



An Automated Methodology for Conducting a Social Network Study of a University Faculty^{*}

CASOS Technical Report

Terrill L. Frantz, Kathleen M. Carley, Jana Diesner^{**}

February 2005

CMU-ISRI-05-106

Carnegie Mellon University

School of Computer Science

ISRI - Institute for Software Research International

CASOS - Center for Computational Analysis of Social and Organizational Systems

Abstract

This technical report describes the data-collection and analysis methodology used in a social-network study of computer science faculty recently conducted at a top U.S. university. The study involved the construction of a social metanetwork that combines multiple dimensions of the faculty's knowledge, task, and collaborative networks. The process of collecting the source-data, then messaging the data for analysis was completely automated using custom computer software. This allowed for an entirely nonobtrusive data collection process and simplified and repeatable analytic tasks. In this report: we first introduce the metamatrix framework and discuss its utility as a tool for complex network-analysis; next, we describe the data collection methodology; then, we explain metamatrix construction process; and finally we present some closing remarks. This report does not discuss specific analysis or findings of the underlying research, which are regarded as being confidential.

^{*} This work was supported by the Army Research Lab and the Office of Naval Research. 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 Army Research Lab, the Office of Naval Research, or the U.S. government.

^{**} The authors extend a special thanks to Corinne A. Coen, Assistant Professor, Organization and Human Resources Dept., at the University at Buffalo, The State University of New York, for her intellectual contributions to this research project.

Keywords: social network analysis, data collection, automation, collaboration network, knowledge network, metamatrix, metanetwork

Table of contents

I.	Index of Tables	iv
II.	Index of Figures	iv
1.	Motivation.....	5
2.	Introduction.....	5
3.	Metamatrix	5
4.	Data Collection Methodology.....	6
4.1	Identifying the Faculty Population.....	7
4.2	Identification of Task Assignments	7
4.3	Identifying Author Collaboration	8
5.	Metamatrix Construction	9
5.1	Actor Components	9
5.2	Task Components.....	10
5.3	Knowledge Components.....	10
5.4	Actor-Actor Subnetworks	10
5.5	Actor-Task Subnetworks	11
5.6	Actor-Knowledge Subnetworks.....	11
5.7	Task-Knowledge Subnetworks	11
5.8	Knowledge-Knowledge Subnetworks	12
6.	Metamatrix Descriptive Statistics	12
7.	Closing Remarks	15
8.	References.....	16

I. Index of Tables

Table 1. Submatrix Combinations in a Metamatrix.....	6
Table 2. Descriptive Measures of Individual Networks in the Metamatrix.....	12

II. Index of Figures

Figure 1. Histograms of Degree Counts (Unimode Subnetworks).....	13
Figure 2. Histograms of Degree Counts (Bimodal Subnetworks).....	14

1. Motivation

The research lab of the Center for Computational Analysis of Social and Organizational Systems (CASOS) was commissioned to conduct a social-network analysis of a department at a top university located in the United States. The study focused on the members of the research and teaching faculty making up the computer-science department. The purpose of this technical report is to document the study from the perspective of its data-collection methodology and to describe the ensuing social metamatrix dataset. This report does not present analysis or findings pertaining to the study. If used, actual names of subjects are masked in this report, as well as in the publicly-available data; the complete dataset is available for academic research from the CASOS Lab.

2. Introduction

We recently completed a social network study that mapped the knowledge, task and collaboration networks of a faculty of a university department. The study not only provided invaluable information to the department administration and individual faculty members, but it also demonstrated the successful application of an automated and nonobtrusive data-collection methodology. The methodology proved to be low-cost, yet effective, and is a repeatable process that may be applied to other studies with similar data-collection and analysis requirements. In addition, this study has yielded a real-life social-network dataset useful for other—albeit generic—social-network research.

An essential step in conducting this study involved the collecting and analyzing of data pertaining to a subject population of faculty members. The source data was collected from secondary sources without the direct involvement of the faculty members. After the data-collection step and some indispensable processing by computer software, a complete metamatrix was constructed from the source data. This metamatrix mapped the multidimensional relationships between faculty, their tasks and their knowledge. After the metamatrix was constructed, detailed reports were created for both the department's administration and for each individual faculty member. The individual reports were in the form of personalized egocentric network graphics and descriptive statistical tables.

