ORA: Organization Risk Analyzer*

CASOS Technical Report

Kathleen M. Carley, Jeff Reminga

July 2004
CMU-ISRI-04-106

Carnegie Mellon University
School of Computer Science
ISRI - Institute for Software Research International
CASOS - Center for Computational Analysis of Social and Organizational Systems

Abstract

ORA is a network analysis tool that detects risks or vulnerabilities of an organization's design structure. The design structure of an organization is the relationship among its personnel, knowledge, resource, and task entities. These entities and relationships are represented by the Meta-Matrix. Measures that take as input a Meta-Matrix are used to analyze the structural properties of an organization for potential risk. ORA contains over 50 measures which are categorized by which type of risk they detect. Measures are also organized by input requirements and by output. ORA generates formatted reports viewable on screen or in log files, and reads and writes networks in multiple data formats to be interoperable with existing network analysis packages. In addition, it has tools for graphically visualizing Meta-Matrix data and for optimizing a network's design structure. ORA uses a Java interface for ease of use, and a C++ computational backend. The current version ORA 1.2 software is available on the CASOS website http://www.casos.cs.cmu.edu/projects/ora/software.html.

* This work was supported in part by the National Science Foundation under grants: No. ITR/IM IIS-0081219, NSF IGERT 9972762 in CASOS, NSF IRI9633 662, the Office of Naval Research (ONR), Office of Naval Research Grant No. N00014-97-1-0037 and N0014-02-10973, and NASA.. Additional support was provided by the center for Computational Analysis of Social and Organizational Systems (CASOS) and the Institute for Software Research International at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the office of naval research, NASA, or the U.S. government.

Keywords: dynamic network analysis, measures, meta-matrix, organization risk

Table of Contents

I. Index of Tables

II. Index of Figures

1. ORA Motivation and Description

ORA is a network analysis tool that detects risks or vulnerabilities of an organization's design structure. The design structure of an organization is the relationship among its personnel, knowledge, resources, and tasks entities. These entities and relationships are represented by a collection of networks called the Meta-Matrix. ORA analyzes the Meta-Matrix using measures, and reads and writes network data in multiple formats to make it interoperable with existing network analysis software.

The modeling of organizations as networks and the development of measures to examine their design structure is well developed. Even a cursory analysis of the literature reveals a wide variety of measures for assessing organizational risk and vulnerability [1] [2] [4] [6] [9] [10]. Such measures vary dramatically in the detail and type of data needed to determine that measure. They span from the assessment of critical employees, to the tendency to group think, to the potential for adaptability. In fact, it is possible to provide a suite of measures and metrics that capture both the organizational design and the possible changes in that design that are likely to result in group think, error cascades, and IP loss [7].

Given the high potential number of vulnerabilities and risks, what is needed is a framework for evaluating this set of metrics, assessing the value of existing metrics, locating gaps in the existing metrics, developing new metrics as needed, and so providing a more comprehensive guide to which metrics to use when. ORA has been designed to provide this framework.

A large number of metrics for assessing organizational vulnerability and design have been assessed and over 50 of them are now incorporated in ORA. As metrics are incorporated, if they cannot handle binary data, then we are developing a non-binary form. Help is provided for each measure that describes the measure definition and formula, input data constraints, and computational complexity. Fastest known algorithms are incorporated, employing sparse and non-sparse matrix techniques. Further, the number of nodes of any one type – personnel, knowledge, resources, tasks etc. in ORA is limited only by machine memory and processor speed. All measures are based on the Meta-Matrix and take into 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. Where possible, metrics are normalized to be within 0 and 1 to provide a consistent framework.

ORA can be used to do a risk audit for the organization of its individual and organization risks. Such risks include, but are not limited to, tendency to groupthink, overlook of information, communication barriers, and critical employees. It evaluates potential organizational risks based upon underlying social, knowledge, resource, and task networks. This tool takes the Meta-Matrix data at a particular point in time and calculates a series of metrics assessing the team's design, particularly the command and control structure, and the associated organizational risks. ORA has been used to assess risk in various organizational and government settings including NASA, nursing hospitals, and joint task force settings.

2. Input

2.1 The Organization as Meta-Matrix

The main unit of input in ORA is the organization. An organization can be modeled and characterized as a set of interlocked networks connecting entities such as people, knowledge resources, tasks and groups. These interlocked networks can be represented using the Meta-Matrix conceptual framework (see [5] [7] [8]) presented in Table 1.

Table 1: Meta-Matrix Showing Networks of Relations Connecting Node Entities

	People	Knowledge	Resources	Tasks/Projects
People	Social Network *Who talks to, works with, and reports to whom*	Knowledge Network *Who knows what, has what expertise or skills*	Resource Network *Who has access to or can use which resource*	Assignment Network *Who is assigned to which task or project, who does what*
Knowledge		Information Network *Connections among types of knowledge, mental models*	Resource Usage Requirements *What type of knowledge is needed to use that resource*	Knowledge Requirements *What type of knowledge is needed for that task or project*
Resources			Inter-operability and Co-usage Requirements *Connections among resources, substitutions*	Resource Requirements *What type of resources are needed for that task or project*
Tasks/ Projects				Precedence and Dependencies *Which tasks are related to which*

This Meta-Matrix serves as an integrating feature of a managerial toolkit. The Meta-Matrix serves several purposes; 1) it provides a way of conceptualizing the set of entities and relations among them that the research and associated tools will focus on; 2) it brings to the forefront the recognition that the data that is collecting will be not just the attributes of the entities (people, knowledge, resources, tasks and/or projects, and groups or teams) but also the set of relations or ties among them; 3) it provides an identification of the class of entities and relations that will be used in doing organizational design, analysis and risk evaluation; and 4) it provides a common ontology for talking about and representing organizational information.

2.1 Meta-Matrix Data Formats

To make ORA interoperable with existing network analysis software, ORA reads and writes Meta-Matrix network data in multiple formats. The networks that constitute a Meta-Matrix can be stored in separate files, with one network per file, or they can be collected into a single file.

ORA supports the DL, Extended-DL (EDL), CSV, and RAW formats for reading and writing a file containing a single type of network (for example, the type Agent x Agent). The RAW and DL formats are defined by the network analysis package UCINET [3]. DyNetML and EDL are the supported formats for representing in a single file the multiple network types of the Meta-Matrix.

2.1.1 DyNetML

DyNetML is an XML specification that represents the node entities Agent, Knowledge, Resource, and Task and the networks defined on them. DyNetML supports multiple Meta-Matrices to be within the same file, and each Meta-Matrix can have different Agent, Knowledge, Resource, and Task node sets. Because DyNetML is XML it is humanly readable. DyNetML is described more fully in the DyNetML Technical Report.

2.1.1 Extended-DL

Extended-DL (EDL) is the DL format of UCINET with two extensions. The first extension adds more header information to the DL file using two additional tokens. The two tokens identify the type of row nodes and the type of column nodes. The two tokens are ROW TYPE and COLUMN TYPE and are followed by one of the following: AGENT, KNOWLEDGE, RESOURCE, or TASK. COLUMN TYPE can be shortened to COL TYPE. Note that tokens in DL and EDL are case insensitive. The tokens allow the user to specify the network type of the data. For example, the following EDL file specifies a Knowledge Network (Agent x Knowledge):

```
ROW TYPE = AGENT
COLUMN TYPE = KNOWLEDGE
DL
NR=3, NC=6
FORMAT = FULLMATRIX
DATA:
0 1 1 0 0 0
1 0 0 0 1 0
0 0 0 1 1 1
```

Note that the ROW TYPE and COLUMN TYPE tokens appear before the DL token, and because the tokens are optional, a valid DL file is a valid EDL file. In short, an EDL file can be created by adding the two new tokens to the beginning of an existing DL file.

The second extension in EDL is the ability to have multiple network types in a single file. Each network type section must be a complete and valid EDL file. The sections are separated by two vertical bars ('||'). Thus, the extension simply allows files to be concatenated into a single file. For example, the following file contains two types of networks from a Meta-Matrix, a Knowledge Network (Agent x Knowledge) and a Communication Network (Agent x Agent):

```
ROW TYPE = AGENT
COL TYPE = KNOWLEDGE
DL
NR=3, NC=6
FORMAT = FULLMATRIX
MATRIX LABELS:
“Knowledge Network”
DATA:
0 1 1 0 0 0
1 0 0 0 1 0
0 0 0 1 1 1
||
ROW TYPE = AGENT
COL TYPE = AGENT
DL
N=3
FORMAT = FULLMATRIX
MATRIX LABELS:
“Communication Network”
DATA:
0 1 1
1 0 0
0 0 0
```

To summarize, when ORA creates a Meta-Matrix as output the user can choose to save each type of matrix in the Meta-Matrix in a separate file, or to save the entire Meta-Matrix in a single file. When saving individual network types to a file, the following formats are available: EDL, DL, RAW, and CSV. When saving the entire Meta-Matrix to a single file, the following are available: EDL and DyNetML. These output formats make ORA interoperable with other network analysis packages.

3. Meta-Matrix Measures

An ORA measure is a function that takes a Meta-Matrix as input. Each measure examines a particular aspect of the mathematical structure of the Meta-Matrix. The metrics in ORA include social network, task management, and dynamic network metrics. ORA contains over 50 measures, and provides three classifications of them based on risk and vulnerability, input requirements, and type of output produced. The three classifications enable the user to quickly find a measure based on its properties.

3.1 Measure Risk Categories

The first classification divides the measures into seven categories of risk and vulnerability: Communication Risk, Critical Employee Risk, Resource Allocation Risk, Redundancy Risk, Personnel Interaction Risk, Task Risk, and Performance Risk. The measures in each category analyze the Meta-Matrix structure to detect the type of risk. A single measure can be classified into more than one risk category. Each of the seven categories is briefly discussed below.

