INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®
CONTEXT+: Development Environment for 3P-able Context Maps

Kang Zhou

A MAJOR REPORT
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

CONCORDIA UNIVERSITY
MONTREAL, QUEBEC, CANADA

July 2002

© Kang Zhou, 2002
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-72952-4
Abstract

CONTEXT+: Development Environment for 3P-able Context Maps

Kang Zhou

This report introduces a kind of high-level notational technology, "Context Maps", first introduced by Dr. Wojciech M. Jaworski, that can be used in information system and software engineering. In practice, with map applications multiplying, in order to use this technology better, the development of efficient visualization CONTEXT+ tool is necessary and important. This report describes the process of developing and applying CONTEXT+ tool for 3P-able Context Maps in detail. All programs are developed with Visual Basic and the Context Maps are displayed within MS Excel spreadsheet. By applying and processing the various applications, the functionality and attributes of CONTEXT+ tool are completely demonstrated in this report. The main objective is to supply a set of advanced, easy to learning and powerful tool for 3P-able Context Maps. The purpose of this project is to illustrate that Context Maps is not only a methodology and formalized notation for representing the knowledge, but also a powerful tool that can be used directly to manipulate on the formatted knowledge.
Acknowledgements

I wish to thank all those who made the final realization of this dissertation possible. It is not possible to mention all their names, however I would like to express my special gratitude to the following contributors.

I am greatly indebted to my supervisor, Professor Wojciech M. Jaworski, who encouraged my interest in the knowledge representation field, for his technical advice, generous attention, constant help and critical remarks throughout this work. He patiently guided to me with his knowledge about information systems and encouraged me in tackling various difficult issues.

I am pleased to express my gratitude to Professor Peter Grogono for accepting to be my co-supervisor. He gave me many helpful comments on the report.

My sincere thanks are extended to Professor Olga Ormandjieva, who is an examiner for the report, for her valuable suggestions and support.

My thanks also go to Halina Monkiewicz, the Graduate Program Secretary, for her support and collaboration.

I would like to thank a group of students, ShengTian Yang, YanHong Li, Li Xuan, Yi Chen, MinHua Chen and Zhu Lin Lu, for helping me to convert and verify the source code into ContextMaps.

Finally, my special thanks are also due to the faculties and staffs in the Computer Science Department at Concordia University, who provided extensive support during my master program’s study.
Table of Contents

List of Figures ........................................................................................................ vi
List of Tables .......................................................................................................... viii

1. Introduction ....................................................................................................... 1
   1.1 Background ................................................................................................. 1
   1.2 Objective of Study ...................................................................................... 2
   1.3 Research Procedure .................................................................................... 3

2. Context Maps: A Notational Technology ..................................................... 6
   2.1 Context Maps Overview ........................................................................... 6
   2.2 Context Maps Paradigm and Technology .............................................. 7
   2.3 Context Maps Syntax and Process ........................................................... 8
   2.4 Context Maps Terms and Constraints .................................................. 11
   2.5 Context Maps Notation ............................................................................. 12
   2.6 Context Maps Tools ................................................................................ 14

3. CONTEXT+: Development Environment .................................................. 16
   for Context Maps ............................................................................................. 16
   3.1 Main Goal .................................................................................................. 16
   3.2 3P-able Format Introduction .................................................................... 17
   3.3 Tool Features ............................................................................................ 18
      3.3.1 Query Usefulness in Context Maps .................................................... 19
      3.3.2 Mode Selection Usage ...................................................................... 23
      3.3.3 Apply Color Usage .......................................................................... 25
      3.3.4 Context Map Help ........................................................................... 25
   3.4 Practical Applications ................................................................................. 25

4. CONTEXT+: Installation ............................................................................... 27
   4.1 Required Files .......................................................................................... 27
   4.2 Installation ................................................................................................ 28
4.3 Development Tool ......................................................... 28
4.4 Code Design ................................................................. 29
4.5 General Constraints ....................................................... 29

5. CONTEXT+: ContextMaps Documentation .................................. 30

6. Conclusion and Recommendation ........................................... 41

6.1 General Conclusions ..................................................... 41
6.2 Recommendations for Future Works .................................... 42

Bibliography ...................................................................... 44

A. Printed Materials ................................................................ 44
B. Online Resources ............................................................ 45

Appendix A---- CONTEXT+: User Manual .................................... 46

A.1 Start Program .................................................................. 46
A.2 CONTEXT+ Tool Functionality ......................................... 47
  A.2.1 Show Schema .......................................................... 47
  A.2.2 Computer Cardinality ................................................. 47
  A.2.3 Apply Color ............................................................ 48
  A.2.4 Help .................................................................. 52
  A.2.5 Query .................................................................. 53
    A.2.5.1 Query Section Functionality ................................. 53
    A.2.5.2 Mode Section Functionality ............................... 53
    A.2.5.3 Output Section Functionality .............................. 53
    A.2.5.4 Run Section Functionality ................................. 53