For the remainder of this report: we first introduce the metamatrix framework and discuss its utility as a tool for complex network-analysis; next, we describe the data collection methodology; then, we explain metamatrix construction process; and finally we present some closing remarks.

3. Metamatrix

A metamatrix is a framework that integrates multiple and related network matrices into a single interrelated unit (Carley, 2002). Typically, a metamatrix represents a collection of graphs for a single organization or group. The matrices making up the metamatrix can be either single mode or bimodal, i.e., made up of same-type nodes or of two different types. This flexibility allows for sophisticated and holistic social network analysis beyond the standard single-node-type and single matrix analysis most commonly conducted. In particular, revealing multi-matrix statistics can be calculated by combining measures across matrices.

The metamatrix constructed for this study utilizes a portion of the PCANS (Krackhardt and Carley, 1998) structure of organization. It also closely resembles the metamatrix conceptualized in Carley's (2002) introductory paper. The primary components making up the multiple networks represented are: a collaboration (social) network, a task network, and a knowledge

network. From these three components, up to six combinations of different sub-network types can be constructed as shown in Table 1: (a) Actor-Actor, (b) Actor-Task, (c) Actor-Knowledge, (d) Task-Task, (e) Task-Knowledge, and (f) Knowledge-Knowledge. While all are important perspectives on the organization's social network, only a subset of these were actually constructed for this study. The subset chosen is derived from the specific research questions asked by the clients of this study

Table 1. Submatrix Combinations in a Metamatrix.

	Actors	Tasks	Knowledge
Actors	AA	AT	AK
Tasks		TT	TK
Knowledge			KK

The metamatrix data is represented by the DyNetML format. DyNetML (Tsvetovat, Reminga, & Carley, 2004) is a computer file format that is increasingly being used for complex social-network studies. DyNetML, as its name suggests, is an XML-based format that has expanded data features beyond other standard network-related data formats.

4. Data Collection Methodology

The source data was obtained from three sources: (a) the department's faculty-directory webpage, (b) various lists provided by the department administrator, and (c) a publicly available online scientific-paper repository. From these sources, three metamatrix component matrices were constructed: (a) actors, (b) tasks, and (c) knowledge. The source data collection process is further explained in the remainder of this section and the particulars for the construction of the metamatrix are presented in Section 5.0.

The process of collecting data in this study was fully automated and was nonobtrusive to the subjects. Custom, yet reusable, software was developed specifically to enable the automation of handling data for the entire process from the original source data reformatting to the creation of reports. Relationship was obtained from electronic sources without the subjects' direct involvement; the department administrator provided electronic source-data and we utilized an electronic scientific-paper database.

To normalize the various source-data formats for faculty names, project titles, grant names, and student advisory assignments into a common format, customized, but generalized, software was developed to make the automated conversion. The software was written in Perl. The use of Perl not only simplified the programming task, but it also allowed for external software to be easily executed under its scripting control—thus allowing the entire data-handling and report-creation process to be fully automated and executable in a simple and hands-off manner. While Perl was selected for this application, any number of other computer languages certainly would have sufficed.

In order to collect source data without the subjects' direct involvement, the department administrator provided project, grant and student-advisor activity lists along with a master listing identifying exactly which members of the faculty they considered to be members of the department (since faculty often are members of multiple departments, schools, projects, etc.). The paper-authoring data were collected from a web-based database using Perl programs designed to scrape the web pages.

4.1 Identifying the Faculty Population

The faculty directory listing, which is publicly available on the department's website, was used to determine the proper names of the faculty members. Accurate spelling is critical in order to accurately search any scientific paper database accurately. Specific to this study, the administration provided a separate list of faculty names that delineated which faculty was of interest. The list supplied by the administration was used as the arbiter to exclude a few names that were on the web-based directory, which may suffer from update lag-time.

The names were then each normalized and encoded for simplified and consistent labelling of the network actor components, a.k.a., faculty. Several names required special exception handling in the conversion of informal names, like *Bob* to the formal version, in this case, *Robert*. This name mapping was facilitated by a data file used by the software to make the name conversions. Researcher intervention was necessary during this step.