3.1.1 Critical Employee Risk

Critical Employee Risk is the risk based on employees having exclusive knowledge, resources, or task assignments. Measures in this category assess in part: would the removal of one employee from the organization greatly affect the ability to complete tasks? Do employees tend to have exclusive access to knowledge or resources?

3.1.2 Resource Allocation Risk

Resource Allocation Risk is the risk based on how the organization's resource allocation affects its ability to complete tasks. Measures in this category assess: is agent workload evenly distributed? Do agents have access to the resources they need to complete tasks? Do agents have access to resources they do not use?

3.1.3 Communication Risk

Communication Risk is the based on the level of communication and the authority structure of the organization. This category seeks to answer the following questions: are agents able to communicate when necessary to complete tasks? Is communication too centralized or decentralized? Do agents have recourse to managers to settle disputes?

3.1.4 Redundancy Risk

Redundancy Risk is the risk based on redundancy in task assignments, resource access, and knowledge access. An organization with little redundancy is more adversely affected by an agent or resource no longer being available. On the other hand, too much redundancy makes an organization inefficient.

3.1.5 Task Risk

Task Risk is the risk based on task precedence and task assignment. Measures in this category are able to evaluate the following questions: do agents have the resources to complete their tasks? are tasks highly interdependent so that the inability to perform one task prevents many other tasks from being completed?

3.1.6 Personnel Interaction Risk

Personnel Interaction Risk is the risk based on agent communication, either agents communicating who should not be, or vice-versa. Measures in this category examine the organization design structure to assess the following: are agents with similar skills interacting? Are agents with complementary skills interacting? Are there groups of agents communicating in unexpected ways? Is there a group of agents that has extensive reach in the organization, or whose removal would greatly fragment the organization.

3.1.7 Performance Risk

Performance Risk is the risk based on ability to complete tasks accurately. Measures in this category assess the following questions: is the organization able to complete all tasks? How well does the organization build consensus? How many tasks would be left undone if a single employee were selected for removal?

3.2 Measure Input Requirements

A second way that measures can be classified is according to input requirements. Measures that take as input a single matrix (a cell of the Meta-Matrix) are called Single-Cell measures; measures taking more than one cell are called Multi-Cell measures. In addition, some measures require only one matrix as input, but it need not correspond to a specific cell in the Meta-Matrix, but only a square sub-section of the Meta-Matrix; these are called Square measures. For example, all of the Centrality measures are Square measures, and as such they can take as input the AxA (Communication) matrix from the Meta-Matrix, but they could also take the entire Meta-Matrix – which is square, or the TxT matrix.

3.3 Measure Output Types

A third classification of measures is by output. A measure produces one of two types of output: graph level or node level. A graph level measure's output is associated with one or more matrices (also called graphs) from the Meta-Matrix. Graph-Level measures are always scalar valued. For example, Density is a graph level measures because it outputs a scalar value that describes a property of the input graph as a whole. The output of a Node Level measure, on the other hand, is associated with the members of a node entity. For example, the Cognitive Load measure is a Node Level measure because it produces a scalar value for each Agent node.

To summarize the three classification schemes, Table 2 classifies some of the measures available in ORA. A complete listing of measures available in ORA with descriptions, formulas, input requirements, and output data is in Appendix A: ORA Measures. Measures are listed by risk category in Appendix B: ORA Risk Measure Categories.

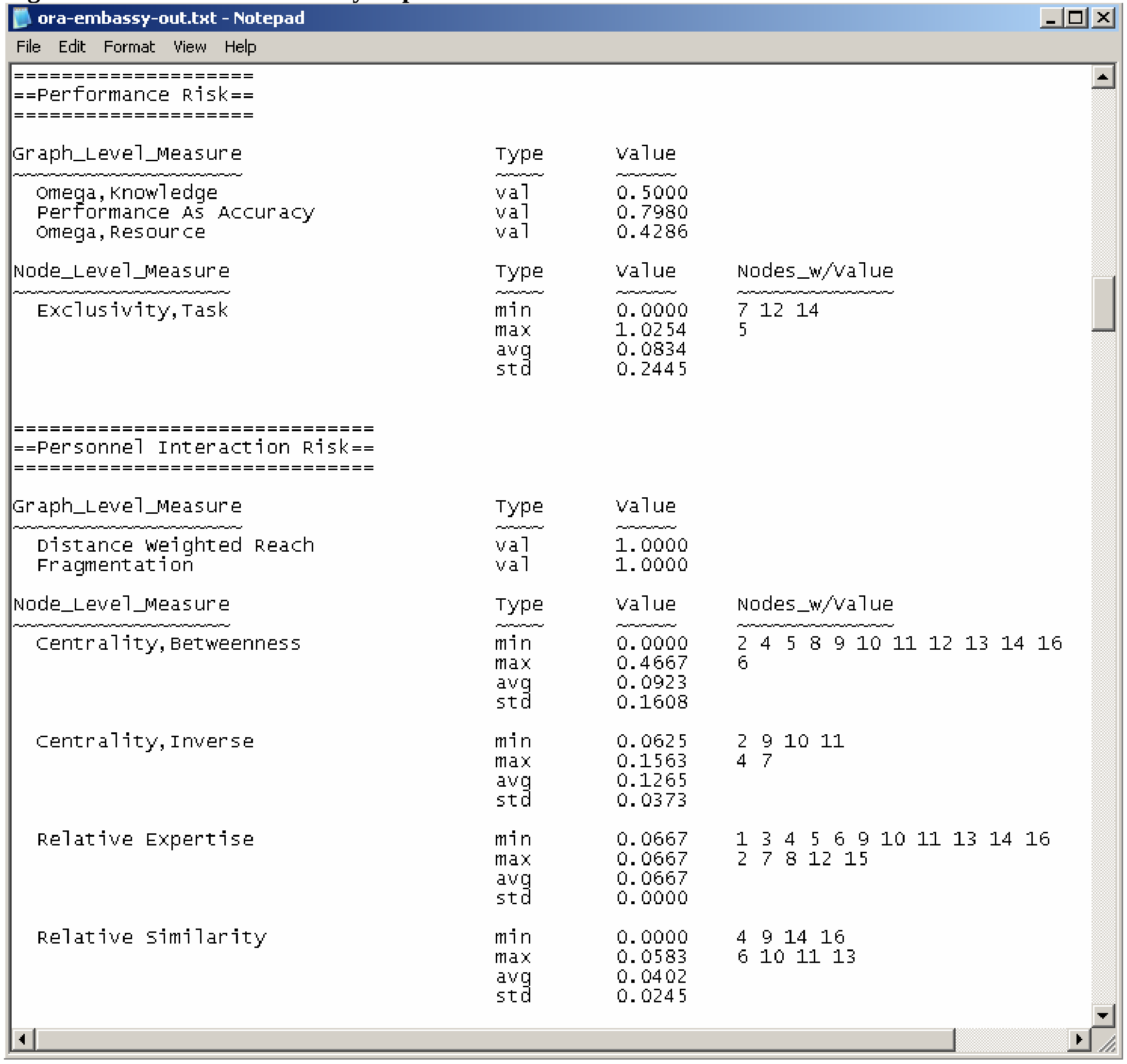
Table 2: Illustrative metrics categorized by Input, Risk, and Output

Metric	Meaning	Output Level	Risk	Input Data
Degree Centrality	In the social network, number of others the person is connected to.	Node	Communication	Single-cell
Task Exclusivity	Detects agents who exclusively perform tasks.	Node	Critical Employee Performance	Single-cell
Cognitive Load	Measures the total amount of effort expended by each agent to do its tasks.	Node	Critical Employee	Multi-cell
Resource Congruence	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.	Graph	Resource Allocation, Task	Multi-cell

4. Reports

ORA generates text reports from the measure analysis. A report is a predefined output data format. ORA currently produces a single report, the Risk and Vulnerability Report. This report can be saved in one of three formats: plain text, CSV, or DyNetML. The Risk and Vulnerability Report is a risk audit of an organization, which groups the measures by risk category and lists the measure values. The three formats are three different mediums for outputting the report data. The DyNetML report format contains in one file the original input Meta-Matrix together with all measures computed on the Meta-Matrix. The user selects which of the reports to generate, and ORA creates separate output files for each. Figure 1 below shows a portion of the text file format of a Risk and Vulnerability Report for a single Meta-Matrix organization.

