Task Exclusivity Index (TEI) for agent i is defined exactly as for Knowledge Based Exclusivity, but with the matrix AK replaced by AT.

Hierarchy	The degree to which a square network N exhibits a pure hierarchical structure. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Krackhardt, 1994	Let N be a given square network. The Hierarchy of N is the Reciprocity of the Reachability Network for N.
Interdependence	The percentage of edges in a network N that are Pooled or Reciprocal. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Carley, 2002	Let G=(V,E) be the graph representation of a square network. Let a = Pooled Edge Count and b = Reciprocal Edge Count of the network. Then Interdependence = (a+b)/|E|
Interlocker and Radial	Interlocker and radial nodes in a square network have a high and low Triad Count, respectively. **Type** Node Level **Input** N:square **Output** Binary	Carley, 2002	Let N=(V,E) be a square network. Let t_i = Triad Count for node i, $1 \le i \le \|V\|$. Let u = the mean of { t_i } Let d = the variance of { t_i } Then if $t_k \ge (u+d)$, then agent k is an *interlocker*. If $t_k \le (u-d)$ then agent k is a *radial*.
Load, Knowledge	Average number of knowledge per agent. **Type** Graph Level **Input** AK:binary **Output** $\Re \in [0,\|R\|]$	Carley, 2002	Knowledge Load = sum(AK)/ (|A|)
Load, Resource	Average number of resources per agent. **Type** Graph Level **Input** AR:binary **Output** $\Re \in [0,\|R\|]$	Carley, 2002	Resource Load = sum(AR)/ (|A|)
Negotiation, Knowledge	The extent to which personnel need to negotiate with each other because they lack the knowledge to do the tasks to which they are assigned. **Type** Graph Level **Input** AT:binary; AK:binary; KT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Compute the percentage of tasks that lack at least one resource: let Need = (AT'*AK) - KT' let S = { i | $1 \le i \le \|T\|$, $\exists$ j : Need(i,j) < 0 } Then Need for Negotiation = |S| / |T|
Negotiation, Resource	The extent to which personnel need to negotiate with each other because they lack the resources to do the tasks to which they are assigned. **Type** Graph Level **Input** AT:binary; AR:binary; RT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Identical to Knowledge Negotiation, replacing AK with AR, and KT with RT.

Network Centralization, Betweenness	Network centralization based on the betweenness score for each node in a square network. This measure is define for symmetric and non-symmetric networks. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Freeman, 1979	Let G=(V,E) represent the square network, and let n = |V| let d_i = Betweenness Centrality of node i let $\bar{d} = \max\{d_i \mid 1 \le i \le n\}$ Then Network Betweenness Cent. = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / (n-1)$.
Network Centralization, Closeness	Network centralization based on the closeness centrality of each node in a square network. This is not defined for unconnected or directed networks. **Type** Graph Level **Input** N:square, symmetric, connected **Output** $\Re \in [0,1]$	Freeman, 1979	Let G=(V,E) represent the square network, and let n = |V| let d_i = Closeness Centrality of node i let $\bar{d} = \max\{d_i \mid 1 \le i \le n\}$ Then Network Closeness Cent. = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / ((n-2)(n-1)/(2n-3))$.
Network Centralization, Column Degree	A centralization based on the out degree of column vertices in a network N. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	NetStat	Let M be the adjacency matrix representation of a rectangular network with n rows and o columns. let $d_j = sum(M(:, j))$ = out degree of column node j, $1 \le j \le o$ let $\bar{d} = \max\{d_j \mid 1 \le j \le o\}$ Then Column Degree Network Centralization = $\left(\sum_{1 \le j \le o} \bar{d} - d_j\right) / ((o-1) * n)$.
Network Centralization, Degree	This centralization is defined on a square network N and is based on node out-degree. The scaling of the measure depends on whether the network is symmetric. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Freeman, 1979	Let M be the adjacency matrix representation of a square network. And let n=|M|. let $d_i = sum(M(i,:))$ = out degree of node i let $\bar{d} = \max\{d_i \mid 1 \le i \le n\}$ Then Degree Network Centralization = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / ((n-1)(n-2))$. Note: if the network is not symmetric, then the scaling factor is $(n-1)^2$
Network Centralization, Row Degree	A centralization based on the out degree of row vertices in a network N. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	NetStat	Let M be the adjacency matrix representation of a rectangular network with n rows and o columns. let $d_i = sum(M(i,:))$ = out degree of row node i let $\bar{d} = \max\{d_i \mid 1 \le i \le n\}$ Then Row Degree Network Centralization = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / ((n-1) * o)$. Note: dividing by (n-1)*o normalizes the value to be in [0,1]