As a result of the name collection and labelling process, 89 unique faculty names were identified. Starting from the faculty directory listing and in conjunction with the official list provided by the department administration, the group faculty names delineated the strict actor boundary for the study.

Each name was encoded into a unique all-capital-letter identifier forming a consistent label format across faculty members. The name label was formed from the full last name, an underscore character, and the first letter of their first name, e.g., *John Doe* becomes *DOE_J*. In several instances, an individual has one or more name-aliases, namely different first names, e.g., *Elizabeth* and *Betsy*, or *Robert* and *Bob*. Surnames (last names), like a woman's maiden or a non-western-culture name can also occur. In this study, whether or not a woman was using her maiden name or not, was not indicated to the researchers; however, there were two cases of nonwestern-culture names for which adjustments were made. A computer-readable table of these aliases was coded and maintained accordingly. The table mapped any name, alias or original, with the all-capital-letter identifier.

Any instance of a name entering the domain of the study was first encoded and mapped, via the name-identifier table, into the unique name identifier. This mapping process and resulting name map has particular relevance and importance in the automated process of collecting large amounts of data. Names are particularly difficult to search accurately when outside the confines of the more name-normalized domain of the university records.

4.2 Identification of Task Assignments

The department's administration provided four lists that served as the source data artefacts for this study. The four were: (a) faculty names (discussed above), (b) projects and affiliated faculty, (c) grants and affiliated faculty, and (d) faculty advisors to students. These lists which were originally provided to the researchers formatted in basic spreadsheet form, were converted into an easier-to-manipulate text-based form using the custom Perl software.

The details of the formats for the original lists provided is not relevant and will not be discussed in detail here, but it should be noted that the conversion of a faculty member's name to his or her key i.d. required more than simplistic text-reformatting techniques. This conversion process is discussed in the previous section. While some of the supplied names had spelling mistakes, some faulty names in the lists were not of interest. There were 131 unique project names, 115 unique grant names, and 132 individual students identified in these three lists.

In each list, from one to many faculty names could be associated with a given item in the list. Frequently, faculty collaborate in pairs or teams on a project, grant, or as advisers to students.

Any faculty name found on these lists that was not in the list of 89 subject names was removed and discarded from the study. Any project, grant, or student without at least one of the 89 faculty associated with it was removed from the data.

4.3 Identifying Author Collaboration

The source used to identify scientific papers written by the faculty was the CiteSeer Scientific Literature Digital Library. CiteSeer is a web-based self-archiving database of papers written by the computer-science community. While CiteSeer provides an invaluable service to the research community, unfortunately, the database contains duplicate entries, spelling errors, and other imperfections that make automated access (via software) prone to complications and error. Every effort was made to accommodate and work with or around the known data issues through the customized software. Nevertheless, using CiteSeer as a source of data resulted in a multitude of source data errors and incomplete data, which must be considered in any final analysis. While, these source-data imperfections can result in significantly compromised data, the researchers concluded that for the most part, the data indeed remained pertinent and valuable as a resource based on the intended objectives of this study.

Using each of the 89 faculty names and several possible combinations of each individual's name, a CiteSeer search-by-author was executed for each possibility. This process, performed by the Perl software, resulted in an article index listing that was saved for each faculty name. The collection of these sets of possible articles was first combined into a single list of interest. Next, using the combined list, duplicate entries were removed by comparing each article's web address, i.e., the URL, as the authoritative indicator of a duplicate entry. It was found that in some cases, coauthors had each posted the same article to CiteSeer separately, creating a situation in which the same article had different URLs. A software process to remove duplicate titles caught this duplication, except in the case of different misspellings of the title.

The list of the CiteSeer article URLs was applied systematically to the internet with the resulting title and author information being collected in HTML format. The articles identified were screened to make sure the subject was the author (as opposed to being a referring article, or such). For each article, the abstract text was then captured if it was available. Author names were compared to the articles collected: Any names identified as authors, but not on the list, were removed; any article without a name was removed. This process resulted in a list of only those articles with one or more authors who were in our faculty list.