Appendix B---- Program Source Code ........................................ 73

A. User Form Source Code .................................................. 73
  A.1 frmProjection ............................................................ 73
  A.2 frmSelectConcepts ...................................................... 77
  A.3 frmSelectWorksheets .................................................. 77

B. Modules Source Code ....................................................... 77
  B.1 mApplyColor ............................................................. 77
  B.2 mAutoByColor .......................................................... 79
  B.3 mMerge ................................................................. 86
  B.4 mProjection ............................................................ 89

C. Classes Source Code ....................................................... 98
List of Figures

Figure 2-1 State Machine of “Inception” Phase .................................................. 8
Figure 2-2 The Context Maps Notation ............................................................... 12
Figure 3-1 Real Example of Context Maps ......................................................... 20
Figure 3-2 The Result Maps after Running “OR” Operation .............................. 21
Figure A-1 MS Excel Standard Dialog Box ......................................................... 46
Figure A-2 Sub-Menus of the CONTEXT+ Menu ............................................... 47
Figure A-3 Syntax for Cardinality ................................................................. 48
Figure A-4 The Color Index .............................................................................. 49
Figure A-5 Input Test sheet for Apply Color Function .................................. 51
Figure A-6 Output Result from Apply Color Function .................................. 51
Figure A-7 Excel Version of Help Sheet ........................................................ 52
Figure A-8 Query Form for CONTEXT+ Menu .............................................. 53
Figure A-9 Query Section ............................................................................. 54
Figure A-10 Test Input Sheet for AND, XOR, OR, NOT Operation .................. 54
Figure A-11 Result for AND Operation .......................................................... 55
Figure A-12 Result For XOR Operation ......................................................... 56
Figure A-13 Result For OR Operation ............................................................ 56
Figure A-14 Result for NOT Operation .......................................................... 57
Figure A-15 Input Test Sheet for Query By Color .......................................... 59
Figure A-16 Temporary Result Sheet For Implementing Predefined Query .... 60
List of Tables

Table 1: ContextMap Schema for mapping VB code .............................................. 33
Table 2: Top-Map Schema and part of map .............................................................. 33
Table 3: ContextMap of Code Sub Merge ............................................................... 34
Table 4: ContextMap of Code Sub InteractiveProjection ........................................ 36
Table 5: ContextMap of Code Sub PromptWorkSheets ........................................... 37
Table 6 ContextMap of Code Sub CheckBold ....................................................... 38
Table 7 ContextMap of User Manual “CONTEXT+ Toolbar Menu” ....................... 40
Table 8 ContextMap of User Manual “CONTEXT+ Query Form” ......................... 40

Important Notice:

All of the Context Maps Syntax and Patterns shown in above figures are Copyrights
Chapter 1

Introduction

1.1 Background

Context Maps, originally called jMap (jointed map), were first introduced by Dr. W.M. Jaworski in 1999. The technology was initially developed for recovering and refining knowledge from legacy systems. By using the concept of spreadsheet structure, it is feasible to describe and process conceptual information. It is easy to integrate different information into one consistent map using Context maps notation, so it is a kind of high-level notation technology. The use of Context Maps notation allows efficient recovery and modeling of generic schemata. This technology can be applied in many domains, such as modeling, mining, evaluation of contexts, enterprises, methods, processes, projects, artifacts, databases, websites, information system, knowledge models with generic templates, domain experts, and proprietary notational technology.

Until today, Context Maps can deal with large amounts of data and represent knowledge assets in 3P-able format. The “3P-able” means Plug-able, Process-able, and Pattern search-able. And “large amounts” indicates data grow exponentially. In order to enhance efficiency and satisfy the new technical requirement, it becomes very necessary to develop the special purpose tools applied in Context Maps. By considering the basic structure of 3P-able format, it is possible for us to supply a set of advanced, easy to learn, and powerful tools for Context Maps to manipulate the formatted knowledge. The CONTEXTt+ tool is developed under this environment. The full package contains basic
functionalities (some of which was developed by predecessors (Varacalli [1998] and Stoyanov [1999]) and new functionalities. All of the work was directly instructed by Dr. W. M. Jaworski. He provides theoretical constructs for CONTEXT+ tool.

1.2 Objective of Study

This report introduces a notational technology named Context Maps. The application environment of Context Maps is Microsoft Excel. Microsoft Excel is a spreadsheet program that has been widely used for data storage and manipulation. It allows user easily to analyse data and mine data by using its standard toolkit. However, query in Context Maps requires visualization of increasingly large data sets in multiple dimensions. With the data multiplying, the Excel inherent toolkit is not satisfactory for scientific research, especially for Context Maps studies. Therefore, efficient visualization tools within this environment are necessary.