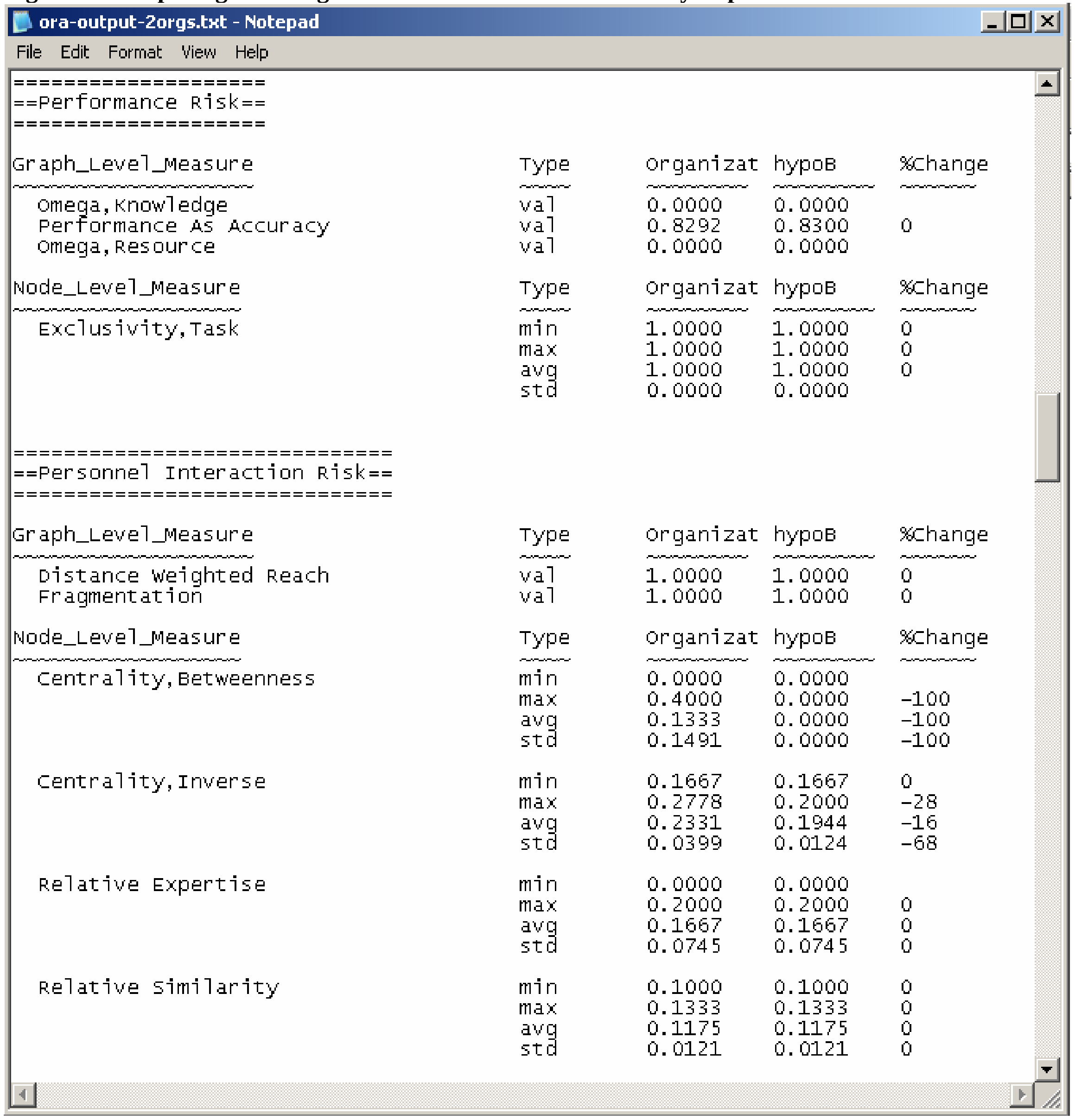
Figure 1: Risk and Vulnerability Report File



```
====================
==Performance Risk==
====================

Graph_Level_Measure                     Type      Value
~~~~~~~~~~~~~~~~~~~                     ~~~~      ~~~~~
  Omega,Knowledge                       val       0.5000
  Performance As Accuracy               val       0.7980
  Omega,Resource                        val       0.4286

Node_Level_Measure                      Type      Value     Nodes_w/Value
~~~~~~~~~~~~~~~~~~                      ~~~~      ~~~~~     ~~~~~~~~~~~~~
  Exclusivity,Task                      min       0.0000    7 12 14
                                        max       1.0254    5
                                        avg       0.0834
                                        std       0.2445


==============================
==Personnel Interaction Risk==
==============================

Graph_Level_Measure                     Type      Value
~~~~~~~~~~~~~~~~~~~                     ~~~~      ~~~~~
  Distance Weighted Reach               val       1.0000
  Fragmentation                         val       1.0000

Node_Level_Measure                      Type      Value     Nodes_w/Value
~~~~~~~~~~~~~~~~~~                      ~~~~      ~~~~~     ~~~~~~~~~~~~~
  Centrality,Betweenness                min       0.0000    2 4 5 8 9 10 11 12 13 14 16
                                        max       0.4667    6
                                        avg       0.0923
                                        std       0.1608

  Centrality,Inverse                    min       0.0625    2 9 10 11
                                        max       0.1563    4 7
                                        avg       0.1265
                                        std       0.0373

  Relative Expertise                    min       0.0667    1 3 4 5 6 9 10 11 13 14 16
                                        max       0.0667    2 7 8 12 15
                                        avg       0.0667
                                        std       0.0000

  Relative Similarity                   min       0.0000    4 9 14 16
                                        max       0.0583    6 10 11 13
                                        avg       0.0402
                                        std       0.0245
```

ORA can compare two Meta-Matrix organizations. The user can select any two meta-matrices and then generate a Risk and Vulnerability Report that compares the two organizations. Figure 2 displays a portion of one such report; it is similar to the single organization report, but it contains a side by side listing of the measure values for each organization followed by the percent by which the measure values differ.

Figure 2: Comparing Two Organizations: Risk and Vulnerability Report

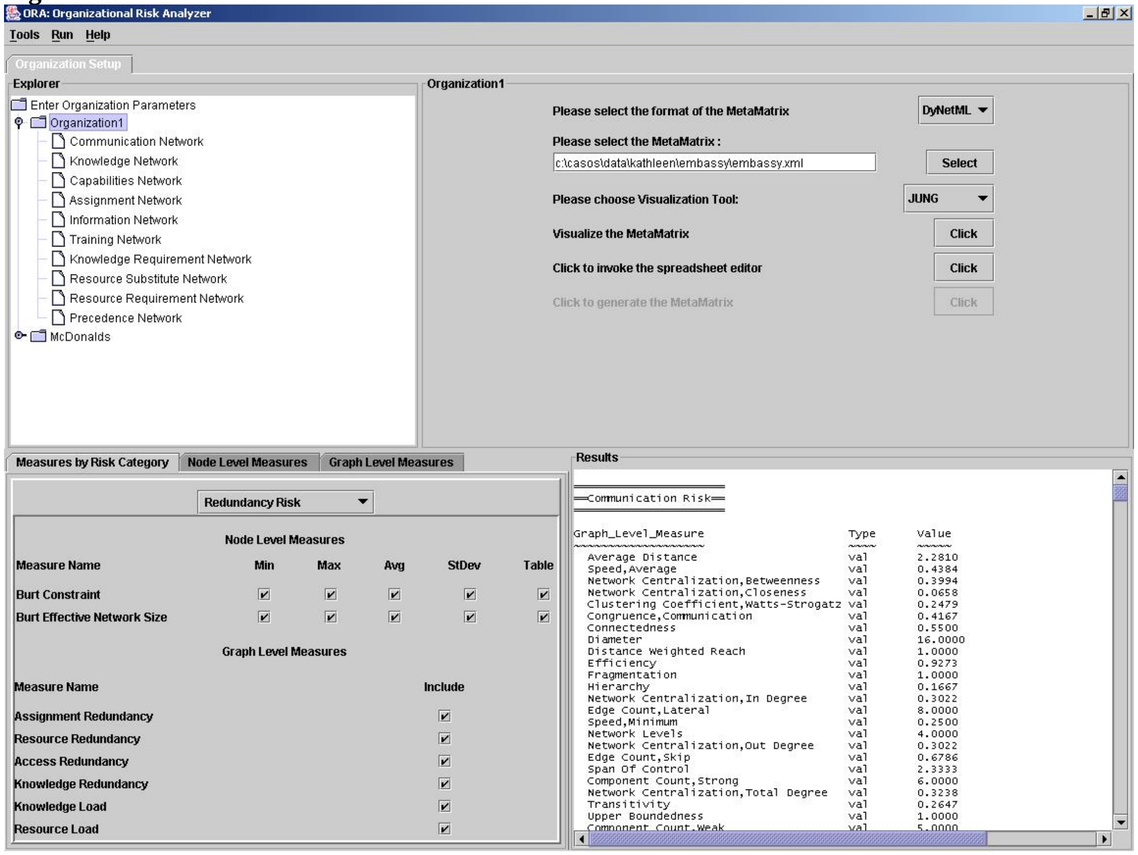


```
====================
==Performance Risk==
====================

Graph_Level_Measure                     Type      Organizat  hypoB      %Change
~~~~~~~~~~~~~~~~~~~                     ~~~~      ~~~~~~~~~  ~~~~~~~~~  ~~~~~~~
  Omega,Knowledge                       val       0.0000     0.0000
  Performance As Accuracy               val       0.8292     0.8300     0
  Omega,Resource                        val       0.0000     0.0000

Node_Level_Measure                      Type      Organizat  hypoB      %Change
~~~~~~~~~~~~~~~~~~                      ~~~~      ~~~~~~~~~  ~~~~~~~~~  ~~~~~~~
  Exclusivity,Task                      min       1.0000     1.0000     0
                                        max       1.0000     1.0000     0
                                        avg       1.0000     1.0000     0
                                        std       0.0000     0.0000


==============================
==Personnel Interaction Risk==
==============================

Graph_Level_Measure                     Type      Organizat  hypoB      %Change
~~~~~~~~~~~~~~~~~~~                     ~~~~      ~~~~~~~~~  ~~~~~~~~~  ~~~~~~~
  Distance Weighted Reach               val       1.0000     1.0000     0
  Fragmentation                         val       1.0000     1.0000     0

Node_Level_Measure                      Type      Organizat  hypoB      %Change
~~~~~~~~~~~~~~~~~~                      ~~~~      ~~~~~~~~~  ~~~~~~~~~  ~~~~~~~
  Centrality,Betweenness                min       0.0000     0.0000
                                        max       0.4000     0.0000     -100
                                        avg       0.1333     0.0000     -100
                                        std       0.1491     0.0000     -100

  Centrality,Inverse                    min       0.1667     0.1667     0
                                        max       0.2778     0.2000     -28
                                        avg       0.2331     0.1944     -16
                                        std       0.0399     0.0124     -68

  Relative Expertise                    min       0.0000     0.0000
                                        max       0.2000     0.2000     0
                                        avg       0.1667     0.1667     0
                                        std       0.0745     0.0745     0

  Relative Similarity                   min       0.1000     0.1000     0
                                        max       0.1333     0.1333     0
                                        avg       0.1175     0.1175     0
                                        std       0.0121     0.0121     0
```

5. User Interface Components

ORA has a Java user interface for cross platform compatibility. The interface contains three main components: 1) Meta-Matrix Manager, 2) Measure Manager, and 3) Output Panel. In addition, it contains the following sub-components: the Visualizer, the Optimizer, and the Regression Tool. Each of these will be briefly described in turn. The descriptions will refer to Figure 3 below, which shows the three main components of the ORA interface.

