

System for the Online Analysis of Distributed Projects

by

Daniel A. Nunes

Submitted to the Department of Electrical Engineering and Computer Science

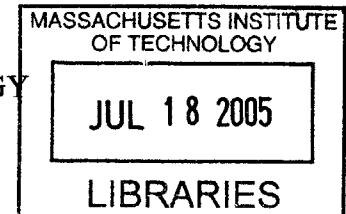
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 19, 2005



Copyright © MMV, Daniel A. Nunes. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and distribute publicly paper and electronic copies of this thesis and to grant others the right to do so.

Author

Electrical Engineering and Computer Science
May 19, 2005

Certified by _____

mings
rvisor

Accepted by _____

Smith
Chairman, Department Committee on Graduate Theses

BARKER

System for the Online Analysis of Distributed Projects

by

Daniel A. Nunes

Submitted to the
Department of Electrical Engineering and Computer Science

May 19, 2005

In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

With global collaboration becoming increasingly common among researchers, project managers are seeking to reduce the negative impact of distance on team performance. There is an increasing demand for an integrated tool to assess and analyze the properties of work groups in order to identify key features which could either facilitate or hinder performance. The described system creates a robust interface for researchers and project managers to calculate statistics, analyze, visualize, and compare assessment data in order to gain insight into the interaction networks of work groups.

Thesis Supervisor: Jonathon Cummings

Title: Assistant Professor, Sloan School of Management

Acknowledgments

I would like to thank Professor Cummings for his persistent guidance and support throughout the past year. He has always been there to provide feedback and direction in my work, taking extra time to make sure I understood the “bigger picture” of my contributions.

I am also indebted to my colleague, Maria Garcia, who has helped me in countless sessions of brainstorming through problems and solutions. I owe special thanks to my parents, Stan and Bea, who have provided never-ending support and encouragement throughout my education, and my fiancée, Meg, for putting up with endless hours of work and the stress that accompanies it.

Contents

1.	Introduction.....	7
1.1.	Motivation.....	7
1.2.	Current Methods	8
1.3.	Design Goals.....	8
1.4.	System Overview	9
2.	System Design and Development	12
2.1.	Technologies	12
2.1.1.	Apache, PHP, MySQL, HTML, and JavaScript	12
2.1.2.	JpGraph.....	13
2.1.3.	NetVis	14
2.2.	Development Strategies	14
2.3.	User Interface.....	15
2.4.	Issues.....	17
2.4.1.	Data Format	17
2.4.2.	Data Synchronization.....	22
2.5.	Maintenance	23
3.	Features.....	24
3.1.	Statistical Measures	24
3.2.	Network Analysis.....	28
3.2.1.	Visualization	28
3.2.2.	Network measures.....	29
3.3.	Administration	30

3.3.1.	Data Synchronization.....	31
3.3.2.	Exporting Data.....	32
3.3.3.	Variable Setup.....	33
4.	Conclusion	36
4.1.	Lessons Learned.....	36
4.2.	Future Work	37
4.2.1.	User Management	38
4.2.2.	Multiple Regression	38
4.2.3.	Variable Setup.....	38
4.2.4.	Saving Analyses.....	39
4.3.	Research Implications.....	39
5.	Bibliography	41
Appendix A	Integrating non-OPAS Data.....	42

1. Introduction

1.1. Motivation

Distributed innovation, the collaboration of individuals from geographically dispersed locations to generate new ideas, has become a focus of organizations seeking to improve their product development [1]. With tools for collaboration becoming increasingly abundant and accessible, companies and research groups with locations around the world are searching for ways to examine their existing network interaction infrastructures and develop methods to streamline their communication and collaboration.

The demand for an online project analysis system is made clear through the popularity of the existing assessment system developed within MIT's Initiative for Distributed Innovation [2] that provides an online survey interface capable of collecting project and relational data [3]. Several large and small-scale companies and research programs have been participating in the assessment, and the number of requests to set up new assessments has been rising steadily. Increased participation creates a valuable and reliable source of data which can be used in aggregate to infer the key components of a successful geographically dispersed project team.

The primary user of the analysis system is the Cambridge-MIT Institute (CMI), though other organizations use the system, as well. CMI runs semiannual assessments collect information about projects funded by CMI and are used to observe the effects of project features, geographical distance, and network interaction on performance [4]. CMI is the primary audience for the analysis system, though it is designed to be applied toward a wide variety of assessments.

1.2. Current Methods

The Online Project Assessment System (OPAS) was developed in Spring 2004 to allow researchers and other project team members log in and answer questions about features of their projects and their interaction with other team members [3]. The assessment format is customizable across different organizations and can be configured to ask different questions to members of the same organization based on their project role. All of the collected data is stored in a database, leaving analysis to be performed manually with the use of external statistical analysis software. There is strong demand for an automated tool to analyze and present the collected survey data directly to the system users and project managers, without the need for human intervention. This would greatly facilitate the process, helping meet the demand for more organizations to participate in the assessment.

1.3. Design Goals

The System for the Analysis of Projects Online (SAPO) has four primary objectives. One objective is that it should interface with the existing assessment system. Currently, survey participants are given only a summary of their answers upon completing the assessment. Participants should be able to connect to the analysis system which would provide instantaneous feedback about their responses, interaction network, and other data collected for the assessment.

A second objective of SAPO is the ability to interactively compile and compare reports. Researchers and project managers would like an interface to view and perform analysis of the data during and after its collection. They can use this data to improve upon the strengths and limitations of their project networks. Possible reports include an

analysis of the effect of geographical distance on project communication, or a comparison of a given project to other projects within an organization or system-wide. These reports could also be used as the basis for refining the questions in a follow-up survey for further analysis.

A third objective of the system is to integrate it with the visualization tool developed within the Initiative for Distributed Innovation in order to deliver graphical network connectivity maps with configurable parameters. These visualizations can allow researchers and managers to view, for example, who the influential team members are on a project, or the relative strength of interactions between project sites around the globe. Such visualizations could potentially be valuable in evaluating performance for promotions or rethinking the distribution of work among sites.

Finally, because researchers and managers may want to investigate further into the survey data or reports, SAPO should provide an interactive interface where reports and raw data can be customized and exported for use in analysis software.

1.4. System Overview

OPAS has been collecting data from various organizations since Spring 2004. OPAS has the functionality to create, administer, and maintain dynamic project assessments. It stores collected data on a server for manual analysis. The System for the Assessment of Projects Online (SAPO) is designed to run concurrently on the same server and is able to share access to databases and files. This connection facilitates the flow of data between the two systems. Using this connection, SAPO is able to perform statistical analyses on the assessment data and present it to survey administrators and users. A feature of SAPO is its ability to plug into another module under development by

MIT's Initiative for Distributed Innovation, the NetVis NV2D plug-in. NetVis is a Java-based utility for visualizing networks of connections. SAPO enables users to use NetVis to view connections indicated by the OPAS assessment data. Figure 1 illustrates the connection between SAPO and other modules under development by the Initiative for Distributed Innovation.