The objective of this project is to develop CONTEXT+ tool working in 3P-able Context Maps, which can allow users to deal with Context Maps very efficiently and easily. For users of Context Maps, the CONTEXT+ tool increases the efficiency of data manipulation and visualization, and allows exploration of large data sets that would not be done by hand.

The application was written in Visual Basic with emphasis on using Micro Office application, the MS Excel spreadsheet is effective used in this project, and the user-
friendly graphical interface (GUI) is utilized to guide users through application steps. Learning CONTEXT+ tool is simple and straightforward.

1.3 Research Procedure

The research work for this report was supervised by Professor Wojciech M. Jaworski. It was started in January 2002. The Context Maps operation environment study has been involved in the development of tools. For CONTEXT+ tool, the predecessor’s developed result is absolutely an essential reference and as a component part.

The important steps in the development of this project are:

1) Map out the knowledge of the target, analyze the requirements of this project, and list all functions to be developed.

2) Study the Context Maps, and try to convert the relative model into a spreadsheet with Context Maps notation, focusing on the understanding the basic expression of this new technology in Excel.

3) Get familiar with predecessor’s program, and understand the structure of the old code.

4) Design project, develop program for CONTEXT+. source coding in MS Excel by using Visual Basic.

5) Integrate the program, test all functions and adjust CONTEXT+.

6) Clean up the source code, and convert all source code and the user manual into the Context Maps.

7) A final project package will contain a full description of manual, sample Excel file and source code.
8) Make a conclusion for this research work and provide recommendations for future works.

Before finish this project, the report is a very important portion. In this report, the first Chapter introduces the background, objective of study, and the procedure of project.

Context Maps information is introduced in Chapter 2, it includes Context Maps paradigm, technology, syntax, notation, and tools of Context Maps.

Context Maps tool development is main part in whole project. Chapter 3 describes the main goal of CONTEXT+ development, practical application and focus on the features of the CONTEXT+ tool introduction.

After CONTEXT+ developed, how to use this CONTEXT+ is very critical to users. Chapter 4 depicts the operation environment, installation, required files, and General Constraints.

In order to illustrating the usage of ContextMap and implementation of the source code, in chapter 5, an efficient means it introduced to represent the source code and user manual: ContextMaps format documentation.

Chapter 6 provides an evaluative conclusions and recommendations for future work.
In addition, there is a full description of user manual for CONTEXT+ tool. Appendix A focusses on the user manual for tool functionality in detail and provide user an operation solution.

Appendix B provides main detail source code which is written by VB.
Chapter 2

Context Maps: A Notational Technology

2.1 Context Maps Overview

In the computer software industry, there are many technologies and methodologies for specifying, constructing and visualizing software-intensive system. For example, Unified Modeling Language (UML) is a tool with standard notation for expressing a system’s blueprint.

Context Maps is advanced, powerful and easy to implement method for information representation. It can be used for representing architectures, structures, and reusable templates of information systems. Context Maps notation allows efficient recovery and modeling of generic schemata for processes, objects and views in these systems.

Concretely, Context Maps is a formal representing method for information system with a set of predefined formal notation. It consists of an unlimited number of context tuples that are generic associations of set members cast in roles. In the extended spreadsheet, a column of roles and the related set members define context tuples. Graphically, a context tuple is represented by a compounded edge and the connected compounded nodes. A directed edge object consists of tail object, middle object and head object. While context tuples represent system behaviors, processes, tasks, procedures and programs, the aggregation of the context tuples forms Context Maps. The Context Maps allows
modeling, mining and evaluating context, processes and view of information system with
generic templates and domain experts.

2.2 Context Maps Paradigm and Technology

The website www.gen-strategies.com built by Dr. Wojciech M. Jaworski contains much
useful information about Context maps including the evolution of Context maps.
Historically, this technology was initially developed as a means of recovering and
refining knowledge from legacy system. During the late 1970s and early 1980s, it was
named as Array Based Language based on conceptual graphs introduced by J. F. Sowa.
In the late 1980s, it was renamed as ABL/W4 (W4 indicates the meaning of what, when,
where and which). In the early 1990s, by considering existing notations and
methodologies, Professor Jaworski first introduced the concept of Context Maps and
named this technology as Context Maps, namely Jointed Map. Until now, Context Maps
can represent knowledge assets in 3P-able format (process-able, plug-able, and pattern-
able).

Context Maps introduces the concept of creating style sheets to control knowledge based
information access and navigation. It represents the relationship between different
information nodes in a spreadsheet by vertical columns. In a technical sense, Context
Maps describe the information set by formally declaring topics, and by linking the
relevant parts of the information set to the appropriate topics.
In view of the special property of Context Maps, it is feasible and efficient to describe and process conceptual information by using the popular concept of spreadsheet structure. And by applying the logic query tool with spreadsheet structure, the specific information that users expect to search from the map can be rapidly acquired. So Context Maps is a collection of different information connected together in a logical means and its technology is very powerful.