Figure 3: ORA Interface



Please select the format of the MetaMatrix: DynetML
Please select the MetaMatrix : c:\casos\data\kathleen\embassy\embassy.xml Select
Please choose Visualization Tool: JUNG
Visualize the MetaMatrix: Click
Click to invoke the spreadsheet editor: Click
Click to generate the MetaMatrix: Click

Redundancy Risk

Node Level Measures

Measure Name	Min	Max	Avg	StDev	Table
Burt Constraint	✓	✓	✓	✓	✓
Burt Effective Network Size	✓	✓	✓	✓	✓

Graph Level Measures

Measure Name	Include
Assignment Redundancy	✓
Resource Redundancy	✓
Access Redundancy	✓
Knowledge Redundancy	✓
Knowledge Load	✓
Resource Load	✓

```
==Communication Risk==

Graph_Level_Measure                        Type      Value
~~~~~~~~~~~~~~~~~~~                        ~~~~      ~~~~~
  Average Distance                         val       2.2810
  Speed,Average                            val       0.4384
  Network Centralization,Betweenness       val       0.3994
  Network Centralization,Closeness         val       0.0658
  Clustering Coefficient,Watts-Strogatz    val       0.2479
  Congruence,Communication                 val       0.4167
  Connectedness                            val       0.5500
  Diameter                                 val       16.0000
  Distance Weighted Reach                  val       1.0000
  Efficiency                               val       0.9273
  Fragmentation                            val       1.0000
  Hierarchy                                val       0.1667
  Network Centralization,In Degree         val       0.3022
  Edge Count,Lateral                       val       8.0000
  Speed,Minimum                            val       0.2500
  Network Levels                           val       4.0000
  Network Centralization,Out Degree        val       0.3022
  Edge Count,Skip                          val       0.6786
  Span Of Control                          val       2.3333
  Component Count,Strong                   val       6.0000
  Network Centralization,Total Degree      val       0.3238
  Transitivity                             val       0.2647
  Upper Boundedness                        val       1.0000
  Component Count,Weak                     val       5.0000
```

5.1 Meta-Matrix Manager

The unifying concept in ORA is the Meta-Matrix. The user needs to be able to manage multiple organizations, both those entered as original input and those output by the Optimizer. These are collected and managed in the Meta-Matrix Manager, which occupies the upper half of Figure 3. From this panel, the user can add organizations to the collection, rename them, and specify data files for each network of the Meta-Matrix. Most user actions in ORA require the selection of one or more organizations as input, and the available organizations are those entered into the Meta-Matrix Manager.

5.2 Measure Manager

The Measure Manager is a separate panel, seen in the lower left panel of Figure 3, that gives the user different views of the measures, according to the categories described above: Risk Category, Node Level, and Graph Level. These are all views of the same set of underlying measures, so selecting or unselecting a measure in one view is reflected in all other views.

5.3 Output Panel

The Output Panel is a text panel that gives the user immediate feedback. The Output Panel displays status information from user actions, and also displays the text content of measure reports. The Output Panel displaying a Risk Report can be seen in the lower right half of Figure 3.

5.4 Tools

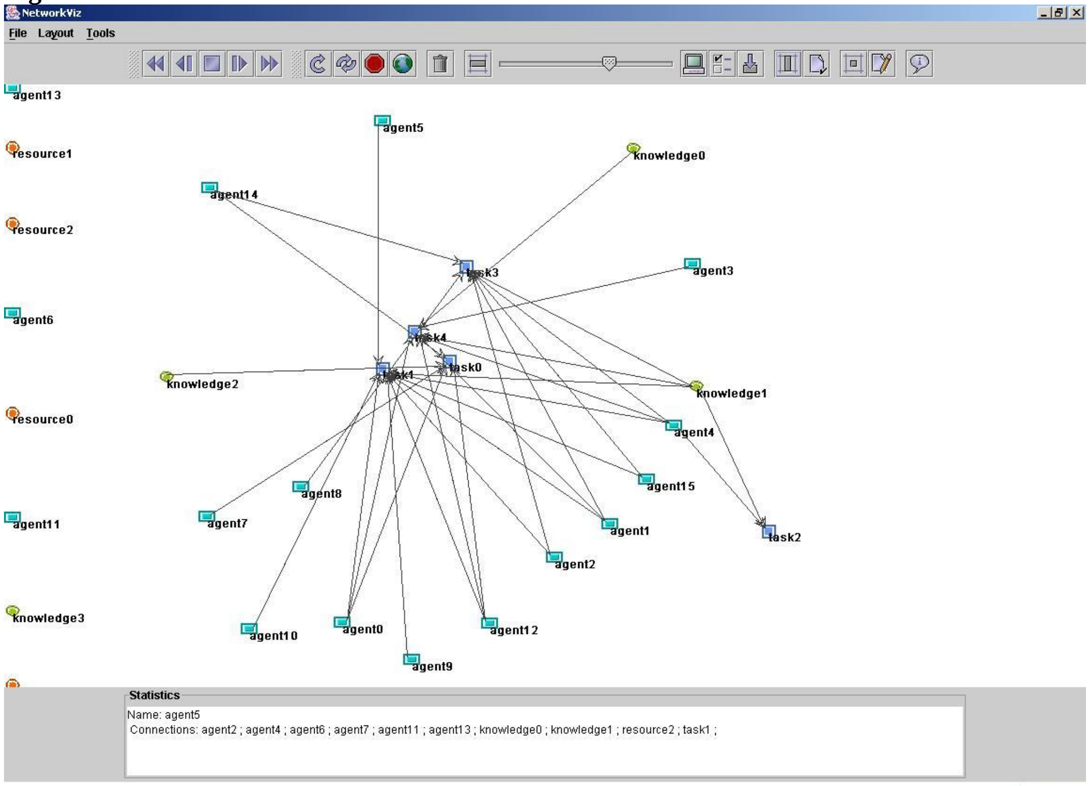
The Visualizer, Optimizer, and Regression Tool are invoked from the main menu and are contained within pop-up windows. The different popup components of ORA constitute an integrated graphical user interface that has proven to be extensible and flexible.

5.4.1 Visualization

The Meta-Matrix contains multiple node entities and different types of edges. Most existing visualization packages cannot display multiple network types simultaneously, and therefore are not suitable for visualizing the Meta-Matrix. ORA contains two integrated visualization packages for displaying an entire Meta-Matrix: NetworkViz and Jung.

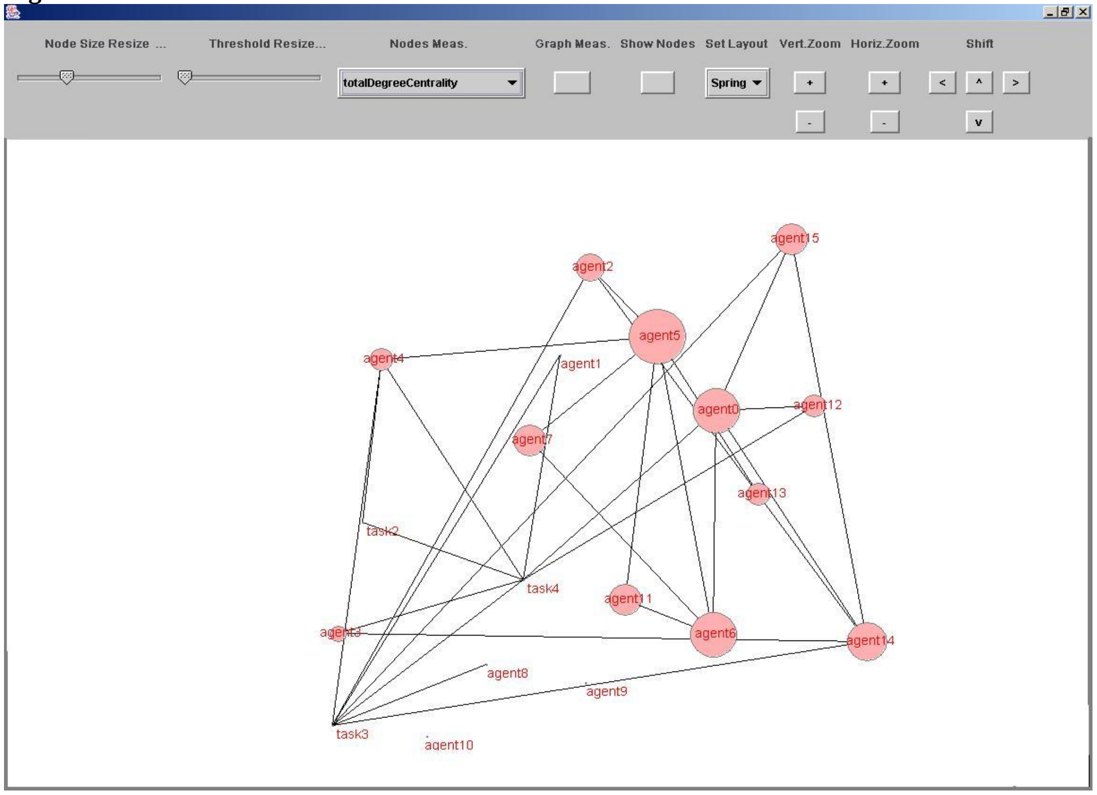
NetworkViz was developed at CASOS specifically for visually analyzing Meta-Matrix data, and is capable of displaying all of the networks of the Meta-Matrix simultaneously. NetworkViz can also display specific parts of the Meta-Matrix, for example, a single network, or all networks defined on one or more node entities. In this manner, the user can isolate and visualize portions of the Meta-Matrix that are of interest. If ORA has computed measures for the Meta-Matrix, then NetworkViz displays them. For example, all Node Level measures computed for a particular Agent node are displayed in a pop-up window when the node is right clicked. Figure 4 contains a sample Meta-Matrix visualization using NetworkViz.

Figure 4: NetworkViz Visualizer



Name: agent5
Connections: agent2 ; agent4 ; agent6 ; agent7 ; agent11 ; agent13 ; knowledge0 ; knowledge1 ; resource2 ; task1 ;

Figure 5: JUNG Visualizer



Nodes Meas.: totalDegreeCentrality

Set Layout: Spring

The Jung visualization package is an open-source code project from Stanford adapted and integrated into ORA. It offers three different layout algorithms, and has been customized to display Node Level and Graph Level measures computed on the Meta-Matrix. With Jung the user can choose portions of the Meta-Matrix to display based on measure values, for example, Figure 5 displays the Agent nodes with sizes proportional to their Total Degree Centrality.

5.4.2 Optimizer

Having detected the risk and vulnerabilities of an organization's design structure, the Optimizer tool in ORA allows the user to change the structure according to user specified criteria. The user selects a single measure or a linear combination of measures to be an objective function, and the Optimizer produces an organization that maximizes or minimizes the objective function by adding and removing relationships (i.e. edges) between node entities. Because the output of the Optimizer is a Meta-Matrix, it can be input to ORA for measure analysis and visualization. Details of the Optimizer can be found in the Optimizer Technical Report.

5.4.3 Regression Tool

The Regression Tool allows the user to compare two vector valued measures, plotting the two vectors in coordinate space with a linear regression line. The two vectors can be the same measure computed on two different organizations, two measures computed on the same organization, or two different measures computed on two different organizations. The plot output can be saved to a file.

6. System Requirements

ORA 1.2 is the latest version of ORA and it runs on any Windows 2000 or XP machine running on an Intel processor. The C++ back-end source code is written so as to be compatible with platforms and processors, and is being ported and tested on other platforms and processors.

7. Conclusion

ORA advances the state of the art in network analysis tools by being organized around the unifying concept of the Meta-Matrix. Measures are organized to facilitate their coherent use. In particular, they are categorized by how they measure the risk and vulnerability of an organization's design structure. ORA reads and writes in multiple data formats and is interoperable with existing network analysis software. Entire Meta-Matrices can be visualized using different layout algorithms. The integrated Optimizer adapts an organization's design structure according to user specified criteria, and the resulting organization can be input into ORA and analyzed and visualized. The computational back end employs NetStatPlus, an open source C++ library of SNA and DNA routines. The Java graphical user interface is designed for ease of use and for extensibility and flexibility as new features are added. ORA is being actively developed and tested in a wide range of contexts.

8. Future Work

Future work in ORA will address all aspects of its core functionality, including (1) managing Meta-Matrix organizations; (2) measure presentation and selection; (3) network visualization; (4) tool sub-components; and (5) generating reports.

Currently a Meta-Matrix can contain only one matrix of each network type. Thus a Meta-Matrix cannot have a Communication Network and a Friendship Network, both of type Agent x Agent. Similarly, time period data for a matrix type is not possible. Future versions will extend the Meta-Matrix Manager to allow multiple matrices of a single type.

The Measures Manager currently does not allow the user to specify the input for measures. Certain measures have predefined input, and so specifying input matrices is unnecessary. For example the Actual Knowledge Workload measure takes always takes as input the following matrices: Agent x Knowledge, Knowledge x Task, and Agent x Task. Other measures, for example Square measures, can operate on any square input matrix. For example, the Betweenness Centrality measure takes any square matrix as input. Currently, such measures run on a pre-determined, default matrix which is not user selectable.

Another tool currently being developed for ORA called the Matrix Tool, which displays matrix data in an editable spreadsheet window. Individual networks can be displayed, or an entire Meta-Matrix. The Matrix Tool lets the user manipulate matrices, such as performing matrix algebra, and computing the Intersection, Union, and Central Matrix of a collection of matrices. By loading a Meta-Matrix in one format and saving in another, the user can convert data from one format to another. The Matrix Tool will be included in the next release of ORA.