Network Levels	The Network Level of a square network N is the maximum Node Level of its nodes. **Type** Graph Level **Input** N:square **Output** $Z \in [0, \|V\| - 1]$	NetStat	Let G=(V,E) be the graph representation of a square network. Then the Levels of G = max { $d_G(i, j)$ | i,j∈ V; j reachable from i in G }
Node Level	The Node Level for a node v in a square network N is the worst case shortest path from v to every node v can reach. **Type** Node Level **Input** N:square **Output** $Z \in [0, \|V\| - 1]$	Carley, 2002	Let G=(V,E) be the graph representation of a square network and fix a node v. Node Level for v = max { $d_G(v, j)$ | j∈ V; j reachable from v in G }
Omega, Knowledge Based	The degree to which an organization reuses knowledge. **Type** Graph Level **Input** AT:binary; KT:binary; T:binary **Output** $\Re \in [0,1]$	Carley, Dekker, and Krackhardt 2000	Let TAT = TA*TA' Let N = ((T'@TAT)*KT')@KT' Then Knowledge Based Omega = sum(N)/sum(KT)
Omega, Resource Based	The degree to which an organization reuses resources. **Type** Graph Level **Input** AT:binary; RT:binary; T:binary **Output** $\Re \in [0,1]$	Carley, Dekker, and Krackhardt 2000	Identical to Knowledge Based Omega, replacing KT with RT.

Performance as Accuracy	Measures how accurately agents can perform their assigned tasks based on their access to knowledge and resources. **Type** Graph Level **Input** AK:binary; AT:binary; AR:binary; KT:binary; RT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Accuracy is computed based on the binary classification problem. It is computed in one of two ways: (1) Knowledge based: Let b be a binary string of length |K|, let N=KT', and let S=AK. Fix a task t. let answer = ($\sum_{1\le k\le \|K\|} N(t,k)b_k \,/ \sum_{1\le k\le \|K\|} N(t,k)$ > .5) , which is the correct classification of b with respect to task t. Now, let let I={ i | AT(i,t)=1}. let answer(i) = ($\sum_{1\le k\le \|K\|} N(t,k)S(i,k)b_k / \sum_{1\le k\le \|K\|} N(t,k)S(i,k)$ > .5), i∈ I. This is agent i's classification of b with respect to t. The group of agents classify b using majority voting. That is, let group_answer = ($\frac{1}{\|I\|}\sum_{i\in I} answer(i)$ > .5). Then, if group_answer = answer, then the group was accurate, otherwise not. This is repeated multiple times for each task, and across all tasks. The percentage correct is Performance as Accuracy. (2) Resource based: let N=RT' and S=AR in the analysis of case (1). If the network has the knowledge and resource graphs to perform both cases, then Performance as Accuracy is the average of the two.
Potential Workload, Knowledge	Maximum knowledge an agent could use to do tasks if it were assigned to all tasks. **Type** Node Level **Input** AK:binary; KT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Potential Knowledge Workload for agent i = sum((AK*KT)(i,:))/sum(KT)
Potential Workload, Resource	Maximum resources an agent could use to do tasks if it were assigned to all tasks. **Type** Node Level **Input** AR:binary; RT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Potential Resource Workload for agent i is identical to Potential Knowledge Workload, replacing AK with AR, and KT with RT.
Reciprocity	The fraction of joined node pairs that are reciprocally joined in a square network N. **Type** Graph Level **Input** N: square **Output** $\Re \in [0,1]$	NetStat	Let G=(V,E) represent a square network. let S = {(i,j) | (i,j)∈ E ∧ (j,i)∈ E} let T = {(i,j) | (i,j)∈ E ∨ (j,i)∈ E} Then the network's Reciprocity = |S|/|T|
Redundancy, Access	Average number of redundant agents per resource. An agent is redundant if there is already an agent that has access to the resource. **Type** Graph Level **Input** AR:binary	Carley, 2002	This is the Column Redundancy of matrix AR.