In practice, using this new technology, program source code can also be expressed clearly and readably by Context Maps notation in spreadsheet. In this report, some typical samples will be demonstrated.

2.3 Context Maps Syntax and Process

The syntax of Context Maps is based on the Relationship-Oriented paradigm, defined by relating Sets (concepts) and Set Members. In Context Maps, the relationships are represented by kTuples (i.e. vertical columns in map). The kTuple consists of Set, Set member and Role Tuples. This construct is the fundamental structure defined by the concepts and instances related by roles.

The relating mechanism is implemented by allocating roles to sets in schema and their instance to set components in map. Compared to diagrams, maps are very compact, and offering a rich context within limited space of a computer screen. Maps are created or edited within an organized electronic sheet (MS Excel spreadsheet) that assures efficient manipulation of relationships (columns) and heavy reuse of components (rows).
Figure 2-1 (source from http://www.gen-strategies.com/images/Inception.htm built by Dr. W.M. Jaworski) is a sample that demonstrates how to represent the state machine of “Inception” phase in Unified Process to Context Maps. In this phase, each stage can be transferred to its subsequent stage after achieving all the tasks involved in this stage. The left figure illustrates the **Context Maps View** of workflow in “inception” phase, and the right one illustrates the **Graph View** of workflow in “inception” phase.

<table>
<thead>
<tr>
<th>Context Maps View</th>
<th>Graph View</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image of Context Maps View" /></td>
<td><img src="image2.png" alt="Image of Graph View" /></td>
</tr>
</tbody>
</table>

**Figure 2-1 state machine of “Inception” phase**

Compared to **Context Maps View** with **Graph View** in the figure 2-1, Sets (Iteration Planning, Requirements, Analysis, Design, Implementation, Test, Iteration Assessment) can be represented as “Set members” in rightmost column under Set name {Workflow State}, and arrows can be represented as “Set Member Roles” in column 1-14 marked with lower case letters “i” “t” “l”. The terminology and symbol of Context Maps are
introduced in 2.3 Context Maps Terms and Constraints and 2.5 Context Maps Notation in detail.

In **Context Maps View**, the schema-level view provides information about Context Maps structure and size. This map contains some sets namely \{Event\}, \{Workflow Criteria\}, \{Workflow State\}, and \{Task\}. There are some Set Roles namely \(E'\), \(G'\) and \(L'\) and four Member Roles namely \(v'\), \(t'\), \(T'\) and \(T\). Set Role \(E'\) was allocated at \{Event\} and \{Task\}. \{Task\} allows clustering of columns with Member Role \(v'\). Set Role \(L'\) was allocated to \{Workflow State\} to allow Member Role \(t'\), \(T'\) and \(T\) allocating under it. Set Role \(G'\) was allocated to \{Workflow Criteria\}.

If users need to develop large Context Maps models, they can hide irrelevant columns and rows, editing visible cells and inserting new columns and new rows.

In general, Sets appear on the right of the map between bold curly brackets. They can be considered as a heading of a table column or a row. The Roles can be the actual contents listed in the table. Each column is to be read vertically using the syntax. For every "Value" at the spreadsheet, user can read up or down a column and across towards the right of the map to find which Role and Set Member the "Value" is referring to. The user can also obtain the schema of Context Maps by hiding set members and irrelevant columns and can get useful information by applying query tool.
2.4 Context Maps Terms and Constraints

The Context Maps terms used in this report include definitions, acronyms and abbreviations.

Figure 2-1, **Context Maps View**, will be used to explain the Context Maps terminology.

- **Context Maps**: represent the relationship between different information Sets and provide functionality of arrays, graphs, relational tables, etc.

- **JMaps**: Abbreviation of Jointed map. The previous name of Context Maps.

- **CONTEXT+**: A set of tools, which was developed for processing Context Maps.

- **Context Tuple**: A generic association of set members cast in roles. In the extended spreadsheet a column of roles and the related set members define context tuple.

- **KTuple**: Abbreviation of Knowledge Tuple. It consists of Set. Set member and Role Tuples, has its own Schema and contains Identifier, Type and Descriptors.


- **Set**: Between the bold {} in column 17, such as {Event}, {Workflow State}, and {Task}.

- **Set Member**: The members below the bold {} in the most-right column, such as “Iteration Planning”, “Requirements” and “Analysis”.

- **Set Roles**: the upper case letters in spreadsheet cells, such as letter “E”, “G”, and “L”.

- **Set Member Roles**: the lower case letters or digits in spreadsheet cells, such as “f”, “t”, and “l”.

- **Cardinality of Roles**: the column 15 counts the number of not empty Roles in every row.
- **Cardinality of Set Member**: the column 16 counts the amount of set members under each Set.

- **Atom**: Anything in a right-most column: set name, value under {Set}.