Finally, ORA will be extended to provide multiple report types. Currently, only the Risk and Vulnerability Report is available. Future report types will output alphabetical lists of measures, or measures categorized by node, and graph level. The Output Panel of the user interface will be extended to display multiple output files, allowing the user to quickly organize and view report files.

References

[1] Ashworth, M. and Kathleen M. Carley, 2003, Critical Human Capital, Working Paper, CASOS, Carnegie Mellon, Pittsburgh PA.

[2] Borgatti, S.P. 2003. The Key Player Problem. Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, R. Breiger, K. Carley, & P. Pattison (Eds.) Committee on Human Factors, National Research Council, 241-252.

[3] Borgatti, S.P., Everett, M.G. and Freeman, L.C. 2002. Ucinet for Windows: Software for Social Network Analysis. Harvard: Analytic Technologies.

[4] Brass, D. 1984. Being in the right place: A structural analysis of individual influence in an organization. Admin. Sci. Quart. 26 331-348.

[5] Carley, Kathleen M. 2002, “Smart Agents and Organizations of the Future” The Handbook of New Media. Edited by Leah Lievrouw & Sonia Livingstone, Ch. 12 pp. 206-220, Thousand Oaks, CA, Sage.

[6] Carley, Kathleen M. and Yuqing Ren, 2001, “Tradeoffs Between Performance and Adaptability for C3I Architectures.” In Proceedings of the 2001 Command and Control Research and Technology Symposium. Conference held in Annapolis, Maryland, June, 2001. Evidence Based Research, Vienna, VA.

[7] Carley, Kathleen M. and Vanessa Hill, 2001, “Structural Change and Learning Within Organizations”. In Dynamics of Organizations: Computational Modeling and Organizational Theories. Edited by Alessandro Lomi and Erik R. Larsen, MIT Press/AAAI Press/Live Oak, Ch. 2. pp 63-92.

[8] Krackhardt, David and Kathleen M. Carley, 1998, “A PCANS Model of Structure in Organization” Pp. 113-119 in Proceedings of the 1998 International Symposium on Command and Control Research and Technology. Conference held in June. Monterray, CA. Evidence Based Research, Vienna, VA.

[9] Thompson, J. D. 1967. Organizations in Action. McGraw-Hill, New York.

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

Appendix A – ORA Measures

A **network** N consists of two sets of nodes, called U and V, and a set of edges $E \subset UxV$. An element e = (i,j) in E indicates 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 **unimodal**; otherwise the network is **bimodal**. For our purposes, unimodal networks will not contain self loops, which means that $(i,i) \notin E$ for $i \in V$.

An **organization** is a collection of networks. A **measure** is a function that maps one or more networks to R^n. Measures are often scalar (n=1) or vector valued with n = |V| or n=|U|.

When defining or implementing measures, a network can be represented as (1) a graph, or as (2) an adjacency matrix. To represent a *unimodal* network as a graph, let G=(V,E), where V is the network's nodes, and E are the ties; *bimodal* networks will not be represented as graphs. Both unimodal and bimodal 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 if $(i,j) \in E$, else let M(i,j)=0. Then M is called the adjacency matrix representation of network N. Unimodal networks are also called **square** networks because their adjacency matrix is square; the diagonal is zero diagonal because there are no self-loops.

Define the following sets of nodes: Agents, Knowledge, Resources, and Tasks. The following networks defined on these node sets are used throughout the documentation:

Symbol	**Node Sets**		**Name**
	U	**V**	
A	Agent	Agent	Communication Network
AK	Agent	Knowledge	Knowledge Network
AR	Agent	Resource	Capabilities Network
AT	Agent	Task	Assignment Network
K	Knowledge	Knowledge	Information Network
KR	Knowledge	Resource	Training Network
KT	Knowledge	Task	Knowledge Requirement Network
R	Resource	Resource	Resource Substitute Network
RT	Resource	Task	Resource Requirement Network
T	Task	Task	Precedence Network

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 ith row and jth column of Matrix
Matrix(i,:)	= ith row vector of Matrix
Matrix(:,j)	= jth 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	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, 2003	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, 2003	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.
Average Distance	The average shortest path length between nodes, excluding infinite distances. **Type** Graph Level **Input** A:binary, square **Output** $\Re \in [0,1]$	NetStat	Let G=(V,E) represent a square network. Define a set S of all pairs (i,j) of nodes such that i can reach j. Then average the shortest paths. let S = {(i,j) | j is reachable in G from j } Then, Average Distance = $\frac{\sum_{(i,j)\in S} d_G(i,j)}{\lvert S \rvert}$.

Centrality, Betweenness	The Betweenness Centrality of node v in a network 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. 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, Eigenvector	Calculates the eigenvector of the largest positive eigenvalue of the adjacency matrix representation of a square network. **Type** Node Level **Input** N:square, symmetric **Output** $\Re \in [0,1]$	Bonacich P, 1972	Calculates the eigenvector of the largest positive eigenvalue of the adjacency matrix representation of a square network. A Jacobi method is used to compute the eigenvalues and vectors.
Centrality, In Degree	The In Degree Centrality of a node in a unimodal network is its normalized in-degree. **Type** Node Level **Input** N:square **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994	Let G=(V,E) be the graph representation of a square network and fix a node v. let deg = $card\{u \in V \mid (u,v) \in E\}$, this is the in-degree of node v. The In Degree Centrality of node v = deg / (|V|-1)

Centrality, Information	Calculate the Stephenson and Zelen information centrality measure for each node. **Type** Node Level **Input** N:square, symmetric **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994 (pg 195)	Calculates the measure described on pg 195-6 of Wasserman and Faust. Nodes with 0 degree are first removed from the network, and the measure computed on the resulting sub-graph. The removed nodes are given centrality value 0.
Centrality, Inverse Closeness	The average closeness of a node to the other nodes in a network. Inverse Closeness is the sum of the inverse distances between a node and all other nodes. This is defined for directed networks. **Type** Node Level **Input** N:square **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994 (pg 195)	Let G=(V,E) be the graph representation of the square network. Fix v$\in$V. let dist = $\sum_{i \in V} \frac{1}{d_G(v,i)}$, where $\frac{1}{d_G(i,i)}$ = 0 and $\frac{1}{d_G(v,i)}$ = 0 if i is not reachable from v. Then Inverse Closeness Centrality of node v = dist/(|V|-1).
Centrality, Out Degree	The Out Degree Centrality of a node in a square network is its normalized out-degree. **Type** Node Level **Input** N:square **Output** $\Re \in [0,1]$	Wasserman and Faust, 1994	Let G=(V,E) be the graph representation of a square network and fix a node v. let deg = $card\{u \in V \mid (v,u) \in E\}$, this is the out-degree of node v. The Out Degree Centrality of node v = deg / (|V|-1)
Centrality, Total Degree	The Total Degree Centrality of a node in a square network is its normalized in plus out degree. **Type** Node Level **Input** N:square, undirected **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 v. let deg = $card\{u \in V \mid (v,u) \in E \vee (u,v) \in E\}$, this is the total degree of node v. The Total Degree Centrality of node v = deg / 2*(|V|-1)

Clustering Coefficient, Watts-Strogatz	Measures the degree of clustering in a network by averaging the clustering coefficient of each node i, defined as the ratio of the number of triangles connected to i to the number of triples centered at i. **Type** Graph Level **Input** N: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 v $\in$ V define the following: let $in_v = \{i \in V \mid (i,v) \in E\}$ let $out_v = \{i \in V \mid (v,i) \in E\}$ let $inconnect_v = \{(i,j) \in E \mid i, j \in in_v\}$ let $outconnect_v = \{(i,j) \in E \mid i, j \in out_v\}$ Then compute for each node v $\in$ V its Clustering Coefficient cc_v using (1) in-degree, (2) out-degree, or (3) total degree. (1) let $cc_v = \frac{\lvert inconnect_v \rvert}{\lvert in_v \rvert^2 - \lvert in_v \rvert}$, if $\lvert in_v \rvert > 1$, else $cc_v = 0$. (2) let $cc_v = \frac{\lvert outconnect_v \rvert}{\lvert out_v \rvert^2 - \lvert out_v \rvert}$, if $\lvert out_v \rvert > 1$, else $cc_v = 0$. (3) let $cc_v = \frac{1}{2}\left(case(1) + case(2)\right)$ Then Clustering Coefficient for the graph = $\left(\sum_{v \in V} cc_v\right) / \lvert V \rvert$

Cognitive Load	Measures the total amount of effort expended by each agent to do its tasks. 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 = $(x_1 + x_2 + x_3 + x_4 + x_5 + x_6)/6$

Communication	Measures the communication need of agents to complete their assigned tasks. **Type** Node Level **Input** A:binary; AT:binary; AR:binary; RT:binary, T:binary **Output** $\Re \in [0,1]$	Carley, 2003	Communication uses the concepts from Communication Congruence: Handoff, Co-Assignment, and Negotiation. let H, C, and N be defined as in Communication Congruence. let M(i,j) = [A + (H+H') + C + (N+N')](i,j) > 0, and M(i,i) = 0 Note that the transpose of H and N is used to make the communication reciprocal. let d = sum(M(i,:)) let d = d / (|A|-1), normalizing d to be in [0,1] Then Communication for agent i is d.
Communicative Need	Measures the percentage of reciprocal edges in a network. **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, Strong	The number of strongly connected components in a network. **Type** Graph Level **Input** N:square **Output** $Z \in [0,\|V\|]$	Wasserman and Faust, 1994 (pg 109)	Given a square network represented by a graph G=(V,E), the Strong Component Count is the number of strongly connected components in G. This is computed directly on G, whether or not G is directed.