	Output $\Re \in [0,(\|A\|-1)*\|R\|]$		
Redundancy, Assignment	Average number of redundant agents assigned to tasks. An agent is redundant if there is already an agent assigned to the task. **Type** Graph Level **Input** AT **Output** $\Re \in [0,(\|A\|-1)*T]$	Carley, 2002	This is the Column Redundancy of matrix AT.
Redundancy, Column	Given a network N, the mean number of non-zero column entries in excess of one in the network's matrix representation. **Type** Graph Level **Input** N of dimension m x n **Output** $\Re \in [0,(m-1)*n]$	Netstat	Let M be the matrix representation for a network N of dimension m x n. let $d_j = \max\{0, sum(M(:, j)) - 1\}$, for $1 \le j \le n$; this is the number of column entries in excess of one for column j. Then Column Redundancy = $\left(\sum_{j=1}^{n} d_j\right)/n$
Redundancy, Knowledge	Average number of redundant agents per knowledge. An agent is redundant if there is already an agent that has the knowledge. **Type** Graph Level **Input** AK **Output** $\Re \in [0,(\|A\|-1)*\|K\|]$	Carley, 2002	This is the Column Redundancy of matrix AK.
Redundancy, Resource	Average number of redundant resources assigned to tasks. A resource is redundant if there is already a resource assigned to the task. **Type** Graph Level **Input** RT:binary **Output** $\Re \in [0,(\|R\|-1)*\|T\|]$	Carley, 2002	This is the Column Redundancy of matrix RT.
Redundancy, Row	Given a network N, the mean number of non-zero row entries in excess of one in the network's matrix representation. **Type** Graph Level **Input** N of dimension m x n **Output** $\Re \in [0,(n-1)*m]$	Netstat	Let M be the matrix representation for a network N of dimension m x n. let $d_i = \max\{0, sum(M(j,:)) - 1\}$, for $1 \le i \le m$; this is the number of column entries in excess of one for row i. Then Row Redundancy = $\left(\sum_{j=1}^{m} d_j\right)/m$