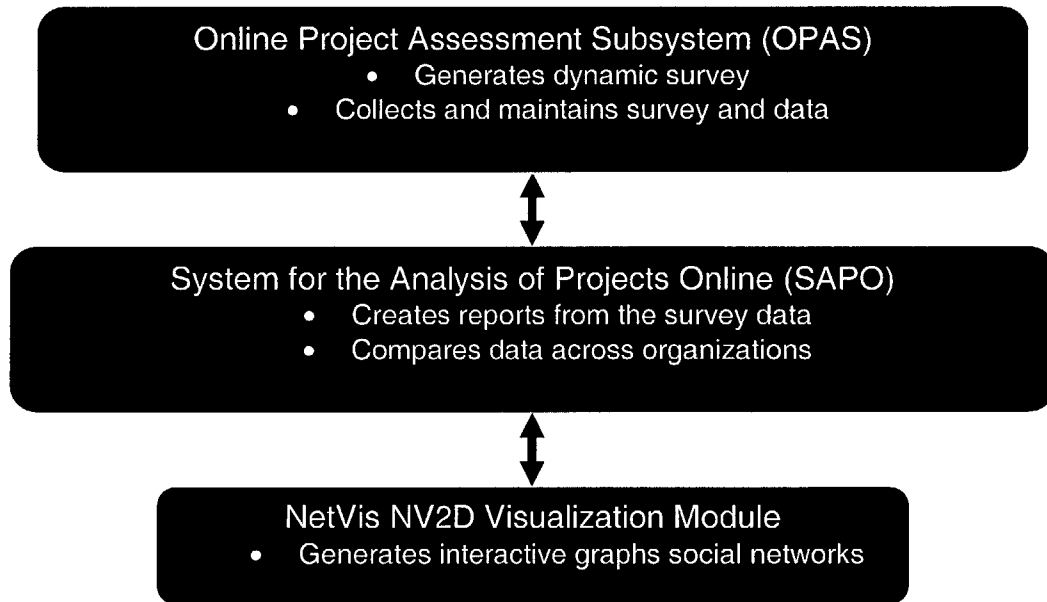


Figure 1: The System for the Analysis of Projects Online bridges components of other systems under development by the Initiative for Distributed Innovation

The analysis subsystem consists of a series of individual analysis scripts, sharing common functions and calling analysis classes. An important set of functions of the analysis system deals with the dataset selection. Since each analysis can be run on any subset of the full dataset, users need a method to filter the dataset to perform the analyses on a single project, set of projects, or set of users. Each analysis page displays the selected dataset for the analysis and a link to allow users to change this dataset. The dataset is stored in session variables which can be assembled by a helper function into an SQL query string. The query string gets appended to all queries used for the various

analyses in order to filter the dataset. This process is shown in Figure

2.

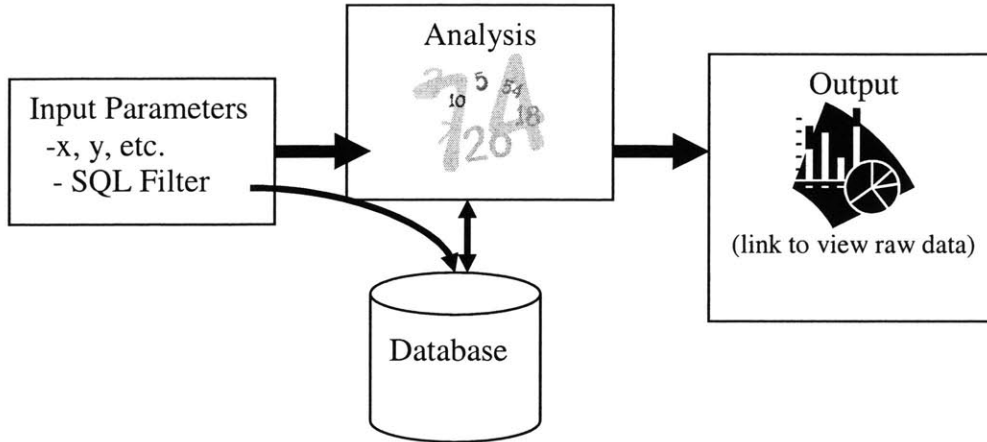


Figure 2: The selected dataset is converted into an SQL query string

Administrative tools allow designated users to configure options related to the analyses. These options include data integration and exporting, and variable configuration.

Since the SAPO system is designed to be released for public use, only open-source technologies were selected to be used in SAPO. Open-source technologies are especially convenient for research since most license conditions allow anyone to use the software free of charge for non-commercial redistribution. The following sections describe the individual software components that power SAPO.

2. System Design and Development

2.1. Technologies

The SAPO analysis system will be designed to run on the existing server architecture used for the assessment system. This model will ensure seamless and fully-compatible integration between the two systems, and since it uses a popular combination of open-source software, extremely flexible and secure web applications can be constructed. The server runs an Apache web server, which handles HTTP requests from users' web browsers. The web server runs PHP code to dynamically generate the HTML code which gets sent back to the client. PHP interacts with a MySQL database stored and secured on the server [6].

The assessment system stores survey data and settings parameters in the MySQL database. Having a shared database system allows the analysis system to have full access to the survey information and results, updated in real-time. This will ensure that system users have a consistent look and feel throughout the assessment experience and that the most accurate view of the data is presented.

2.1.1. Apache, PHP, MySQL, HTML, and JavaScript

Apache is an open-source web server designed to provide a robust means of delivering web content to users over the internet [5]. It forms the backbone of SAPO, receiving requests, executing PHP scripts, and delivering the results to the client's web browser. Apache is well-documented, stable, and efficient, making it a good choice for SAPO.

PHP (PHP Hypertext Preprocessor) is an open-source scripting language that is easy to read, use, and maintain. It easily integrates with the Apache web server and the MySQL database. It allows developers to construct dynamic web applications by providing a means for specifying server-side instructions to be executed by the web server when a client requests a web page [6].

MySQL is an open-source database application that integrates easily with PHP [7]. It was selected over other databases for its widespread use, plentiful documentation and support, and compatibility with OPAS. MySQL has many of the same features as the proprietary databases, but suffers slightly in performance. Additionally, the version running on the server, presently 3.23.58 does not yet support many of the advanced features introduced in later, though more unstable, versions of mySQL. One such example is the lack of support for multi-table updates. This is a minor, but frequently encountered inconvenience that would allow variables in one table to be updated based on the variables in another, and can be remedied by simple PHP scripts. Overall, however, mySQL is able to satisfy the needs of SAPO.

HTML and JavaScript can be dynamically created using PHP and delivered as web content to a user's Internet browser. The HTML defines page formatting, images, links, and content. JavaScript can be used to provide client-side functions and operability to enhance the user interface or input validation.

2.1.2. JpGraph

JpGraph is an object oriented graph creating library for PHP [8]. It is written in PHP and allows for fully-customizable creation of graphs many types of charts and graphs. Graphs are treated as individual objects that get instantiated, have their variables set, and finally receive the command to be drawn. An example use of JpGraph is shown in Listing 1.

```
//define a new graph
include("jpgraph.php");

//instantiate with width 200,
    height 400
$graph = new Graph(200,400)

//set the graph title
$graph->title("Test Graph");

//define a bar graph, passing the
    data to be graphed
include("jpgraph_bar.php");
$bg = new BarPlot($y_data);

//add the bargraph to the drawing
    area
$graph->add($bg);

//make & save the graph as bar.jpg
$graph->Stroke("bar.jpg");
```

Listing 1: PHP code to create a bar graph with JpGraph

2.1.3. NetVis

The NetVis Module is a free open-source Java-based tool to analyze and visualize social networks using data from a variety of input sources [10]. It is currently under redevelopment by MIT's Initiative for Distributed Innovation and provides a robust framework for users to view ties among networks as determined by their assessment data. Plug-ins allow users to view properties about a node or graph and adjust layout algorithms for the nodes on a page. SAPO utilizes NetVis to visualize networks within the organizations.