- **Spreadsheet**: An electronic page in MS Excel, it is used to store and process the data in Context Maps.

The Context Maps constraints:

The Context Maps program must work properly with Windows 95/98/NT, Win2000 and Microsoft Excel. Its development may be influenced by Microsoft Window 98 operating system implementing strategies and policies. MS Excel is constructed by spreadsheet that allows users to organize data, complete calculations, make decisions, graph data, and develop professional reports.

The syntax, schemata, maps, and styles of Context Maps are protected by copyright and trade secret law and may not be disclosed, used or produced in any manner, or for any purpose, except with written permission from Dr. Wojciech M. Jaworski.

### 2.5 Context Maps Notation

Context Maps is graphical technology for specifying, visualizing and modeling generic schemas with information systems. Context Maps notation is an essential element in this technology. It can be widely employed in many fields such as:

- Information system architecture
- Recovery and reuse of system patterns
- Evolving information systems
- Software evaluation and renewal
- Automation of system design
- Modeling of web sites and knowledge hubs
- Systems workstations

Context Maps notation can be illustrated as the following map:
Figure 2-2 The Context Maps Notation

In practice, some symbols can stand for different meanings, and for special need, different users can define by themselves. The defined regulations are flexible, however easy to understanding is more important.

2.6 Context Maps Tools

The Context Maps tools named “CONTEXT+” have been developed for a specific purpose. As custom-made tools, they can work with Context Maps properly to retrieve the useful information and generate the corresponding maps. They are becoming the indispensable parts of Context Maps. The primary functions include:

- Show Schema
- Compute Cardinality
- Apply Color
- Help
- Mode Selection
- Query
- Output Format: Map or Graph
The "Query" is the most important function in "CONTEXT+" tool. It gives user a convenient and flexible way to manipulate and control the entries of Context Maps. For its detailed usefulness, benefits, development and implementation, the following chapter will discuss in depth.
Chapter 3

CONTEXT+: Development Environment

for Context Maps

3.1 Main Goal

The application environment of Context Maps is Microsoft Excel. Microsoft Excel is a spreadsheet program that has been widely used for data storage and manipulation. It allows user easily to analyse data and mine data by using its standard toolkit. However, the Excel program is not tuned for scientific research, especially Context Maps studies. For example, data in Context Maps needs to have color applied based on different value in a big map, which could be thousands of rows and hundreds of columns, or retrieve small part of useful and meaningful data from big map and generate compounding maps, etc. The labor involved with such data exploration has become increasingly onerous. The standard toolkit in Microsoft Excel doesn’t provide functionality to meet this goal. Also there are no such tools from third-parties to provide similar functionality. In this circumstance, the challenge comes out to develop an automatic tool which allows users to manipulate data efficiently. The objectives for developing such tool are:

1. To develop a user-friendly interface which provides user with a number of options to easily and efficiently process data in big maps.

2. To apply this tool for processing some 3P-able format Maps, to calibrate the tool and hence validate its capabilities.
This report presents new tools for Context Maps, namely CONTEXT+. The program is designed using Visual Basic as an Excel macro program; it can be applied on any system with Excel 97 or higher. A user-friendly graphical interface (GUI) is utilized to guide users through application steps; the usage of this tool is simple and straightforward.

3.2 3P-able Format Introduction

With the development of Context Maps technology, Dr. Wojciech M. Jaworski [2002] introduced a new concept “3P-able”. Today, Context Maps can deal with large amounts of data and represent knowledge assets in 3P-able format. 3P-able can be expressed as the following formula:

3P-able = Plug-able + Process-able + Pattern -able

- **Plug-able**: Merge-able horizontally and vertically

Context Maps is a collection of different knowledge connected together in a logical manner. Many scattered knowledge assets, concepts, and relationship among these concepts can be processed and integrated into one Context Map, so that a formatted and cleared view is presented in front of users. Concretely, Context Maps should be able to merge separate knowledge into one view in the horizontal and vertical. The “Join Maps” of CONTEXT+ was developed according to this requirement.

- **Process-able**: Create-able, Read-able, Update-able, Delete-able

Context Maps technology originally supports this requirement. In order to speed up the efficiency of processing Maps, it is more flexible and more convenient to manipulate and view information of Context Maps with CONTEXT+ developed and employed.
• **Pattern-able**: Search-able and Navigate-able by patterns

Context Maps is a kind of notational technology. This notational pattern strongly supports search and navigation. With the use of spreadsheet structure, a large amount of data can be organized logically. It can simplify the procedure in processing Context Maps. By using the custom-make Query of CONTEXT+, it is convenient to get the specific knowledge that users expect to search from the map.

3P knowledge is represented by (vertical) Knowledge Tuples (KTuples).

- KTuple consists of Set, Set Member, and Role Tuples.
- KTuple includes its own Schema.
- KTuple contains Identifier, Type and Descriptors.