Relative Expertise	The degree of dissimilarity between agents based on shared knowledge. Each agent computes to what degree the other agents know what they do not know. **Type** Node Level **Input** AK:binary **Output** $\Re \in [0,1]$	Carley, 2002	The Relative Expertise matrix (RE) is defined as follows: RE(i,i) = 0 RE(i,j) = (~AK*AK') = # knowledge that j knows that i does not know Finally, normalize RE by its row sums: RE(i,:) /= sum(RE(i,:)) The Relative Expertise for agent i = $\left(\sum_{\substack{j=1\\ j\neq i}}^{\lvert A\rvert} RE(i,j)\right)/(\lvert A\rvert-1)$, that is, the average of the non-diagonal elements of row i of RE.
Relative Similarity	The degree of similarity between two agents based on shared knowledge. Each agent computes to what degree the other agents know what they know. **Type** Node Level **Input** AK: binary **Output** $\Re \in [0,1]$	Carley, 2002	Let M = AK*AK' Let w(i) = sum(M(i,:)), $1 \le i \le \lvert A\rvert$ Then Relative Similarity (RS) between agents i and j is RS(i,j) = M(i,j)/w(i). The Relative Similarity for an agent i = $\left(\sum_{\substack{j=1\\ j\neq i}}^{\lvert A\rvert} RS(i,j)\right)/(\lvert A\rvert-1)$, that is, the average of the non-diagonal elements of row i of RS.
Span of Control	The average number of subordinates per supervisor in the Communication Network. **Type** Graph Level **Input** A:binary **Output** $\Re \in [0,\lvert V\rvert-1]$	Carley, 2002	For each agent in the Communication Network who has 1 or more subordinates (a supervisor), sum the number of subordinates, then divide by the number of supervisors.
Speed, Average	The average communication time between any two agents who can communicate via some path. **Type** Graph Level **Input** A **Output** $\Re \in [0,1]$	Carley, 2002	let G=(V,E) be the graph representation of the Communication Network. let D={ $d_G(i,j)$ | i,j∈ V, i $\neq$ j; j reachable from i in G } Then Average Speed = $\left(\sum_{d\in D} d\right)/\lvert D\rvert$
Speed, Minimum	The worst case communication time between any two agents. **Type** Graph Level **Input** A **Output** $\Re \in [0,1]$	Carley, 2002	Minimum Speed = 1 / (Levels for the Communication Network)
Task Completion, Knowledge Based	The percentage of tasks that can be completed by the agents assigned to them, based solely on whether the agents have the requisite knowledge to do the tasks. **Type** Graph Level **Input** AK:binary; AT:binary; KT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Find the tasks that cannot be completed because the agents assigned to the tasks lack necessary knowledge: let Need = (AT'*AK) - KT' let S = { i | $1 \le i \le \lvert T\rvert$, $\exists$ j : Need(i,j) < 0 } Knowledge Based Task Completion is the percentage of tasks that could be completed = (|T|-|S|) / |T|

Task Completion, Overall	The percentage of tasks that can be completed by the agents assigned to them, based solely on whether the agents have the requisite knowledge and resources to do the tasks. **Type** Graph Level **Input** AR:binary; AT:binary; RT:binary; AK:binary, KT:binary **Output** $\Re \in [0,1]$	Carley, 2002	This is the average of Knowledge Based Task Completion and Resource Based Task Completion. If one of the two could not be computed, then the other is returned.
Task Completion, Resource Based	The percentage of tasks that can be completed by the agents assigned to them, based solely on whether the agents have the requisite resources to do the tasks. **Type** Graph Level **Input** AR:binary; AT:binary; RT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Find the tasks that cannot be completed because the agents assigned to the tasks lack necessary resources. Defined identically as Knowledge Based Task Completion, replacing matrix AK with AR and matrix KT with RT.
Transitivity	The percentage of triads i,j,k in a square network N such that if (i,j) and (j,k) are in the network, then (i,k) is in the network. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	NetStat	Let M be the adjacency matrix representation of the network. let I = {(i,j,k) ∈ V^3 | i,j,k distinct } let Potential = { (i,j,k) ∈ I | M(i,j) = M(j,k) } let Empty = { (i,j,k) ∈ I | M(i,j)=M(j,k)=M(i,k)=0 } let Complete = { (i,j,k) ∈ I | M(i,j)=M(j,k)=M(i,k)=1 } Then Transitivity = (|Empty| + |Complete|)/|Potential|
Triad Count	The number of Communication Network triads that an agent is in. **Type** Node Level **Input** A:binary **Output** $Z \in [0, (\|A\|-1)(\|A\|-2)]$	NetStat	let Triad be an agent by agent matrix where Triad(i,i) = 0 Triad(i,j) = card{ k | k != i, k != j; A(i,j) ∧ A(i,k) ∧ A(k,j) }, i ≠ j Then the Triad count for agent i = sum(Triad(i,:))