Constructing the knowledge concepts and relationships for the metamatrix was a complex aspect of the data processing. The relevant text was analyzed using AutoMap (Diesner & Carley, 2004) version 1.2, which encodes links between words in a text and constructs an associative relationship among the words. Thanks to the AutoMap software, the first step of the concept mapping-process was easy. AutoMap applied advanced textual analysis techniques to the full set of text-based data collected in this study. Project and grant titles, as well as article titles and associated abstract text, were passed as input to AutoMap.

From the AutoMap process, text files with concepts and relations among concepts were created. These data files became input to the customized Perl script software, which then mapped the output in the internal program format. AutoMap was found to be effective in the process, accurately identifying the text from which it gleaned relevant concepts based on a computer-science lexicon. This process resulted in knowledge being represented in the form of words and word combinations which could be appropriately joined to form a concept.

5. Metamatrix Construction

Construction of the metamatrix involved the following steps: (a) filtering data from these sources for data applicable only to the faculty of interest, (b) combining the pieces of data into a consolidated repository, and (c) processing the textual information through text-analysis software.

The metamatrix data set constructed for this study was the result of the Perl software taking the input data (project, grant and student lists; department web site; CiteSeer database) and reformatting it into a formatted data file. The entire process took approximately one minute to execute on a personal computer yielded much more than the single metamatrix data file. While several other data files in various data formats, which greatly facilitated the automated process and flexibility enjoyed by this study were created, they will not be discussed here. These files served as input to analysis and visualization software used in this other aspects of the study.

For the purpose of this study, several Actor-Actor (AA) matrices were constructed from the data since the actor relations to one another were the primary focus of the study. As necessity dictated, an Actor-Task (AT) and Actor-Knowledge (AK) matrix were constructed from the data sources. The Task-Task (TT) matrix was not constructed since it was workflow that was of interest; however the Task-Knowledge (TK) matrix which provided information about the relationship between the project and grants and the knowledge concepts required for these particular tasks, was formed. A Knowledge-Knowledge (KK) matrix, which provided a conceptual representation of the knowledge in the department, was also formed. The KK matrix was constructed from the output of the AutoMap text analysis software.

The metamatrix, which followed all associated specifications for formatting and presentation was encapsulated and represented in the form of a DyNetML-formatted (Tsvetovat, Reminga, & Carley, 2004) XML file. This made the faculty metamatrix easily usable in the growing number of social network statistical and simulation tools which incorporate DyNetML.

The DyNetML format allows for multiple node types, and multiple relationship sets to be captured into a single computer file, making data management easier than it would be if a directory full of computer files had to be maintained and allowing for multi-matrix statistics to be easily calculated.

As specified by the DyNetML format, nodes were identified under *nodeset* XML elements and the relations (the adjacency matrices) were identified under the *graph* XML element. The specific nodeset and graph trees collectively making up the metamatrix constructed in this study are discussed below.

5.1 Actor Components

In a simple metamatrix, there can be only one actor nodeset. The actor nodeset identifies the population of actors referenced in any of the subnetworks contained in the specific metamatrix dataset. For this study, the set of actor nodes corresponded to the group of faculty members. Each distinct node in the actor node set represented an individual faculty member. Beyond the required node label used for identification of the node—in this case, the short faculty name identifier—no additional attribute of the actor was shown in the actor nodeset.

Corresponding directly with the number of faculty in this study, there were 89 actor nodes in the nodeset. The labels were derived from the actor's last name with an underscore and first initial added to the end of the label. For the purposes of publication of the related research and later use of the metamatrix, the names have been masked. The actor node labels are all upper case.

5.2 Task Components

Like the actor node set, in a simple metamatrix, there can be only one task nodeset. The task nodeset identifies the set of tasks referenced in any of the subnetworks contained in the specific metamatrix dataset. This study combined the lists of projects, grants and student-advisor assignments into a single list of tasks.

For these tasks in this study, later analysis required that the type of task (project, grant or advisory) be available. Rather than increasing the complexity of the metamatrix data set, it was decided to simply embed the task type in the task name label. To keep the task type with the label name, the label text was prefixed with a *p*, *g*, or *s*—for project, grant, and student advisory, respectively. A total of 378 task nodes consisting of 131 unique project names, 115 unique grant names, and 132 individual student names are contained in this metamatrix;

5.3 Knowledge Components

As with other types of nodesets, in a simple metamatrix, there can be only one knowledge nodeset. The knowledge nodeset identifies the set of concepts, the term *concept* referring to an individual classification of an element of some relevant knowledge, which in this case would be a term or phrase used in the computer science community. The knowledge nodeset identifies all knowledge concepts possible in the metamatrix. Only their identifying label (the term or phrase) is contained in the information. There were 114 concepts in the knowledge nodeset for this study.