2.2. Development Strategies

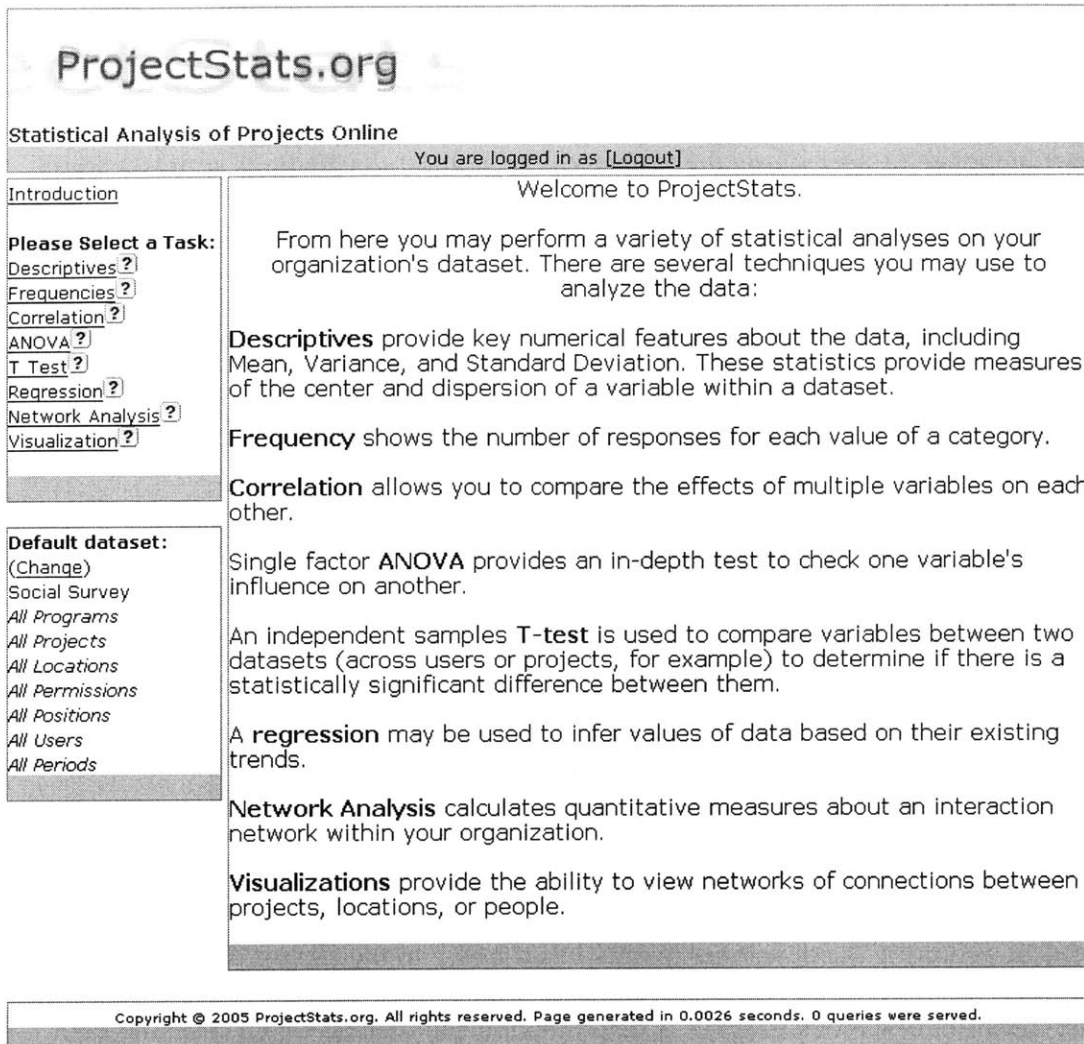
The SAPO system was developed in a modular manner to allow easy inclusion and removal of analysis methods and permission-level access to data. The modular organization enables development of the individual parts of the system: descriptive analyses, predictive analyses, comparative analyses, network analyses, and administrative

functions. Each operation can be run from a standalone script that is supported by a framework held together by session variables, common functions, and robust database access functions. In developing the framework to support the analyses scripts, careful attention was paid in order to ensure the architecture provided access to functions and data where needed, while not imposing too much structure on the scripts that would limit their functionality.

2.3. User Interface

A description of the features offered in SAPO is presented when a user logs in (Figure 3). SAPO is designed to present an intuitive interface for both novice and experienced users. To help users get acquainted with the system, question mark icons are placed next to each keyword on a page. Users can click these icons for a pop-up description of the item and, if applicable, common uses. The help text interface is easy to implement, requiring the developer to only insert row into the *strings* database with the unique identifier of a block of text. A help function can then be called from the PHP code to insert the JavaScript and HTML necessary for the popup.

Analyses functions are presented down the left column, with the default dataset displayed below it. Clicking on the “change dataset” link brings up a new window where users can apply filters to the initial dataset (Figure 4). Users can use the default dataset to pre-configure a filter on the analyses they intend to perform (looking specifically at one project, for example). Each analysis page also has its own dataset which can be set to the default, or be analysis-specific.



ProjectStats.org
Statistical Analysis of Projects Online

You are logged in as [\[Logout\]](#)

Introduction

Please Select a Task:
[Descriptives ?](#)
[Frequencies ?](#)
[Correlation ?](#)
[ANOVA ?](#)
[T Test ?](#)
[Regression ?](#)
[Network Analysis ?](#)
[Visualization ?](#)

Default dataset:
[\(Change\)](#)
[Social Survey](#)
[All Programs](#)
[All Projects](#)
[All Locations](#)
[All Permissions](#)
[All Positions](#)
[All Users](#)
[All Periods](#)

Welcome to ProjectStats.

From here you may perform a variety of statistical analyses on your organization's dataset. There are several techniques you may use to analyze the data:

Descriptives provide key numerical features about the data, including Mean, Variance, and Standard Deviation. These statistics provide measures of the center and dispersion of a variable within a dataset.

Frequency shows the number of responses for each value of a category.

Correlation allows you to compare the effects of multiple variables on each other.

Single factor **ANOVA** provides an in-depth test to check one variable's influence on another.

An independent samples **T-test** is used to compare variables between two datasets (across users or projects, for example) to determine if there is a statistically significant difference between them.

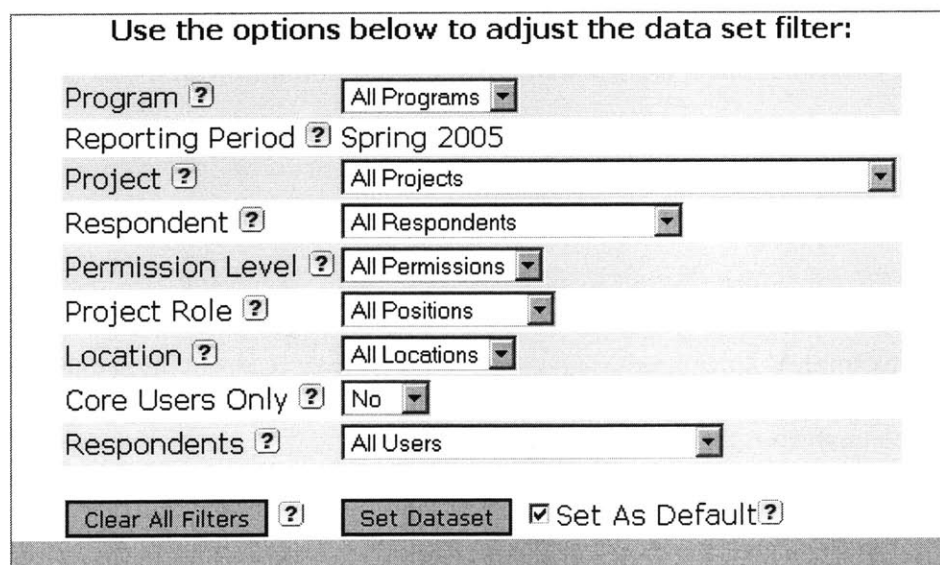
A **regression** may be used to infer values of data based on their existing trends.

Network Analysis calculates quantitative measures about an interaction network within your organization.

Visualizations provide the ability to view networks of connections between projects, locations, or people.

Copyright © 2005 ProjectStats.org. All rights reserved. Page generated in 0.0026 seconds. 0 queries were served.