Component Count, Weak	The number of weakly connected components in a network. **Type** Graph Level **Input** N:square, symmetric **Output** $Z \in [0,\|V\|]$	Wasserman and Faust, 1994 (pg 109)	Given a square, symmetric network represented by a graph G=(V,E), the Weak Component Count is the number of connected components in G. Such components are called "weak" because the graph G is undirected.

Congruence, Communication	Measures to what extent the agents communicate when and only when it is needful to complete tasks. Perfect congruence requires reciprocal communication. **Type** Graph Level **Input** A:binary; AT:binary; AR:binary; RT:binary, T:binary **Output** $\Re \in [0,1]$	Carley, 2002	Communication Congruence = 1 iff agents communicate when and only when it is needful to complete their tasks. There are three task related reasons when agents i and j need to communicate: (a) *Handoff*: if i is assigned to a task s and j is assigned to a task t and s directly precedes task t (b) *Co-Assignment*: if i is assigned to a task s and j is also assigned to s (c) *Negotiation*: 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. 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 Note that C is always symmetric, but not necessarily H and N. let Q(i,j) = [(H+H') + C + (N+N')](i,j) > 0. Communication Congruence requires reciprocal communication, explaining the transposes of H and N to make them symmetric. let d = hamming distance between Q and A, which measures the degree to which communication differs from that which is needed to do tasks. The maximum value for d is d_max = |A|*(|A|-1) Then Communication Congruence = 1 - (d /d_max), which is in [0,1].
Congruence, Knowledge	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** $\Re \in [0,1]$	Carley, 2002	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

Congruence, Resource	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** $\Re \in [0,1]$	Carley, 2002	Identical to Knowledge Congruence with AR replaced by AK and KT replaced by RT.
Connectedness	Measures the degree to which a square network's underlying (undirected) network is connected. **Type** Graph Level **Input** N:square, symmetric **Output** $\Re \in [0,1]$	Krackhardt, 1994	The Connectedness of a square, symmetric network is the Density of its Reachability Network.
Constraint, Burt	The degree to which each node in a square network is constrained from acting because of its existing links to other nodes. **Type** Node Level **Input** N:square **Output** $\Re \in [0,1]$	Burt, 1992	This is the Constraint measure described by Equ. 2.4 on pg. 55 of Burt, 1992. Note that the matrix Z is the adjacency matrix representation of the network N.
Density	The ratio of the number of edges versus the maximum possible edges for a network. **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 unimodal, then m=n and M has a zero diagonal, and therefore Density = sum(M)/(m*(m-1)). If the network is symmetric, then Density is multiplied by two. For bimodal networks, Density = sum(M)/(m*n).
Diameter	The maximum shortest path length between any two nodes in a unimodal network G=(V,E). If there exist i,j in V such that j is not reachable from i, then |V| is returned. **Type** Graph Level **Input** N:square **Output** $Z \in [0,\|V\|]$	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.

Distance Weighted Reach	A generalization of graph theoretic distance, this measures the distance from a *set* of nodes in the network to all other nodes. **Type** Graph Level **Input** N:square, undirected **Output** $\Re \in [0,1]$	Borgatti, 2003	Consider a square, undirected network represented by G=(V,E). let S $\subseteq$ V For any j $\notin$ S, define $d_G(S,j) = \min\{d_G(i,j) \mid i \in S\}$. Then, Distance Weighted Reach $= 1 - \frac{\sum_{j \notin S} \frac{1}{d_G(S,j)}}{\lvert V - S \rvert}$,
Diversity, Knowledge	The distribution of difference in idea sharing. This is the Herfindahl-Hirshman index applied to column sums of AK. **Type** Graph Level **Input** AK:binary **Output** $\Re \in [0,1]$		This is the Herfindahl-Hirshman index (economics: sum of the squares of each firm's market share) applied to the normalized column sums of AK. This measures the degree to which knowledge is equally known. let $w_k = \sum_{i=1}^{\lvert A \rvert} A(i,k)$, for $1 \le k \le \lvert K \rvert$ let W = $\sum_{k=1}^{\lvert K \rvert} w_k$ Then Diversity $= 1 - \sum_{k=1}^{\lvert K \rvert} (w_k / W)^2$
Diversity, Resource	The distribution of difference in resource sharing. This is the Herfindahl-Hirshman index applied to column sums of AR. **Type** Graph Level **Input** AR:binary **Output** $\Re \in [0,1]$		Identical to Knowledge Diversity, with AK replaced by AR.
Edge Count, Lateral	The percentage of lateral edges in a unimodal network. 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 unimodal network. And fix a node x $\in$ V to be the root node. Let S = {(i,j) $\in$ E | $d_G(x,i) = d_G(x,j)$ } Then Lateral Edge Count = |S| / |E|

Edge Count, Pooled	The percentage of pooled edges in a unimodal network. A pooled is an edge (i,j) such that there exists at least one other edge (i,k) in the network, and $k \neq j$. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Carley, 2002	Let M be the adjacency matrix representation of a unimodal network. Let S = { (i,j) | $M(i,j)=1 \wedge sum(M(:,j))>1$ } In other words: edge (i,j) is a pooled edge iff the in-degree of node j > 1. Then Pooled Edge Count = |S| / |E|
Edge Count, Reciprocal	The percentage of edges in a unimodal network that are reciprocated (also called Reciprocity). An edge (i,j) in the network is reciprocated if edge (j,i) is also in the network. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$		Let G=(V,E) be the graph representation of a unimodal network. Let S = card{$(i,j) \in E$ | i<j, $(j,i) \in E$ } Then Reciprocal Edge Count = |S| / |E|
Edge Count, Sequential	The percentage of edges in a unimodal network 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:square **Output** $\Re \in [0,1]$	Carley, 2002	Let G=(V,E) be the graph representation of a unimodal 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 fraction of edges in a unimodal network that skip levels. An edge (i,j) is a skip edge if there is a path from node i to node j even after the edge (i,j) is removed. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Carley, 2002	A skip edge in a unimodal network represented by G=(V,E) is an edge $(i,j) \in E$ 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 a node's ego network based on redundancy of ties. **Type** Node Level **Input** N:square **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 matrix Z is the adjacency matrix representation of the network N.

Efficiency	The degree to which each component in a network contains the minimum edges possible to keep it connected. **Type** Graph Level **Input** N:square, symmetric **Output** $\Re \in [0,1]$	Krackhardt, 1994	Let G=(V,E) be the graph representation of a square, symmetric network. let n = number of components in G let c_i = number of nodes in component i let penalty = |E| - |V| + C let maxPenalty = C - |V| + $\sum_i c_i(c_i - 1)/2$ Then Efficiency = 1 - penalty/maxPenalty
Exclusivity, Knowledge	Detects agents who have singular knowledge. **Type** Node Level **Input** AK:binary **Output** $\Re \in [0,1]$	Ashworth, 2003	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	Detects agents who have singular resource access. **Type** Node Level **Input** AR:binary **Output** $\Re \in [0,1]$	Ashworth, 2003	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	Detects agents who exclusively perform tasks. **Type** Node Level **Input** AT:binary **Output** $\Re \in [0,1]$	Ashworth, 2003	The Task Exclusivity Index (TEI) for agent i is defined exactly as for Knowledge Based Exclusivity, but with the matrix AK replaced by AT.
Fragmentation	The proportion of nodes in a network that are disconnected. **Type** Graph Level **Input** N:square, undirected **Output** $\Re \in [0,1]$	Borgatti, 2003	Consider a square, undirected network represented by G=(V,E). let n = |V| let s_k be the number of nodes in the kth component of G, $1 \le k \le n$ Then, Fragmentation = $1 - \frac{\sum_k s_k(s_k - 1)}{n(n-1)}$.
Hierarchy	The degree to which a unimodal network exhibits a pure hierarchical structure. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Krackhardt, 1994	Let N be a unimodal network. The Hierarchy of N is the Reciprocity of the Reachability Network for N.

Interdependence	The percentage of edges in a unimodal network 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|
Interlockers and Radials	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, $1 \le i \le \lvert V \rvert$. 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** $\Re \in [0, \lvert K \rvert]$	Carley, 2002	Knowledge Load = sum(AK)/ (|A|)
Load, Resource	Average number of resources per agent. **Type** Graph Level **Input** AR:binary **Output** $\Re \in [0, \lvert R \rvert]$	Carley, 2002	Resource Load = sum(AR)/ (|A|)
Negotiation, Knowledge	The extent to which agents need to negotiate with each other because they lack the knowledge to complete their assigned tasks. **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 | $1 \le i \le \lvert T \rvert$, $\exists$ j : Need(i,j) < 0 } Then Need for Negotiation = |S| / |T|
Negotiation, Resource	The extent to which agents need to negotiate with each other because they lack the resources to complete their assigned tasks. **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 defined for directed and undirected 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 $\bar{d} = \max\{d_i \| 1 \le i \le n\}$ 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 defined only for connected, undirected 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 $\bar{d} = \max\{d_i \| 1 \le i \le n\}$ 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 degree of the column nodes of a network. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	NetStat	Let N be a network with n column nodes. let d_j = degree of column node j, $1 \le j \le n$ let $\bar{d} = \max\{d_j \| 1 \le j \le n\}$ Then Column Degree Network Centralization = $\left(\sum_{1 \le j \le n} \bar{d} - d_j\right) / (n)$.