The selection of what constitutes a concept was a subjective decision made by the researchers. While the introduction of a specific term was driven by the term's appearance in the text processed in AutoMap, the choice of keeping the term in the data or possibly combining terms into a conjunction was at the discretion of the researchers. In this study, the experience of the researchers in computer science—congruent with the faculty's being in the computer science community—and their familiarity with that science's special terms made it reasonable for them to make this type of judgement.

5.4 Actor-Actor Subnetworks

Seven Actor-Actor subnetworks (AA) were created from various configurations of the actor, task, and knowledge nodes and the underlying relationship ties among the nodes. The AAs created for the study were: (a) AA Project, (b) AA Grant, (c) AA Student, (d) AA Three Exist, (e) AA Article, (f) AA Total Collaboration, and (g) AA Knowledge. Some were formed via a simple pairing of actor-entity and entity-actor sets of relations, while others were slightly more complicated.

The *AA Project* subnetwork involved creating a network tie between two faculty members when both were assigned to at least one of the same projects. The ties were weighted according to the number of projects the two faculty members have in common.

The *AA Grant* subnetwork involved creating a network tie between two faculty members when they both were assigned to at least one of the same grants. The ties were weighted according to the number of grants the two faculty members have in common.

The *AA Student* subnetwork involved creating a network tie between two faculty members when they both were assigned to advise at least one of the same students. The ties were weighted according to the number of students two faculty members were assigned in common.

The *AA Three Exist* subnetwork involved creating a network tie between two faculty members when both were assigned to at least one of the same projects, grants, or students. The

ties were weighted with a value of 1, 2, or 3, according to the number of times a tie existed between the two faculty members in a combined network of AA Project, AA Grant, and AA Student.

The *AA Article* subnetwork involved creating a network tie between two faculty members when they both were found to have been coauthors on the same scientific paper. The ties were weighted according to the number of papers the two faculty members had coauthored with each other.

The *AA Total Collaboration* subnetwork involved creating a network tie between two faculty members when both had collaborated on a project, grant, student advisory, or coauthored the same paper. The ties were weighted according to the number of instances of collaboration between the two faculty members.

The *AA Knowledge* subnetwork involved creating a network tie between two faculty members when both were tied to the same knowledge nodes. The ties were weighted according to the number of times the two faculty members were both tied to the same knowledge concept.

5.5 Actor-Task Subnetworks

An actor-task subnetwork (AT) is a construct that identifies ties between faculty members, e.g., actors, and the tasks to which they are assigned. The subnetwork i.d. for the AT subnetwork is "AT Faculty x Proj/Grant/Advise." This was a bimodal graph, there were no ties in this network between like nodes. In particular, there were no ties either between the two tasks, or the two actors. The task entities were organized from the conjunction of the distinct sets of faculty projects, university grants and graduate-student advisor responsibilities. The labels identifying the task nodes were prefixed with an identification of the task type—*p*, *g*, or *s*, representing *project*, *grant*, or *student advisory*, respectively. The weight for each tie was set to 1.

5.6 Actor-Knowledge Subnetworks

An actor-knowledge subnetwork (AK) is a construct that identifies the ties between faculty members, e.g., actors, and the knowledge concepts to which they have been affiliated. The subnetwork i.d. for this subnetwork was "AK Faculty x Knowledge." Since this was a bimodal graph, there were no ties in this network between *like* nodes. In particular, there are no ties between two concepts or between two actors. The weight for each tie will always be the value 1.

The concept associations were drawn from the AT matrix and the associated task labels (project names and grant titles), paper titles and their abbreviated abstracts collected then processed through the AutoMap software. This process is explained with greater depth in section 5.3.