Figure 3: The main user interface of SAPO



Use the options below to adjust the data set filter:

Program ? All Programs ▾

Reporting Period ? Spring 2005

Project ? All Projects ▾

Respondent ? All Respondents ▾

Permission Level ? All Permissions ▾

Project Role ? All Positions ▾

Location ? All Locations ▾

Core Users Only ? No ▾

Respondents ? All Users ▾

Clear All Filters ? Set Dataset Set As Default ?

Figure 4: The dataset selection tool

A feature to note is that each analysis output page displays a link to the raw data and database query used in the analysis to allow users and administrators to view and save the data for problem resolution or further analysis.

2.4. Issues

The initial implementations of SAPO were very straightforward. A series of scripts operated directly on OPAS data, and generated the graphs and results they were designed to produce. As the system grew, and became more robust and expandable, several issues were faced to meet these needs.

2.4.1. System Architecture

It proved to be a challenge to provide a framework for standalone analysis scripts that would provide them with access to common variables and functions, while not imposing a rigid form on their design. Most issues had to deal with defining a balance between the variables, particularly those particular to the dataset, that would be shared across all components of the system, and those that would be up to the individual analysis script to collect. Setting too many common variables might result in forcing a user who wishes to perform a simple analysis to select among options that aren't relevant to that analysis. Setting too few would mean that a user has to repeat input options from page to page.

The issue was solved by having a main dataset that can be filtered by a number of variables or none at all. This dataset is available to all scripts, but each script can choose to use the existing one or define a new one that will remain unique to the script. Scripts have more overhead by having preloaded all of the common functionality, but this

relatively small cost enables them to make use of as much of or as little preset variables as they find necessary.

Despite efforts to impose the fewest requirements upon analyses scripts, a few required components exist. First, a login and permission check ensures that a user has proper credentials to view a page. The page should also check if an action was passed in calling it. The action is an instruction that informs the page how to handle a given request. For example, the “descriptive statistics” page can be passed the action “getstats” which will instruct it to use subsequent parameters to calculate a requested statistic. One of the features of this design is that it allows for statistic calculation either from user input on the form that is presented when no action is passed, or by another script which may have the feature of saving actions. This would allow administrators configure analyses in looking for an interesting trend, and if one is found, the analysis can be saved and re-run with one click by the administrator or anyone else that is granted access to the analysis. It should be noted that at as of this writing, the framework is present for the saving and reloading of analyses, but the interface for their execution is still under development.

Each analysis output page can use a function that allows it to export the query used in the analysis and its results as a CSV file for users and administrators to view and save the data for problem resolution or further analysis.

2.4.2. Data Format

OPAS stores data in result columns formatted with one column per assessment page, often resulting in the need to separate values by commas, semicolons, or other delimiters needed to preserve various attributes of the data. This format makes for easy storage and retrieval of assessment data on a per-question basis, but can significantly

slow down performance of statistic calculations. Operating directly on OPAS data quickly became a slow, repetitive, and cumbersome process. Even with helper functions to perform repetitive operations, and the storage of frequently used variables and tables in temporary databases, operating directly on the OPAS dataset became prohibitive. I had been avoiding creating a copy of the database because of the ongoing nature of assessments and the implications copies could have on database synchronization.

Since a conversion of the data seemed necessary, I decided to convert the data to the widespread Hierarchical Linear Modeling (HLM) format. HLM offers substantial advantages over the OPAS format in terms of compatibility, exportability and ease of data retrieval and analysis. OPAS data that was separated by commas, semicolons, and other separators would be expanded to have one column for each user response and one row for every person-to person interaction (Figure 5). NULL values are inserted in place of missing responses. User responses are also checked for illogical values (a user entering all zeros, for example). HLM is not very space-efficient with tables in the present dataset averaging 2MB per 100 survey respondents, but the benefits of reduced calculation time, compatibility with other statistical analysis tools, and ease of understanding outweigh the storage costs. Having already written scripts to convert OPAS data to HLM data for external analysis, the decision to migrate to the HLM format seemed clear.

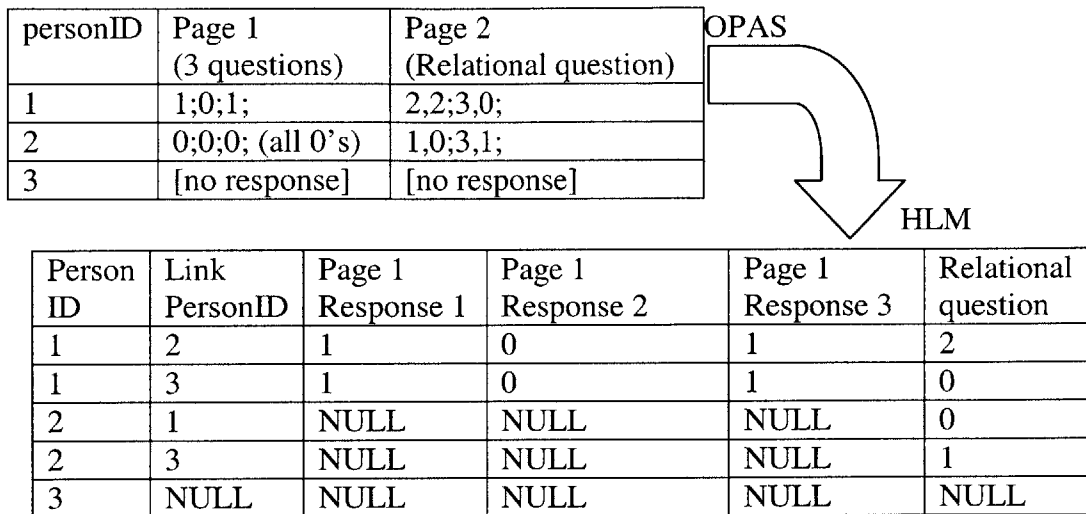


Figure 5: The conversion from the OPAS data format to HLM makes one column for each question, and one row for each interaction link. Blank or invalid responses are set to NULL.

Designing a system around the HLM format was not without complications.

Because each user has one row for each relational link reported in the data, analyses must be careful not to over count user-level data. This problem was solved by separately populating separating the HLM table into user and person tables. The user table contains information particular to each respondent. This could include responses to questions like “How long have you worked at this institution?” or “Does your workgroup share a common lab space?” The user table contains one row for each *respondent*, with columns for each assessment question.

The person table contains information particular to each person-to-person interaction. This could include responses to questions like “How often do you work with John Doe?” or “Have you produced a publication with John Doe in the last six months?” The person table contains one row for each *interaction*, with columns for each question specific to the interaction.