Trust	The trust value for an agent is the average trust that exists between it and the other agents.
	Type Node Level
	Input AR:binary; AK:binary; AT:binary, A:binary
	Output $\Re \in [0,1]$
		Carley, 2002	let Trust be a matrix of dimension |A| x |A| defined as follows:
			Trust(i,i) = 0
			Trust(i,j) = (# triads with both i and j) +
			AR(i,:)' * AR(j,:) + // # resources i and j share
			AK(i,:)' * AK(j,:) + // # knowledge i and j share
			AT(i,:)' * AT(j,:) + // # tasks i and j share
			A(i,j) ∧ A(j,i) + // reciprocal communication tie between i and j
			|A| / $d_P(i,j)$ // inverse communication time between i and j
			Trust is then normalized so that each entry is in [0,1].
			The trust value for agent i = sum(Trust(i,:)) / |A|

Under Supply, Knowledge	The extent to which the knowledge needed to do tasks are unavailable in the entire organization. **Type** Graph Level **Input** AK:binary; AT:binary; KT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Compute the average number of needed knowledge per task: let Need = (AT'*AK) - KT' let TaskNeed(i) = card{ j | Need(i,j)<0 }, for 1<=i<=|T| Then UnderSupply is sum(TaskNeed)/ |T|

Under Supply, Resource	The extent to which the resources needed to do tasks are unavailable in the entire organization. **Type** Graph Level **Input** AR:binary; AT:binary; RT:binary **Output** $\Re \in [0,1]$	Carley, 2002	Under Resource Supply is identical to Under Knowledge Supply, replacing AK with AR, and KT with RT.
Upper Boundedness	The degree to which pairs of agents have a common ancestor. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Krackhardt, 1994	

$\in$ $\neq$ $\exists$ $\subset$ $\leq$ $>$

Bibliography:

Bonacich, Phil 1987: "Power and centrality: A family of measures." American Journal of Sociology 92: 1170-1182.

Burt, Ronald. *Structural Holes: The Social Structures of Competition*. Cambridge, MA: Harvard University Press, 1992.

Carley, Kathleen 2002. *Summary of Key Network Measures for Characterizing Organizational Architectures*. Unpublished Document: CMU 2002

Cormen, Leiserson, Rivest, Stein 2001. *Introduction to Algorithms, Second Edition*. Cambridge, MA: MIT Press, 2001.

Carley, K, Dekker, D., Krackhardt, D (2000). "How Do Social Networks Affect Organizational Knowledge Utilitization?"

Fienberg, S.E., Meyer, M.M., and Wasserman, S.S. (1985). ``Statistical Analysis of Multiple Sociometric Relations," Journal of the American

Freeman, L.C. (1979). Centrality in Social Networks I: Conceptual Clarification. Social Networks, 1, 215-239.

Krackhardt, D. 1994. *Graph Theoretical Dimensions of Informal Organizations*. In Computational Organization Theory, edited by Carley, Kathleen M. and M.J. Prietula. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.

Newman MEJ, Moore C, Watts DJ Mean-field solution of the small-world network model PHYS REV LETT 84 (14): 3201-3204 APR 3 2000

Newman MEJ, Watts DJ Renormalization group analysis of the small-world network model PHYS LETT A 263 (4-6): 341-346 DEC 6 1999

Newman MEJ, Watts DJ Scaling and percolation in the small-world network model PHYS REV E 60 (6): 7332-7342 Part B DEC 1999 Statistical Association}, 80, 51-67.

Wasserman, Stanley and Katherine Faust. Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press, 1994.

Watts DJ Networks, dynamics, and the small-world phenomenon AM J SOCIOL 105 (2): 493-527 SEP 1999

Watts DJ, Strogatz SH Collective dynamics of 'small-world' networks NATURE 393 (6684): 440-442 JUN 4 1998