5.7 Task-Knowledge Subnetworks

A task-knowledge subnetwork (TK) is a construct that ties task nodes to knowledge concepts. The subnetwork i.d. for this subnetwork was "TK Proj/Grant x Knowledge." The task entities were organized from the conjunction of the distinct sets of faculty projects, and university grants (Graduate-student advisor responsibilities were not included) and the knowledge concepts. This association between task and knowledge was initially made by the AutoMap text analysis software, then further linked via the custom Perl software written for this study. Since TK was a bimodal graph, there were no ties in this network between like nodes. In particular, there were no ties between two tasks or between two concepts. The weight for each tie was set to 1.

5.8 Knowledge-Knowledge Subnetworks

A knowledge-knowledge subnetwork (KK) is a construct that ties task nodes to knowledge concepts. The subnetwork i.d. for this subnetwork was "KK Knowledge x Knowledge." The relationship-ties between concepts were determined by AutoMap software and involved linking specific phrases according to where they appear. If they appeared together in the same project title, grant name, paper (title or abstract) they were deemed to be associated terms and a tie was included in the sub-network. The weight for each tie pertains to the number of times the concepts appeared together in a project title, grant name, or paper.

6. Metamatrix Descriptive Statistics

The data collection and later processing resulted in the construction of a metamatrix. The metamatrix consolidates several perspectives of the faculty social network into a single dataset. Included in the metamatrix, for this study, are relationship ties among actors, tasks and knowledge concepts.

There are 89 actors, 378 tasks—made up of 131 projects, 115 grants, and 132 students—and 114 knowledge concepts in the metamatrix. Table 2 shows a survey of the number of edges contained in each of the subnetworks making up the metamatrix dataset. Notice that three of the subnetworks are bimodal and do not have a square adjacency-matrix.

Table 2. Descriptive Measures of Individual Networks in the Metamatrix.

Subnetwork Id	Adjacency Matrix Dimension	Number of Edges	Graph Density
AA Total Collaboration	89 x 89	296	.038
AA Project	89 x 89	232	.030
AA Grant	89 x 89	187	.024
AA Article	89 x 89	132	.017
AA Student	89 x 89	40	.055
AA Knowledge	89 x 89	4,750	.606
AA Three Exist	89 x 89	368	.047
AT Faculty x Proj/Grant/Advise	89 x 378	479	.014
AK Faculty x Knowledge	89 x 114	912	.090
TK Proj/Grant x Knowledge	378 x 114	171	.004
KK Knowledge x Knowledge	114 x 114	4,386	.340

A benefit of storing the metamatrix data in DyNetML format, thus being in standard XML, is that the details of the data file format are evident simply by looking at the data file itself. Since it is unnecessary to discuss the data file format, it will not be described here.

Figures 1 and 2 provide a graphical summary of the degree distributions for the nodes in each of the subnetworks contained in the metamatrix. This information is included to provide a sense of the data and the metanetwork. Any statistical measure of interest can be calculated using the ORA network-statistics software (Carley & Reminga, 2004). The faculty metamatrix data set available from CASOS can quickly be run as input into ORA for a full complement of measures, including multiple matrix statistics.