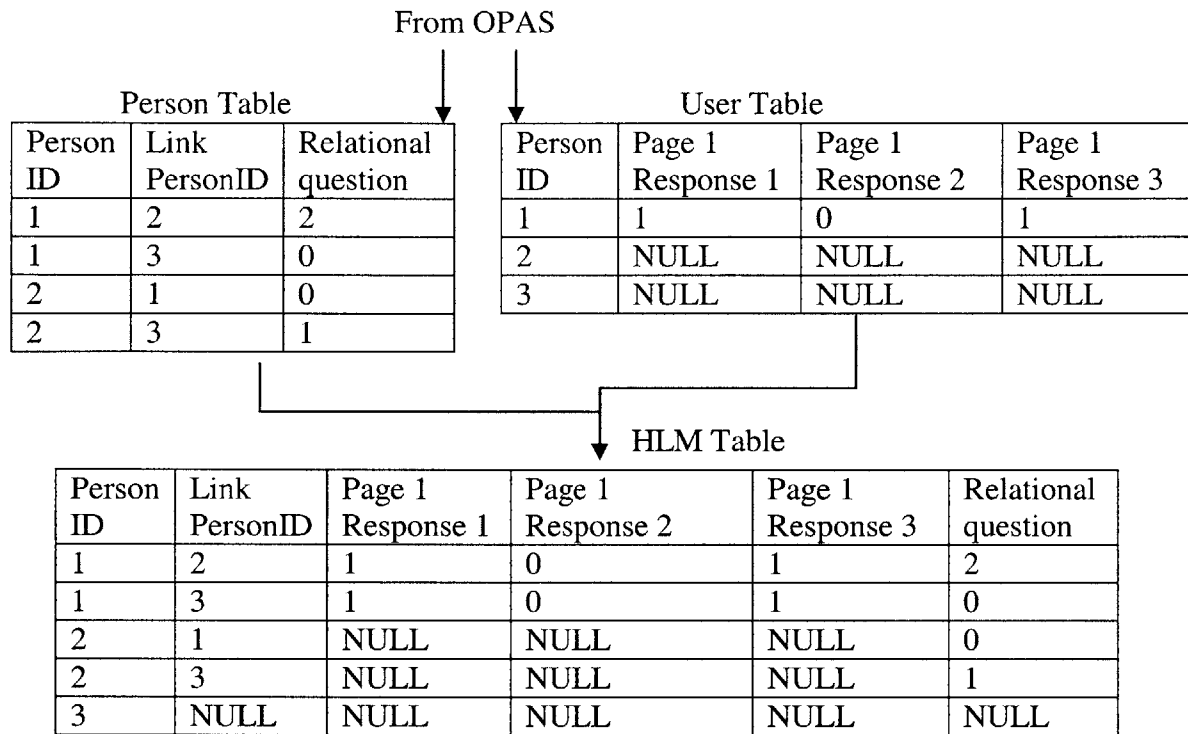


Figure 6: The OPAS data is converted into user and person tables. The user and person tables are then combined to form the HLM table

A conversion script can populate the user and person tables individually from the OPAS data and combine them using a mySQL “join” statement to create the HLM table (Figure 6). Since some analyses may need to be performed on data in both tables, it makes sense to have the data in a single place. However, for those analyses that are specifically directed at user-table questions or person-table questions, having the more concise form may be most appropriate.

The meta table contains information about each column of the user and person tables. The column headings of the user and person table are typically abbreviations used to relate to corresponding columns from the OPAS table they were extracted from. The meta table relates these abbreviated headings to the full question text. It also has a values column that relates integer values to their corresponding text meanings and an attributes column where administrators can specify properties of each question column that are

used to determine which analyses are appropriate for a specific question. Other information stored in the meta table includes the OPAS question number that corresponds to a field, an ordering rank, and the assessment period during which the question was asked.

2.4.3. Data Synchronization

Since SAPO and OPAS had two copies of the data, a method for synchronization became necessary in order to ensure consistency between the databases. As OPAS users complete assessments, SAPO must have immediate access to the new data as it enters the system. A nightly script converts each the OPAS-formatted databases to HLM-formatted databases. This ensures that data accessed by SAPO is no more than 24 hours out of date. If a more immediate view of the data is needed, administrators have the ability to refresh the SAPO dataset. Since intensive restructuring of often large quantities of data is needed, this conversion is slow and would be unadvisable to run very frequently.

Every night, the Linux server executes a Cron script that runs a PHP script to refresh each of the data sets that have been integrated into SAPO. The organizations table in the “projectstats” database lists the databases currently imported into SAPO, with a field to indicate whether the database should be resynchronized by the nightly script. An incremental synchronization of the SAPO database would be the ideal implementation, but would require the creation of a change log for the assessment system. Instead, at present the PHP refresh script takes between 3 and 45 minutes to run for each organization since it erases and recreates the four tables every time it is run, recalculating and inserting composite variables and social network analysis variables, which may change as the dataset changes.

2.5. Maintenance

The SAPO system is designed to easily integrate with any of the existing OPAS organization assessments. It is flexible enough to integrate with non-OPAS assessments, as well. The procedure for this integration is outlined in Appendix A. A database table contains each of the currently imported organizations, and a variable for whether or not that organization should be re-imported on a nightly basis. An administrator can simply insert a row into this database to alert the nightly script to include a new organization in the update. For a quicker synchronization, after inserting the new organization, the administrator can log in to SAPO from the home page and manually force an import of the data. Until the data has finished importing, however, all features of SAPO for that organization will be unusable.

One feature of the OPAS system is that it is modular, allowing developers to add question types to the survey and implement them by filling in a standardized set of functions about the question. SAPO extends this modularity by providing “hooks” in the `hlm_questions.php` file for developers to include a function that converts OPAS data for a new question type into HLM data. Since many questions in OPAS display questions differently for the user while keeping the storage mechanism the same, these hooks can often be duplicated from one another. In the case of a novel question type, a developer must add proper functionality to the `hlm_questions.php` file to alter the user and/or person tables, populating them with data from the stored questions, and update the meta table to provide default text to associate with each response column.

3. Features

The features of SAPO can be divided into three categories. Statistical measures provide numerical description and analysis about the data. Network Analysis provides key features and views of interactions among users within the dataset. Administrative features allow for control over the behavior and operation of SAPO.

3.1. Statistical Measures

Descriptives (or descriptive statistics) express features about the center and distribution of a dataset. These features include the standard deviation, variance, mean, minimum, and maximum of a dataset. These specific statistics were chosen since they are easy to calculate directly through mySQL queries, without having to do (slower) PHP calculations on the data. The added efficiency of retrieving the statistics directly from mySQL allows for easy calculation and comparison of a given organization's statistics with those of other organizations present in the database. Because of the standardized method of importing databases and naming fields between OPAS into SAPO, SAPO is able to match questions that were the same across different organizations since they share the same column name in the database tables. SAPO compares the descriptive statistics for the current user's organization with those of all organizations that have the same column in their database and presents the results side-by-side. This allows a user to see how his or her organization compares with others on questions that were asked of multiple organizations. An example output is shown in Figure 7.

Field	Sample Size	Mean	Standard Deviation	Variance	Minimum	Maximum
Coordination	9,102.00	0.46	0.74	0.55	0.00	2.00
Coordination (other organizations)	1434	0.32	0.65	0.42	0.00	2.00

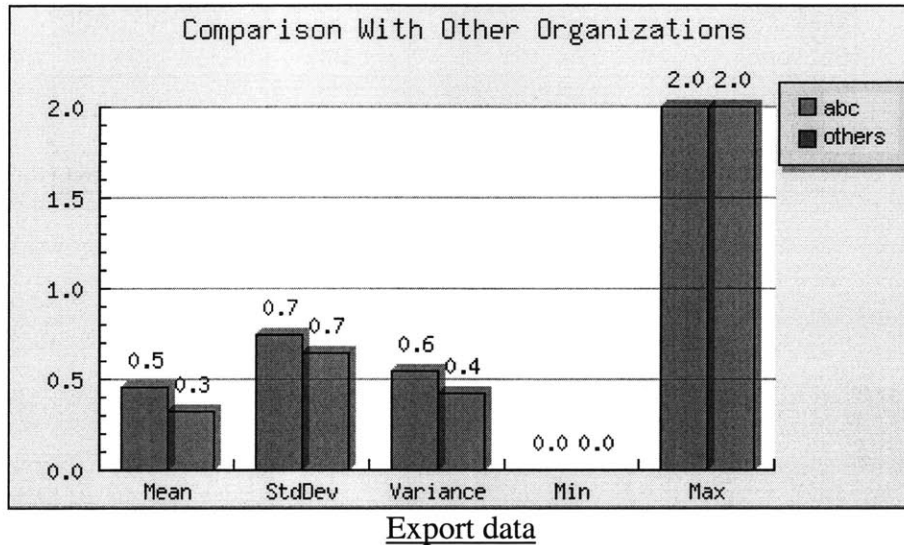


Figure 7: Sample Descriptives Output

Frequencies show the number of responses for each value of a category. Users may be interested in the distribution, for examples, of experienced industry professionals among the projects. Frequencies prepare a numeric and percentage-wise histogram of the data based on the selected dataset and variable of interest. Similar to descriptives, frequencies calculate the averages from other all organizations and display them side-by-side with the observed distribution of the selected organization.