Network Centralization, In Degree	A centralization of a square network based on the In-Degree Centrality of each node. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	NetStat	Let N be a unimodal network with n nodes. let d_i = In Degree Centrality of node i let $\bar{d} = \max\{d_i \| 1 \le i \le n\}$ Then In Degree Network Centralization = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / D$, where D = (n-2) if N is undirected, and (n-1) otherwise.

Network Centralization, Out Degree	A centralization of a square network based on the Out-Degree Centrality of each node. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	NetStat	Let N be a unimodal network with n nodes. let d_i = Out Degree Centrality of node i let $\bar{d} = \max\{d_i \mid 1 \le i \le n\}$ Then Out Degree Network Centralization = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / D$, where D = (n-2) if N is undirected, and (n-1) otherwise.
Network Centralization, Row Degree	A centralization based on the degree of the row nodes in a network. **Type** Graph Level **Input** N **Output** $\Re \in [0,1]$	NetStat	Let N be a network with n row nodes. let d_j = degree of row node j, $1 \le j \le n$ let $\bar{d} = \max\{d_j \mid 1 \le j \le n\}$ Then Row Degree Network Centralization = $\left(\sum_{1 \le j \le n} \bar{d} - d_j\right) / (n)$.
Network Centralization, Total Degree	A centralization of a square network based on total degree centrality of each node. **Type** Graph Level **Input** N:square **Output** $\Re \in [0,1]$	Freeman, 1979	Let N be a unimodal network with n nodes. let d_i = Total Degree Centrality of node i let $\bar{d} = \max\{d_i \mid 1 \le i \le n\}$ Then Total Degree Network Centralization = $\left(\sum_{1 \le i \le n} \bar{d} - d_i\right) / (n-2)$.
Network Levels	The Network Level of a square network is the maximum Node Level of its nodes. **Type** Graph Level **Input** N:square **Output** $Z \in [0, \lvert V \rvert - 1]$	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$\in$V; j reachable from i in G }
Node Level	The Node Level for a node v in a square network is the longest shortest path from v to every node v can reach. If v cannot reach any node, then its level is 0. **Type** Node Level **Input** N:square **Output** $Z \in [0, \lvert V \rvert - 1]$	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$\in$V; j reachable from v in G }; if v cannot reach any nodes, then its level is 0.

Omega, Knowledge	The degree to which agents reuse knowledge while doing their tasks. **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	The degree to which agents reuse resources while doing their tasks. **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** AT:binary; AK: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), $i \in$ 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.

Personnel Cost	Total number of people reporting to an agent, plus its total knowledge, resources, and tasks. **Type** Node Level **Input** A:binary; AK:binary; AR:binary; AT:binary **Output** $\Re \in [0,1]$	Carley, 2003	Personnel Cost for agent i is defined as follows: Let d = sum(A(:,i)) + sum(AK(i,:)) + sum(AR(i,:)) + sum(AT(i,:)) The value is then normalized to be in [0,1]: Let d = d / ((|A|-1) + |K| + |R| + |T|) The Personnel Cost for agent i is d.
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.
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 **Output** $\Re \in [0,(\|A\|-1)*\|R\|]$	Carley, 2002	This is the Column Redundancy of matrix AR.
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	The mean number of column node edges in excess of one. **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	The mean number of row node edges in excess of one. **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 = $\left(\sum_{\substack{j=1 \\ j \ne i}}^{\|A\|} RE(i,j)\right)/(\|A\|-1)$, 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,:)), $1 \le i \le \|A\|$ 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 = $\left(\sum_{\substack{j=1 \\ j \ne i}}^{\|A\|} RS(i,j) \right) / (\|A\| - 1)$, that is, the average of the non-diagonal elements of row i of RS.
Span of Control	The average number of out edges per node with non-zero out degrees. **Type** Graph Level **Input** N:square **Output** $\Re \in [0, \|V\| - 1]$	Carley, 2002	let S = set of nodes in V that have positive out-degree let K = $\sum_{i \in S} outDegree(i)$ Then Span of Control = K / |S|
Speed, Average	The average shortest path length between node pairs (i,j) where there is a path in the network from i to j. If there are no such pairs, then Average Speed is zero. **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 D={(i,j) | i,j $\in$ V, j reachable from i in G } Then Average Speed = $\left(\sum_{(i,j) \in D} d_G(i,j) \right) / \|D\|$
Speed, Minimum	The maximum shortest path length between node pairs (i,j) where there is a path in the network from i to j. If there are no such pairs, then Minimum Speed is zero. **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 | $1 \le i \le \|T\|$, $\exists$ 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 edge pairs {(i,j), (j,k)} in the network such that (i,k) is also an edge in the network.
	Type Graph Level
	Input N:square
	Output $\Re \in [0,1]$
		NetStat	Let G = (V,E) be the graph representation of the square network.
			let I = {(i,j,k) $\in V^3$ | i,j,k distinct }
			let Potential = { (i,j,k) $\in$ I | (i,j) $\in$ E, and (j,k) $\in$ E }
			let Complete = { (i,j,k) $\in$ Potential | (i,k) $\in$ E }
			Then Transitivity = |Complete| / |Potential|

Triad Count	The number of triads centered at each node in a square network. **Type** Node Level **Input** N:square of dimension |V| **Output** $Z \in [0,(\|V\|-1)(\|V\|-2)]$	NetStat	Let G=(V,E) represent a square network. And let Triad be a matrix of dimension |V|x|V|. Triad(i,i) = 0 Triad(i,j) = card{ k | k != i, k != j; A(i,j) $\wedge$ A(i,k) $\wedge$ A(k,j) }, i $\neq$ j Then the Triad count for agent i = sum(Triad(i,:))
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	
Weak Boundary Spanner	A node which if removed from a network creates a new component. **Type** Node Level **Input** N:square, symmetric **Output** Binary	Cormen, Leiserson, Riverest, Stein, 2001 p.558	A Weak Boundary Spanner is an *articulation point* of N, as defined in the referenced book.

Bibliography

[1] Ashworth, M. and K. M. Carley, 2003, Critical Human Capital, Working Paper, CASOS, Carnegie Mellon, Pittsburgh PA.

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

[3] Borgatti, S.P. 2003. The Key Player Problem. Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, R. Breiger, K. Carley, & P. Pattison (Eds.) Committee on Human Factors, National Research Council, 241-252.

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

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

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

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

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

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

[10] 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.

[11] 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

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

[13] 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

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

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

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

Appendix B – ORA Risk Category Measures

Metric	Meaning	Level	Risk	Data Needs
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.	Node	Critical Employee	Multi-cell
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.	Node	Critical Employee	Multi-cell
Actual Workload, Knowledge	The knowledge an agent uses to perform the tasks to which it is assigned.	Node	Resource Allocation	Multi-cell
Actual Workload, Resource	The resources an agent uses to perform the tasks to which it is assigned.	Node	Resource Allocation	Multi-cell
Average Distance	The average shortest path length between nodes, excluding infinite distances.	Graph	Communication	Single-cell
Centrality, Betweenness	The Betweenness Centrality of node v in a network 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.	Node	Communication	Single-cell
Centrality, Closeness	The average closeness of a node to the other nodes in a network. 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.	Node	Communication	Single-cell
Centrality, Eigenvector	Calculates the eigenvector of the largest positive eigenvalue of the adjacency matrix representation of a square network.	Node	Communication	Single-cell
Centrality, In Degree	The In Degree Centrality of a node in a unimodal network is its normalized in-degree.	Node	Communication	Single-cell
Centrality, Information	Calculate the Stephenson and Zelen information centrality measure for each node.	Node	Communication	Single-cell

Centrality, Inverse Closeness	The average closeness of a node to the other nodes in a network. Inverse Closeness is the sum of the inverse distances between a node and all other nodes. This is defined for directed networks.	Node	Communication	Single-cell
Centrality, Out Degree	The Out Degree Centrality of a node in a square network is its normalized out-degree.	Node	Communication	Single-cell
Centrality, Total Degree	The Total Degree Centrality of a node in a square network is its normalized in plus out degree.	Node	Communication	Single-cell