The order of KTuple elements is loss-less.

- K-Tuple is a member of one or more KViews.
- K-View is a member of one or more KHubs.
- K-Hub is a member of one or more KDomains.

### 3.3 Tool Features

The "CONTEXT+" tool mainly includes the following four functions:

1) Show Schema
2) Query
3) Compute Cardinality
4) Apply Color
“Query” is the most useful function in the “CONTEXT+” tool. It prompts a user-friendly interface, and gives users a convenient and flexible method to manipulate and control Context Maps. “Show Schema”, “Compute Cardinality” and “Apply Color” are autonomic commands which provide producing a zoom-in map i.e. work frame of Context Maps, computing the number of non-empty cells in concept rows and the number of sub-concept value under this concept, and applying different color based on the values of cells functions individually. They also appear in the Query Form, which can be combined with other options, and provide the same functionality.

In the Query Form GUI, there are four sub-sections that are “Mode”, “Query”, “Output” and “Run”. The following will describe the main features of them.

3.3.1 Query Usefulness in Context Maps

The Query Usefullness function provides user an efficient way to retrieve the useful and meaningful data from map. Normally, a signal map of Context Maps may have thousands of rows and hundreds of columns so that the whole sheet can’t be displayed within the limited space of computer screen. Therefore, manually handling the data with such big map is very time consuming work and also errors are easily generated. In this case, Query provides user a tool to automatically finding and retrieving relevant data based on different operations.

In the “Query” section of Query Form, there are four options “AND”, “XOR”, “OR”, “NOT” with another choice “By Color”.

19
First of all, "AND" allows user to select at least two cells from different rows to generate corresponding map which contains all the not empty columns within these rows.

Second, "XOR" allows user to select at least two or multiple cells from different rows to generate corresponding map, which contains the exclusive not empty columns within these rows.

Third, "OR" allows user to select at least one or multiple cells from different rows to generate corresponding map which contains the each of the not empty columns from within these rows.

Finally, "NOT" allows user to select only one cell to retrieve negation columns and generate the corresponding map.

All these operations provide different ways to efficiently retrieve a small map with useful and meaningful data from a big map. Thus user can easily get the useful information or process the map.

The following, Figure3-1, is an example for showing the Context Maps.
Figure 3-1 Real Example of Context Maps

This Figure is a small example of Context Maps. It contains around 458 rows and 56 columns (most of them are invisible in order to display on this report). Even though it is not a very big Context Maps. the whole sheet can’t be showed on the computer screen.

In order to display this, we have to group the rows and columns together. Now let’s try the Query function. If the user selects cell R66:C55 and run the “OR” operation, this process can get the relevant data with “AutoSBrightYellow” variable and save the corresponding data into a small map.
The result of running “OR” on cell R66:C55 is showing below:

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>52</td>
<td>2</td>
<td>(Context)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>map</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>(Transition Description)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>Begin sub</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>equals &quot;S&quot; finish Case to next</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>51</td>
<td>2</td>
<td>(Data Object)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td></td>
<td></td>
<td>Dim mySheet As Excel.Range</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td></td>
<td></td>
<td>Dim c As Range</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>1</td>
<td>51</td>
<td>19</td>
<td>(Color Object)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td></td>
<td></td>
<td>(Pre-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td></td>
<td></td>
<td>Case &quot;S&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td></td>
<td></td>
<td>29</td>
<td>(State)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td></td>
<td></td>
<td>1</td>
<td>Start the main program define different variable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td></td>
<td></td>
<td>2</td>
<td>Call the checkColor sub</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td></td>
<td></td>
<td>3</td>
<td>S23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td></td>
<td></td>
<td>23</td>
<td>S29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>S</td>
<td>51</td>
<td>29</td>
<td>(Action)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td></td>
<td></td>
<td>1</td>
<td>Public Sub ApplyColor_Next()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td></td>
<td></td>
<td>1</td>
<td>c.Interior.Color = AutoSBrightYellow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>S</td>
<td>51</td>
<td>104</td>
<td>(Edited VB Code)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td></td>
<td></td>
<td>1</td>
<td>S1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td></td>
<td></td>
<td>1</td>
<td>Public Sub ApplyColor_Next()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td></td>
<td></td>
<td>1</td>
<td>Dim mySheet As Excel.Range</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>169</td>
<td></td>
<td></td>
<td>1</td>
<td>Dim c As Range</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td></td>
<td></td>
<td>1</td>
<td>AutoAqua = RGB(80, 195, 238)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td></td>
<td></td>
<td>1</td>
<td>AutoN = RGB(230, 180, 183)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td></td>
<td></td>
<td>1</td>
<td>AutoF = RGB(0, 0, 0)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-2: The Result Maps after Running “OR” Operation

This result map only contains no more than hundred rows and 5 columns. And all data in this result map are relevant to the selected ‘AutoSBrightYellow’. The useful information user will get is as following:

{Pre-Condition}

Case “S”

{Action}

c.Interior.Color=AutoSBrightYellow
"By Color" is another important function in "Query" section. It is used for querying data and generating handy map with relevant data, which can provide user pre-defined information. This is made more efficient by pre-defining the customized rule to get the corresponding map. Every active sheet should have at least one predefined query in order to run this function. The predefined query is several cells with Red, Yellow, Green and Blue as background color in one column. In the predefined query, the Red means "AND" operation. Yellow means "XOR" operation, "Green" means "OR" operations and "Blue" means "NOT" operation. It also can work with standalone "AND", "XOR", "OR" and "NOT" operation, however it has to have at least two predefined query selected.

3.3.2 Mode Selection Usage

This function is an assistant function for "Query" function. Namely, it works with "Query" function to provide user more flexible and convenient way for querying data. Under "Mode" section, it allow user to select "Visible Map", "Select Sets", "Select Roles", "Select Maps" and "Join Maps".

"Visible Maps" allows user to specify whether querying data should be taken only from the visible part of the map or from the whole map, which includes hidden rows and columns. As stated before, mostly the Context Maps are too big to show on the computer screen so that it's not easy for user to process the data. Sometimes, some parts of the data are not useful for the particular operation. Then the user can hide those parts of data and let the operation work only on visible part within a map.
"Select Sets" allows the user to choose the Sets collection for querying data from the Context Maps. Sets are necessary and important roles in the Context Maps. "Select Sets" function provides a more flexible way to let the user select any part of Sets collection for querying data. All the elements under selected Sets will take part in querying data action. Others will be hidden.

"Select Roles" provides another flexible way for the user to select useful part of map for querying data. It allows the querying data faster and more efficient. Roles are basic element to consist of Context Maps. Some roles may not useful for particular operation. The user can select only necessary roles for querying data from the Context Maps.

"Select Maps" allows the user to work on multiple maps for querying data. Excel has a restriction for spreadsheet. It only allows containing 255 columns in one spreadsheet. However, the Context Maps are mostly very big in the real world so that it's hard to construct a Context Maps within only 255 columns. In term of this situation, Context Maps allow using several sheets to consist of one map. Therefore, querying data should enable user to select multiple maps to work on. All the sheets in the current active workbook are available for selection. And user should specify queried data in each selected maps before implementation.

"Join Maps" enable merging of multiple maps and saving the result into a new map. Querying will generate a small map with relevant data so that "Join Maps" can merge couple of these maps. It is a bit different from other operations in this section because it
can work with either “Select Maps” or by itself. This operation has nothing to do with querying data. When it works with “Select Maps”, it merges all the result maps from “Select Maps”. Otherwise, the user has to select maps for merging. For merging maps, it compares the Set values and Set member values for all selected maps. If they are the same, it conjuncts all the roles from the multiple maps and saves the result into a new map. If they are not the same, it adds additional rows in relevant position of the new map.

3.3.3 Apply Color Usage

Context Maps consist of Sets, Set Members, Set Roles, Set Members Roles, Cardinality of Roles, and Cardinality of Set Members etc. different elements. This functionality provides automatic way to applying different background color based on the elements usage and value. This gives the user a clear vision of Context Maps and enable user easily and efficiently find useful information from the big map.

3.3.4 Context Map Help

In this application, there are two versions of “Help” files. One is Excel version that can be triggered from Query Form. The other is MS Word documentation of user manual. The Excel version supplies brief description for “QUERY” functions. The user manual provides a detailed description about how to use this program and give some examples for some functions.

3.4 Practical Applications
Using "CONTEXT+" tool in applications is more convenient for users to compare the similarities and differences in processing Context Maps. The powerful tool "CONTEXT+" can retrieve and express the relationship of original materials clearly. By simply selecting the specific components, "CONTEXT+" tool can analyze the relationships of each component within the process. In Context Maps representation, users can easily plug in another relationship or diagram notation in one map, even different kind of processes for comparison. "CONTEXT+" tool also supports these operations.

During the development stage, some practical applications were used as tests to measure flexibility and performance of the CONTEXT+: the most relevant ones are listed here.

- "DUE DILIGENCE STRATEGY WITH CONTEXT MAPS" major report by XU GUO ZHU.
- "KNOWLEDGE MINING WITH CONTEXT MAPS, CONVERTING OPEN PROCESS METHODOLOGY TO A 3P-ABLE FORMAT" major report by JING BAI.
- Self-consistency test with the examples in this report and part of source code.

Other applications will be part of the manual for the facilities or be part of other examples available to users.
Chapter 4

CONTEXT+: Installation

4.1 Required Files

The goal of this project is to build Context Maps tool under 3p-able format, which will allow users to deal with Context Maps very efficient. The program must run on computers equipped with MS Excel. The central element in the process of information manipulation is based on the Context Maps formal notation technology.