Correlation allows users to test the co-dependence of variables within a dataset. The output, a matrix of r -values between -1 and 1 displays of a predictor the selected variables are of one another. Users may select any subset of the dataset and any number

of variables present in their dataset in order to test for co-dependence. The correlation performs calculations on the lowest common denominator of the dataset, discarding user responses where any of the variables to be analyzed have missing values. The correlation of each pair of variables is calculated and displayed in a half-matrix form.

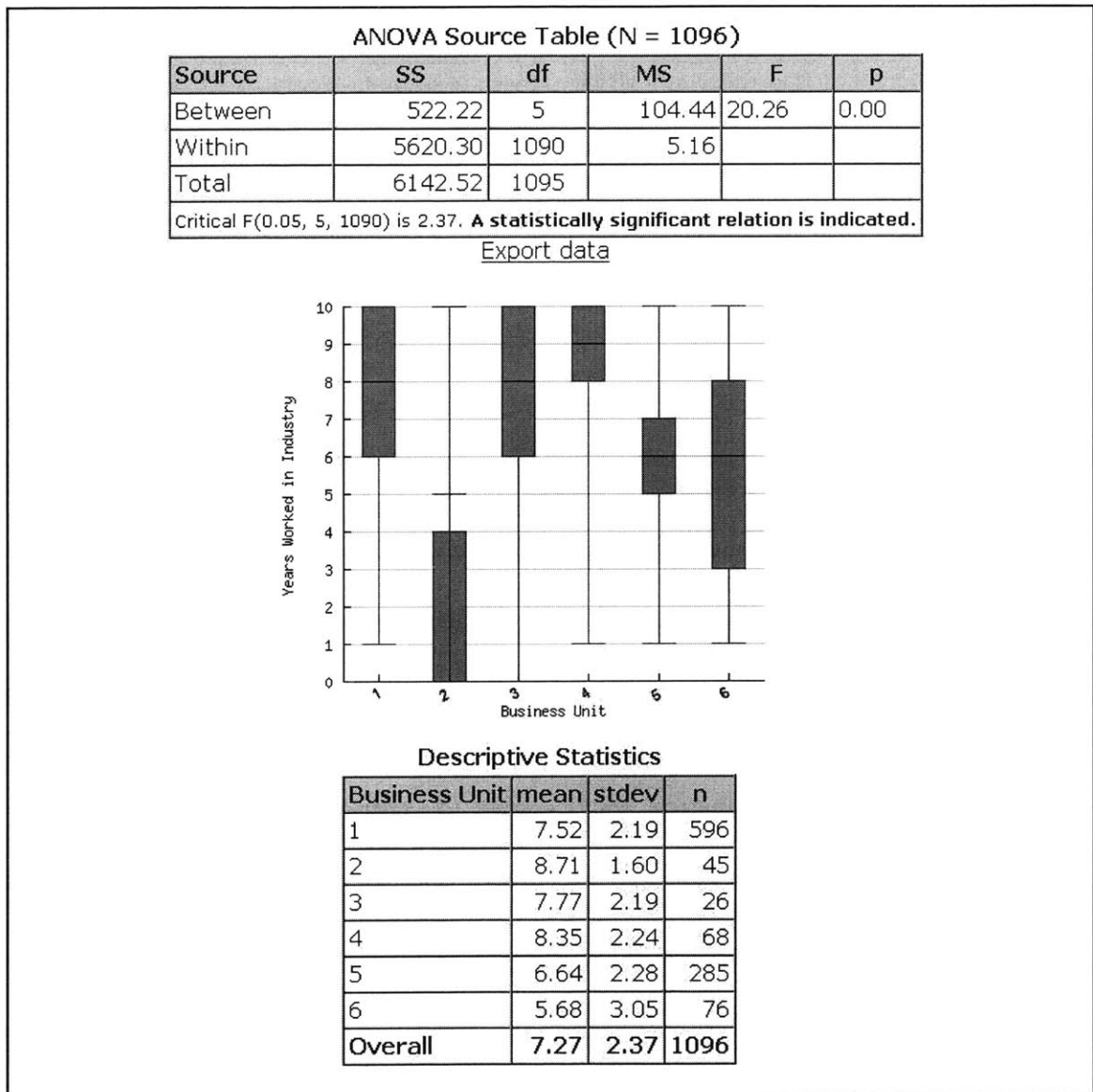


Figure 8: The output of an ANOVA analysis

Single factor Analysis of Variance (ANOVA) studies the effect of one variable on another. The user specifies a treatment (independent) variable and response (dependent)

variable for analysis on the selected dataset. Suppose a researcher wants to compare the effect different locations on communication. The treatment variable would be the location, and the response variable would be communication. An ANOVA class computes the ANOVA table and displays the results (Figure 8). The researcher would most probably first look at the p-value. The closer this number is to zero, the more likely the chance that the treatment variable has a significant effect on the response variable.

It is often helpful for researchers to see a histogram of the data in order to visually observe a relation, if one exists. The ANOVA class produces a “box & whiskers” plot depicting the minimum, maximum, median, upper quartile, and lower quartile of the response variable for each category of the treatment variable.

An independent samples T-Test is like ANOVA, but tests for the significance in change of a variable between two groups. The input is a variable, direction (if only a one-sided test is desired), and two datasets. The output is a t -statistic, significance (p) value, and pooled variance estimate. The strength of this significance is indicated by the p-value. Again, a lower value indicates a greater chance of significance. The pooled variance estimate is a measure of the dispersion in the overall data that is weighted by the percent of information from each dataset.

Simple linear regression attempts to fit a line to a set of data points by the least squares method. A predictor (x) and response (y) variable are taken as inputs, and the regression parameters are the output. The regression class estimates the slope and intercept of the least square line, and the r and r^2 indicators of model fit. It also produces plots of the independent vs. dependent variable with the predicted line, and predicted vs. residuals. These serve as graphical representation of the “goodness of fit” of the model.

3.2. Network Analysis

SAPO provides network analysis tools in order that enable users to study properties of interaction networks within their organization. It is helpful for organizations to be able to identify key features of networks that are considered to be successful, and see “key players” within these networks.

3.2.1. Visualization

The visualization page is an interface to a class that converts HLM data into a special format used by the NetVis visualization applet. An example implementation of the class is shown in Listing 2. On the interface page, a user specifies the relational question, the direction of the question, a response threshold, and the cluster variable. The relational question is a question about a user’s interaction with another user, such as “How often have you worked with this person in the past 6 months?” The direction tells the visualization which way to direct the responses: outward if the question asks about a user’s interaction with other users, inward if the question asks about another user’s interaction with the user completing the assessment, and bi-direction if the question asks about joint interaction. The response threshold sets a minimum value for a link to be considered, and the cluster variable allows the user to control the variable that gets displayed as nodes. The cluster variable enables users to view interactions on a variety of levels: person-to-person interactions, project-to-project interactions, etc. The visualization tool also displays a link to the input file for the applet that can be saved and input directly into the standalone NetVis application.

```

//import and instantiate the visualization class
import("Visualization.php");
$vis = new Visualization;

//set the node labels to be the users' first and last names
$vis->SetLabel("concat_ws(' ',trim('firstName'),trim('lastName'))");

//set which fields get displayed on the node (name => table column)
$fields = array("Business Unit"=>"unit", "Location"=>"location");
$vis->SetNodeInfo($fields);

//set the database table, and the field names that contain the ego, alter,
//question fields, and cutoff threshold
$vis->SetTable("dbTable");
$vis->SetEgoField("PersonID");
$vis->SetAlterEgoField("Link_PersonID");
$vis->SetQuestion("Relational_Question");

//set a threshold cutoff, and specify the direction of the question
$vis->SetThreshold(3);
$vis->SetDirection("outward");

//generate the NetVis CSV file
$vis->GenerateCSV();

//display the NetVis java applet, or output an error
if(!$vis->failed)
    $vis->ShowApplet();
else
    echo $vis->failed;

```

Listing 2: Example usage of the Visualization Class

3.2.2. Network measures

Network measures are statistics that provide quantitative values for the strength, size, and dispersion, and other features of a network. It provides a means for researchers to classify networks. The interface page takes similar arguments as the visualization page: relational question, direction, and threshold. It passes these variables to a social network analysis class that calculates the network measures and displays them in table form. Each of the measures and its meaning is displayed in Table 1.