Clustering Coefficient, Watts-Strogatz	Measures the degree of clustering in a network by averaging the clustering coefficient of each node i, defined as the ratio of the number of triangles connected to i to the number of triples centered at i.	Graph	Communication	Single-cell
Cognitive Load	Measures the total amount of effort expended by each agent to do its tasks.	Node	Critical Employee	Multi-cell
Communication	Measures the communication need of agents to complete their assigned tasks.	Node	Communication	Multi-cell
Component Count, Strong	The number of strongly connected components in a network.	Graph	Communication	Single-cell
Component Count, Weak	The number of weakly connected components in a network.	Graph	Communication	Single-cell
Congruence, Communication	Measures to what extent the agents communicate when and only when it is needful to complete tasks. Perfect congruence requires reciprocal communication.	Graph	Communication	Multi-cell
Congruence, Knowledge	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.	Graph	Resource Allocation, Task	Multi-cell
Congruence, Resource	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.	Graph	Resource Allocation, Task	Multi-cell

Connectedness	Measures the degree to which a square network's underlying (undirected) network is connected.	Graph	Communication	Single-cell
Constraint, Burt	The degree to which each node in a square network is constrained from acting because of its existing links to other nodes.	Node	Critical Employee, Redundancy, Communication	Single-cell
Density	The ratio of the number of edges versus the maximum possible edges for a network.	Graph		Single-cell
Diameter	The maximum shortest path length between any two nodes in a unimodal network G=(V,E). If there exist i,j in V such that j is not reachable from i, then |V| is returned.	Graph	Communication	Single-cell
Distance Weighted Reach	A generalization of graph theoretic distance, this measures the distance from a *set* of nodes in the network to all other nodes.	Graph	Communication, Critical Employee	Single-cell
Diversity, Knowledge	The distribution of difference in idea sharing. This is the Herfindahl-Hirshman index applied to column sums of AK.	Graph	Resource Allocation	Single-cell
Diversity, Resource	The distribution of difference in resource sharing. This is the Herfindahl-Hirshman index applied to column sums of AR.	Graph	Resource Allocation	Single-cell
Edge Count, Lateral	The percentage of lateral edges in a unimodal network. 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.	Graph	Communication	Single-cell
Edge Count, Pooled	The percentage of pooled edges in a unimodal network. A pooled is an edge (i,j) such that there exists at least one other edge (i,k) in the network, and $k \neq j$.	Graph	Task	Single-cell
Edge Count, Reciprocal	The percentage of edges in a unimodal network that are reciprocated (also called Reciprocity). An edge (i,j) in the network is reciprocated if edge (j,i) is also in the network.	Graph	Task	Single-cell
Edge Count, Sequential	The percentage of edges in a unimodal network that are neither Reciprocal Edges nor Pooled Edges. Note that an edge can be both a Pooled and a Reciprocal edge.	Graph	Task	Single-cell

Edge Count, Skip	The fraction of edges in a unimodal network that skip levels. An edge (i,j) is a skip edge if there is a path from node i to node j even after the edge (i,j) is removed.	Graph	Communication	Single-cell
Effective Network Size	The effective size of a node's ego network based on redundancy of ties.	Node	Critical Employee, Redundancy, Communication	Single-cell
Exclusivity, Knowledge	Detects agents who have singular knowledge.	Node	Critical Employee	Single-cell
Exclusivity, Resource	Detects agents who have singular resource access.	Node	Critical Employee	Single-cell
Exclusivity, Task	Detects agents who exclusively perform tasks.	Node	Critical Employee Performance	Single-cell
Fragmentation	The proportion of nodes in a network that are disconnected	Graph	Critical Employee, Communication , Personnel Interaction	Single-cell
Hierarchy	The degree to which a unimodal network exhibits a pure hierarchical structure.	Graph	Communication	Single-cell
Interdependence	The percentage of edges in a unimodal network that are Pooled or Reciprocal.	Graph	Task	Single-cell
Interlockers and Radials	Interlocker and radial nodes in a square network have a high and low Triad Count, respectively.	Graph	Critical Employee	Single-cell
Load, Knowledge	Average number of knowledge per agent.	Graph	Resource Allocation, Redundancy	Single-cell
Load, Resource	Average number of resources per agent.	Graph	Resource Allocation, Redundancy	Single-cell
Negotiation, Knowledge	The extent to which agents need to negotiate with each other because they lack the knowledge to complete their assigned tasks.	Graph	Resource Allocation, Task	Multi-cell
Negotiation, Resource	The extent to which agents need to negotiate with each other because they lack the resources to complete their assigned tasks.	Graph	Resource Allocation, Task	Multi-cell
Network Centralization, Betweenness	Network centralization based on the betweenness score for each node in a square network. This measure is defined for directed and undirected networks.	Graph	Communication	Single-cell
Network Centralization, Closeness	Network centralization based on the closeness centrality of each node in a square network. This is defined only for connected, undirected networks.	Graph	Communication	Single-cell

Network Centralization, Column Degree	A centralization based on the degree of the column nodes of a network.	Graph		Single-cell
Network Centralization, In Degree	A centralization of a square network based on the In-Degree Centrality of each node.	Graph	Communication	Single-cell
Network Centralization, Out Degree	A centralization of a square network based on the Out-Degree Centrality of each node.	Graph	Communication	Single-cell
Network Centralization, Row Degree	A centralization based on the degree of the row nodes in a network.	Graph		Single-cell
Network Centralization, Total Degree	A centralization of a square network based on total degree centrality of each node.	Graph	Communication	Single-cell
Network Levels	The Network Level of a square network is the maximum Node Level of its nodes.	Graph	Communication	Single-cell
Node Level	The Node Level for a node v in a square network is the longest shortest path from v to every node v can reach. If v cannot reach any node, then its level is 0.	Graph	Communication	Single-cell
Omega, Knowledge	The degree to which agents reuse knowledge while doing their tasks.	Graph	Task, Performance	Multi-cell
Omega, Resource	The degree to which agents reuse resources while doing their tasks.	Graph	Task, Performance	Multi-cell
Performance as Accuracy	Measures how accurately agents can perform their assigned tasks based on their access to knowledge and resources.	Graph	Task, Performance	Multi-cell
Personnel Cost	Total number of people reporting to an agent, plus its total knowledge, resources, and tasks.	Node		Multi-cell
Potential Workload, Knowledge	Maximum knowledge an agent could use to do tasks if it were assigned to all tasks.	Node	Resource Allocation	Multi-cell
Potential Workload, Resource	Maximum resources an agent could use to do tasks if it were assigned to all tasks.	Node	Resource Allocation	Multi-cell
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.	Graph	Resource Allocation, Redundancy	Multi-cell
Redundancy, Assignment	Average number of redundant agents assigned to tasks. An agent is redundant if there is already an agent assigned to the task.	Graph	Resource Allocation, Redundancy	Multi-cell
Redundancy, Column	The mean number of column node edges in excess of one.	Graph		Single-cell

Redundancy, Knowledge	Average number of redundant agents per knowledge. An agent is redundant if there is already an agent that has the knowledge.	Graph	Resource Allocation, Redundancy, Task	Single-cell
Redundancy, Resource	Average number of redundant resources assigned to tasks. A resource is redundant if there is already a resource assigned to the task.	Graph	Resource Allocation, Redundancy, Task	Single-cell
Redundancy, Row	The mean number of row node edges in excess of one.	Graph		Single-cell
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.	Node	Personnel Interaction	Single-cell
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.	Node	Personnel Interaction	Single-cell
Span of Control	The average number of out edges per node with non-zero out degrees.	Graph	Communication	Single-cell
Speed, Average	The average shortest path length between node pairs (i,j) where there is a path in the network from i to j. If there are no such pairs, then Average Speed is zero.	Graph	Communication	Single-cell
Speed, Minimum	The maximum shortest path length between node pairs (i,j) where there is a path in the network from i to j. If there are no such pairs, then Minimum Speed is zero.	Graph	Communication	Single-cell
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.	Graph	Performance	Multi-cell
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.	Graph	Performance	Multi-cell
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.	Graph	Performance	Multi-cell

Transitivity	The percentage of edge pairs {(i,j), (j,k)} in the network such that (i,k) is also an edge in the network.	Graph	Communication, Task	Single-cell
Triad Count	The number of triads centered at each node in a square network.	Node	Communication	Single-cell
Under Supply, Knowledge	The extent to which the knowledge needed to do tasks are unavailable in the entire organization.	Graph	Resource Allocation, Task	Multi-cell
Under Supply, Resource	The extent to which the resources needed to do tasks are unavailable in the entire organization.	Graph	Resource Allocation, Task	Multi-cell
Upper Boundedness	The degree to which pairs of agents have a common ancestor.	Graph	Communication	Single-cell
Weak Boundary Spanner	A node which if removed from a network creates a new component.	Node	Critical Employee	Single-cell