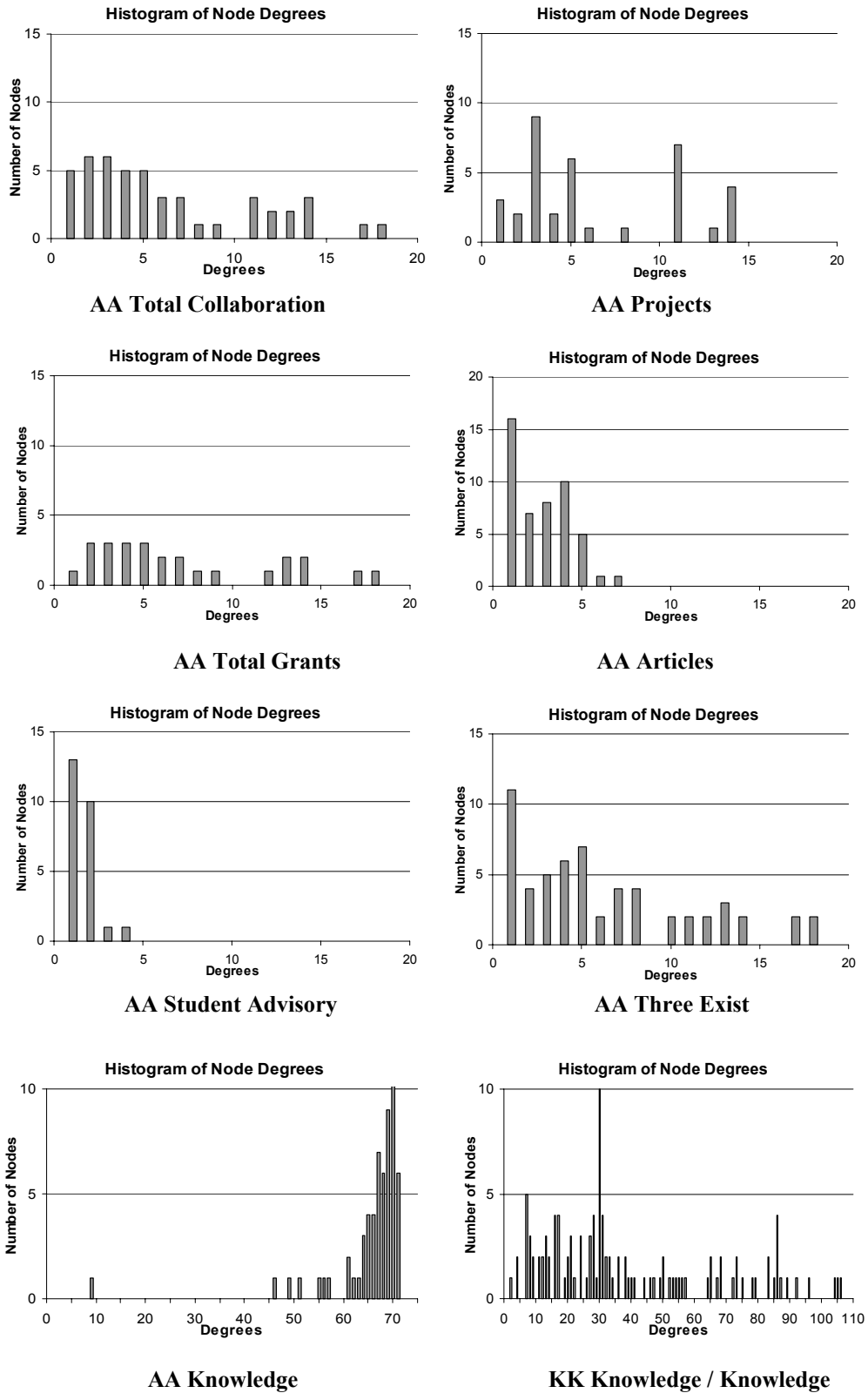


Figure 1. Histograms of Degree Counts (Unimode Subnetworks)