Table 1: Summary of network measures and their meanings

Network Measure	Description
Degree centrality	Centrality based on degree, or the ties into and out from nodes
Betweenness Centrality	Centrality based on betweenness, or the bridges among nodes
Closeness Centrality	Centrality based on closeness, or the shortest path among nodes

Density	Sum of node ties divided by the number of possible node ties
Transitivity	Fraction of triads which are transitive. (If a goes to b, and b goes to c, does a go to c?)
Structural Holes	Redundancy and constraint of the social network
Constraint	Measure of structural holes involving the extent of the central node's investment in each contact who is invested in the other contacts.
Effective Size	Actual size of the central node minus the average degree of alters not including to ego
Efficiency	Effective Size divided by the number of alters in each person's network
Hierarchy	Extent to which the constraint on a central node is concentrated on a single contact
Cohesion	Distance between all nodes in a social network
Shortest Path	The geodesic distance to the shortest path between a pair of nodes
Core/Periphery	Identifies dense/connected core and a sparse/unconnected periphery

Adapted from <http://www.NetVis.org>

3.3. Administration

A console was needed in order to manage some the data that goes into the analyses that users are able to perform. Initially, users each had the ability to save their own copy of the data that allowed them to exercise administrative privileges so they could filter the data, and manage variables within the dataset. As the development of SAPO went on, it became apparent that users would get the most out of a dataset that has been pre-configured by an administrator, while still allowing them some flexibility over what gets analyzed. As a result, users all work on the same master copy of the data that administrators can pre-configure with composite variables and social network analysis variables. Administrators also oversee the management of data as it gets copied in to and out of SAPO.

3.3.1. Data Synchronization

The data synchronization feature allows administrators to manually refresh the SAPO dataset when there are new respondents in the OPAS assessment, and waiting for the nightly update would be inconvenient. The import script compares the number of respondents in the current dataset with the number of respondents in the OPAS database to alert the administrator if a re-import is needed. This count is simply a comparison of response entries and does not take into account respondents who may have updated their responses. Because of the differing data formats and the lack of change logs, there is no easy way to check for updated responses. Instead, when a re-import is performed, the entire dataset is refreshed into temporary tables, and when the refresh is complete, the temporary table replaces the working copy.

Since the import process is time-consuming, a status page was needed to inform the administrator how far along the process is. Since PHP is strictly a server-side application, status pages are not straightforward. Several approaches were studied in order to design this implementation. One method involves computing a mean time-per-user of the import, and using this number to estimate a time for the import based on the number of users. A JavaScript clock could count-down until the import is estimated to be complete. This design was decided against since the import time per user proved highly variable and is most heavily dependent on the number of between users, which is difficult to determine with quick calculations. Instead, a status page was designed as an iframe (a page within a page) that compares the number of users in the imported dataset with the number of users in the original dataset in order to determine the percent completion. The iframe refreshes itself using an HTML META parameter on a regular basis that depends

on the number of users in the import. This is done because an import with 500 users is less likely to notice a change in the same time interval as an import with 50 users. The combination of user-side refreshing and server-side calculation, allows PHP to update the administrator on the progress of the import process. The calculation is not perfect, however, since it does not take into account the time to recalculate composite variables and social network analysis variables, but that amount of time is difficult to estimate and generally small when compared to the time for the import.

3.3.2. Exporting Data

Once data is refreshed in the system, administrators are able to export the raw data for use in statistical analysis software. The export page allows administrators to specify the data separator, enclosure, dataset, line break character, and filename. These options allow users complete flexibility over the file that is created. For imports into packages that prefer comma-separated data (CSV) administrators can specify a comma as the separator. Alternatively, semicolons or any other ASCII character could be used. Since Windows, Mac, and UNIX systems all use different line break characters, the export page allows the user to select between separating rows of data by PC line breaks (`\r\n`), Mac line breaks (`\r`), UNIX line breaks (`\n`), or HTML output (`
`). Administrators are also given the option to export only user information, person-to-person interaction information, or the HLM table that combines the two. The dataset filtering available for the analyses is also available for the export, which can have the effect of filtering the output to only display data matching a particular subset of users or projects.

Since converting thousands of rows of a mySQL table to a text file takes roughly one second per 1000 rows, the export can be time consuming for larger datasets. In order to alleviate this wait, the export file as performed with the most common set of options is cached between updates. A link to this file is presented on the export page, but if a user wishes to save the file with non-standard options, the full export will still need to be performed.

3.3.3. Variable Setup

When SAPO imports its data from the OPAS system, it has access to a full set of information about the assessment questions, as well as user responses. Using this information, the import script is able to determine properties about most variables that it imports. On the most basic level is whether or not a variable is text or numeric. Numeric variables are permitted for most analyses, while text variables are not. In case a variable is mislabeled, administrators need to manually edit the attributes of the variable in the organization's meta table.

The present version of SAPO has a loose configuration of which variables get shown for which analyses. Each variable can have one or more have configurable attributes as defined in Table 2.

Table 2: Variable attributes and their functions

Attribute	Use
Numeric	Identifies fields that can be used in statistical analyses
Categorical	Text field that may define an independent variable for ANOVA
Relational	Identifies questions that ask about a person's network
Vis-node	Identifies fields that will be displayed on the visualization output when the mouse is hovered over the node
Static	Identifying fields of the data, e.g. personID, projectID, etc.
Hidden	Internal, not displayed anywhere except in raw data exports

Other properties associated with the variables include the assessment period in which they were asked, the corresponding OPAS question number, and a rank for ordering in the display. The period is used to hide questions that were not asked of users before a particular assessment period.

The values column allows integer values of the text field to be associated with text labels. For example, if a question has three response levels 0, 1, and 2, the values column can store the semicolon-separated text labels “no;maybe;yes”. As of this writing, the values column is only used for relational questions, but the framework is in place to expand its use to all variable types.

Administrators have the ability to create variables from other variables and add them to the data so they can be used in analyses. The composite variables can be any of three types: sum, average, or binary. A sum is simply the linear combination of all selected variables. If each variable indicated whether or not the user had checked a box on an assessment page, the sum would indicate how many boxes had been checked by a user. Average variables compute the statistical mean of the selected questions and insert that mean as a variable in the user’s dataset. Binary variables take the average of the selected variables, and if their average meets a specified threshold criteria (e.g. is greater than or equal to 3), sets the composite variable to 1, otherwise it gets set to zero.

When a composite variable is inserted into the dataset, an entry is created in the organizations’ settings table of the projectstats database. When the OPAS data is refreshed, all of the composite variables stored in the settings table are re-run in order to ensure they match the new data.

The network measures available for users to view can also be incorporated into the dataset to be used in analyses, much like the composite variables. Administrators can designate a question, direction, response threshold, and network boundary for the inclusion of SNA variables into the dataset. The question, direction, and response threshold are the same parameters passed to the visualization and network measures tools. A network boundary specifies the level of network of interest. This could be an individual's network, a project-level network, or a program-level network. The SNA measures are then inserted as columns in the person and HLM tables. A script calculates the measures once for each wave or period of data for each network boundary, and updates the person and HLM tables to include the measures for all rows that match the calculated period and network boundary. This means that setting a project-level network boundary would result in all users that share a common project and period sharing the same set of network measures.