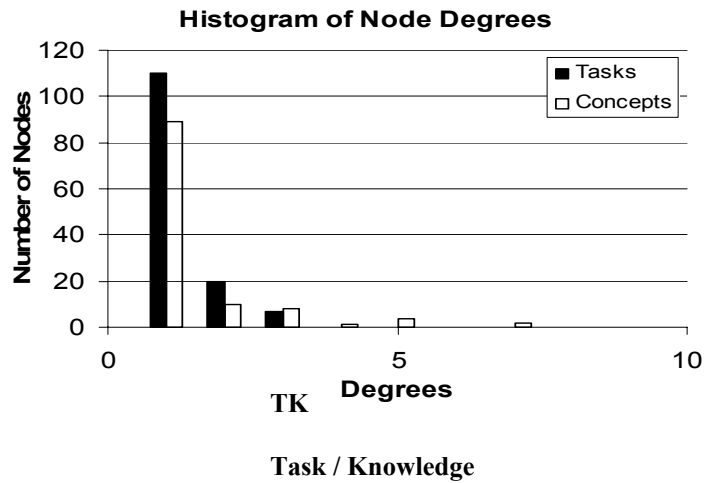
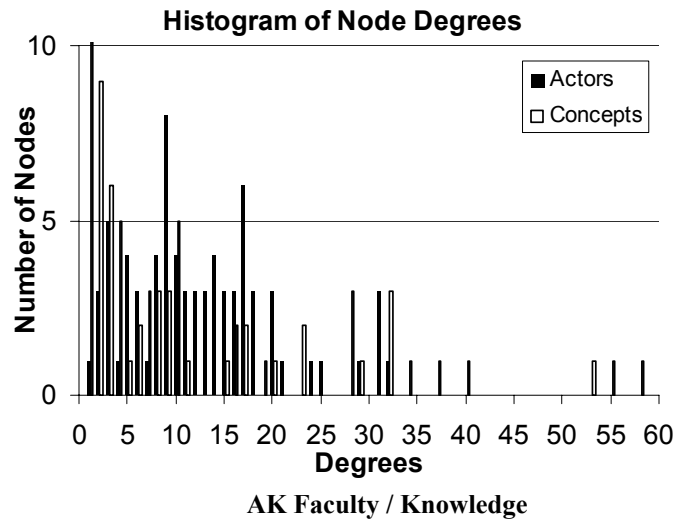
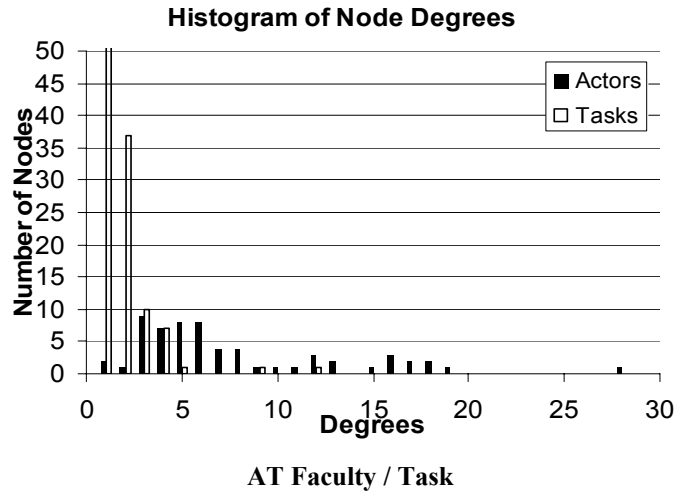


Figure 2. Histograms of Degree Counts (Bimodal Subnetworks)

7. Closing Remarks

The successful completion of the underlying study using the methods we describe in this report demonstrates that complex social-network analysis can be undertaken in an automated fashion at low-cost to researchers. The raw-data processing and reporting process can be, to a great extent, automated thus reducing time and cost for these indispensable steps. We have also demonstrated that direct involvement by study subjects is not necessary to conduct an introspective and informative analysis of a multifaceted and complex group structure—in this case of the faculty of a university.

Automation of these steps also implies that much of the research process is easily re-executable. This not only allows for multiple experiments with different analytic approaches to be conducted with ease, but also allows for entirely separate studies, pertaining to other subjects, to be conducted effortlessly. Studies of faculty groups in other departments, using this same process, are planned by the research lab at CASOS—ultimately demonstrating the absolute reusability of the process and software. For future applications, only the original source-data format needs to be reconciled with the format requirements of this software, then a complete analysis with a full set of reports can then be generated for the data in a matter of minutes.

By applying automated and nonobtrusive methodology as presented in this technical report, complex multinetwork social analysis is possible for even the most resource-restricted researchers.

8. References

Carley, Kathleen M. (2002). Inhibiting adaptation. In *Proceedings of the 2002 Command and Control Research and Technology Symposium*. Conference held in Navel Postgraduate School, Monterey, CA. Evidence Based Research, Vienna, VA.

Carley, Kathleen, M., & Reminga, Jeff. (2004). *ORA: Organization risk analyzer*. Carnegie Mellon University Technical Report, CMU-ISRI-04-106

Diesner, Jana, & Carley, Kathleen M. (2004). *AutoMap1.2 - Extract, analyze, Represent, and compute mental models from texts*. Carnegie Mellon University Technical Report, CMU-ISRI-04-100

Krackhardt, David., & Carley, Kathleen M. (1998). *A PCANS model of structure in organization*. Paper presented at the Proceedings of the 1998 International Symposium on Command and Control Research and Technology (June 1998), Monterrey, CA.

Tsvetovat, Maksim, Reminga, Jeff, & Carley, Kathleen M. (2004). *DyNetML: Interchange format for rich social network data*. Carnegie Mellon University Technical Report, CMU-ISRI-04-105.