4. Conclusion

4.1. Lessons Learned

I learned several lessons in the development of a dynamic project analysis system. As SAPO was being developed, several organizations had already completed assessments in the OPAS system. The most difficult part of developing a dynamic system was designing it. I attempted to plan out SAPO in its entirety before attempting the physical building of the system. Just as I would devise an elaborate scheme for designing a part of SAPO, user requirements would change or an unforeseen challenge would surface, and I was forced to rethink the design. Perhaps the most important lesson I learned from all this planning and re-planning can be summarized by the popular design mantra: “Keep it Simple, Stupid”. The most successful features of SAPO were not designed through detailed planning, but rather by first building a simple system that met the functional requirements of the feature. At that point, as user requirements changed or more features became necessary, it became easy to build on top of the existing feature or rewrite parts of the feature in order to implement these changes. I emphasize that it is still important to carefully plan a system since SAPO would not be nearly as dynamic without the proper framework, but one must be careful not to over-plan a system, since that can make the functional requirements of a system seem prohibitively complicated.

The collaboration of from multidisciplinary fields provides another source of valuable lessons in system design. SAPO and OPAS bring together three types of users: computer scientists, social scientists, and program administrators. As a computer scientist, I often found it difficult to understand functional requirements placed upon the

development by the social scientists and program administrators. Sometimes I found it difficult to understand how a feature would be of any use since it seemed redundant or a simple combination of other functions. Computer science trains one to think in terms of the lowest common denominator, trying to implement functions in the most efficient way possible. For program managers and social scientists, however, these features may make the data more understandable. Often, it was not until after I had implemented a feature and saw it get used that I fully realized the benefit and utility it provided.

Furthermore, in working to create an efficient system that met the needs of program managers and social scientist researchers, I learned of the difficulty of meeting the different perspectives of the two groups. Social scientists are interested in group interactions, network dynamics, and the association between project processes and outcomes. Program managers are interested in creating reports about their project, finding indicators of project success, and allowing OPAS respondents to access data directly about their projects. The different needs are addressed through a permission-based system. Users with different levels of permissions can access different parts of the system. A project team member may then see analyses about his or her project. A program manager can perform analyses related to their program. A social scientist can compare outcomes of analyses for certain programs with those of others. In designing a system to meet the needs of different groups of users, I learned that a “smart” user interface can enable groups with different perspectives utilize a common system.

4.2. Future Work

The availability of data made it easy to test SAPO on a wide variety of organizations and projects, but it also made it difficult to simultaneously meet the

different analysis features requested for each organization. The multitasking that resulted left a few parts of the SAPO system only partially complete. Future work may complete these features, so they are discussed here.

4.2.1. User Management

At present, SAPO has two levels of permissions: users and administrators. Users are able to run analyses, and administrators are able to create import/export data, create composite variables, and import social network analysis variables into the data. The framework exists, but has not fully been utilized to have further levels of permissions, allowing administrators to configure permissions for individual variables and analyses.

4.2.2. Multiple Regression

The current regression script is a simple linear regression. It is, in a sense, a more informative correlation between two variables for a specified dataset. A powerful feature for SAPO would be to extend this functionality to running a regression on multiple variables. This would allow program managers and researchers to simultaneously examine the effect of several indicator variables on a project metric, for example. This feature is planned, but still incomplete as of this writing.

4.2.3. Variable Setup

Administrators would like a way to set up which variables get displayed for specific analyses, and set period or other question info that may not have been dynamically assigned during the import from OPAS. While this information can be edited by directly changing the database, a user interface is necessary to make the process

more intuitive, and to allow users who may not have direct access to the database the ability to edit variable attributes.

A powerful feature of being able to dynamically setup variables would be the ability to assign universal question identifiers to variables that would link them to a question database of workgroup analysis-related questions. During the import, question field names are used as the criteria for determining if questions across organizations are identical and can be used in a comparative analysis. Having a unique identifier would allow administrators to tag questions that weren't dynamically determined to be identical (perhaps because of slightly different question text, or a manual import of the data) as "identical enough" to be used for comparative purposes. Furthermore, having a database of assessment questions help to standardize the question information in SAPO and can feed into the creation process of the assessments as they are designed.

4.2.4. Saving Analyses

Administrators and users would like a way to save analyses for quick access at a later time. This feature would allow administrators to direct users to interesting analyses of trends observed in their data, or users could simply save an interesting analysis for use later. The functionality for saving analysis settings exists in the projectstats database, and on the individual analysis pages but an interface to allow users to save and retrieve these analyses is not yet complete.

4.3. Research Implications

Developing a robust analysis system that integrates into a complete project assessment package has proven to be both a great challenge and opportunity. The

completed system, an integrated series of assessment, analysis, and visualization tools, offers substantial advantages over existing systems. Organizations have a comprehensive online interface enabling them to conduct an assessment and analysis of their project interaction networks. Utilizing research that has linked properties of successful project teams and key network characteristics can encourage more streamlined interactions between project members that can yield better performance and more creative ideas. In turn, the network data, which has traditionally been difficult to collect, can be used to improve upon scientific knowledge of social networks, leading to more accurate and detailed analysis methods [9].

5. Bibliography

- [1] S. Karlin. "From Dilbert to DaVinci: Companies find new ways to harvest their engineers' creativity." *IEEE Spectrum*. Nov 2004
- [2] Initiative for Distributed Innovation. <http://idi.mit.edu>.
- [3] F. Shiraishi. System for the Online Assessment for Distributed Projects. May 2004.
- [4] The Cambridge-MIT Institute. <http://www.cambridge-mit.org>.
- [5] "Apache: HTTP Server Project" http://httpd.apache.org/ABOUT_APACHE.html
- [6] "History of PHP and related projects" <http://us2.php.net/manual/en/history.php>
- [7] "MySQL PHP API" <http://dev.mysql.com/doc/mysql/en/PHP.html>
- [8] "JpGraph – PHP Graph Creating Library" <http://www.aditus.nu/jpgraph/>
- [9] J. Cummings and R. Cross. "Structural Properties of Work Groups and their Consequences for Performance." *Social Networks*, 25(3), 197-210.
- [10] "Netvis Module – Dynamic Visualization of Social Networks"
<http://www.netvis.org>

Appendix A Integrating non-OPAS Data

Since the SAPO back-end maintains tables in the standardized HLM format, it is possible to integrate non-OPAS data into SAPO. This import has been done with two dataset to date. In order to prepare non-OPAS data for import, the HLM data should be separated into user and person tables as outlined above, with a meta table containing attributes of each column of the user and person tables. To allow for user authentication, a database needs to be created by the same name as the data set tables. This database should mimic the OPAS databases, containing at least the administrators, periods, projects, settings, and user tables. These tables are used to allow for user authentication, to reference assessment period and project information, and to maintain terminology settings. The person, user, and HLM tables must also contain the fields outlined in Table 3, in addition to fields used in the analysis.

Table 3: Description of fields required in the SAPO tables

<i>Field</i>	<i>Description</i>
uRowID	User table rowID
pRowID	Person table rowID
uID	User table rowID, can match uRowID
personID	Person table rowID, must be unique and identify relational egos
mat_PID	relationalID for links in the person table. Should match a personID in another row
projected	Identifier for the project the user is on
Program	Identifier of a superset of a project. All rows may have the same program name
Permission	1 for administrators, all others for non-administrators
Position	The user's position (e.g. Project Manager) within the project or organization
periodID	Identifier for an assessment period. Should match periodID in the periods table of the organization's database
Institution	Label name of a respondent's organization
institutionID	Unique identifier of a respondent's organization
firstName	Respondent's first name

lastName	Respondent's last name
Responded	1 if respondent responded to the survey, else 0
new	0 for core users, 1 for non-core

Other fields may be set by any name desired, but should be referenced by the meta table, and given attributes as defined in the variable setup section.