The file ContextMaps.xls contains some samples of input and output sheets to demonstrate the functionality of CONTEXT+. In order to use the Context Maps Tool, first of all, this file has to be opened and the user has to click enable macro message box, then the user can either open a new workbook or insert new sheet to work.

The CONTEXT+ menu buttons are pretty straightforward, and the dialogs will guide users through information-entry steps. If user needs any help to run the program, there are two versions of “Help” file. One is short version of Excel file, user can click Help button from Query Form. Another one is MS Word documentation of user manual, which will be included in this report.
4.2 Installation

Any system must run Microsoft Excel 97 (Office 97) or higher under Microsoft Windows operating system. The Programming environment is the Visual Basic Editor that comes with Excel. No stand-alone VB environment is needed.

The Context Maps file could be placed into any folder. For better file organization, it is suggested that a Context Maps folder can be created and all related files can be placed in this folder.

4.3 Development Tool

- Excel

Microsoft Excel is a spreadsheet program that has been widely used for data storage, data calculation, graphic or chart display and decision-making etc.

The three major parts of Excel are:

1) Worksheets (which means the same as spreadsheet) that allow user to better calculate, manipulate, and analyze data such as numbers and text.

2) Charts, which pictorially represent data. Excel can draw a variety of two-dimensional and three-dimensional charts.

3) Databases, which manage data. For example, once users enter that data, they can search for specific data, and select data that meet the criteria.

- Visual Basic

A Visual Basic Application can provide us with the means to accomplish a wide range of the programmatic functionalities. With VBA, we can create full-fledged custom applications in Microsoft Excel.
Visual Basic supports a set of objects that correspond directly to elements in Microsoft Excel. An object in Visual Basic can represent every element in Microsoft Excel, such as workbook, worksheet, range, cell, and so on. By creating procedures that control these objects we can automate tasks in Microsoft Excel.

### 4.4 Code Design

This program is written in Visual Basic. The source code of approximately 3000 lines is stored as different .frm, .bas and .cls files and is open to the public. The code can be viewed in the Visual Basic Editor in Excel.

The whole source code was grouped by 4 modules, 3 forms and 1 class. The modules and forms are organized and named by functionality. All modules are named as "Mxxxx". All forms are named as "frmxxxx". All classes are named as "Cxxxx".

Interested readers can be able look at individual modules and forms once they have access to this program. With Professor W.M. Jaworski’s approval, the source code can be freely copied or modified in order that users can customize their own needs and add new functions.

### 4.5 General Constraints

This program is constrained only to run on Microsoft Windows operating system: Win 95/98, Win 2000 and Win NT. And also Microsoft Excel 97 or above is required to be installed. The user needs to know some basic function of Microsoft Excel.
Chapter 5

CONTEXT+: ContextMaps Documentation

Context Maps is a powerful method for representing systems architecture, structures and processes. Context Maps can incorporate instances, concepts, roles, knowledge tuples and views. It has simple semantics, which generate different views of the underlying knowledge for users. By using the Context Maps models technology, the information structure is rewritten from narratives into a knowledge frame, and create schema view of the Context Maps model. And CONTEXT+ tool gives user more convenient to retrieve and extract the relevant information from the mass and complicated diagrams.

The code written for this program consists of three Form interface, four Modules and one Class. The detail source code list will be found in the appendix of this paper.

The structure for the code is:

A. Form Interface
   
   A.1 frmProjection: this Form is used as QUERY interface for user to select main operations. It will validate and save the user’s selection.

   A.2 frmSelectConcepts: this Form is used to display all the Concept Sets from the current active ContextMap, and save the selected Concept Sets in a collection.

   A.3 frmSelectWorksheets: this Form is used to display all the Worksheets from the current active Workbook and save the selected Worksheets in a collection.
B. Modules

B.1 mApplyColor: this module has two subroutines which are ApplyColor and CheckBold. The ApplyColor is used for applying color based on the different value on the map. The CheckBold is used for changing the font to bold for the whole rows which are Concept Sets and set the gray as background color for the Concept Sets.

B.2 mAutoByColor: this module has eight subroutines. StartMultiQuery is used for validating the selection by calling ValByColorParameter and implements the "AND", "XOR", "OR", "NOT" for predefined queries by calling AutoQueryByColor. PromptWorkSheets is used for getting all the worksheets from current active Workbook for user to select. SPAAndHideCols, SPXorHideCols, SPORideCols and SPNotHideCols are called by AutoQueryByColor for predefined queries.

B.3 mMerge: this module has three subroutines as its major parts, which are Merge, JoinMaps and findConceptSet. FindConceptSet is used for getting all the Concept Sets value and their row number from the map. Merge is used to add a new sheet and rename it as "Merge Result". JoinMaps is used to merge multiple selected sheets.

B.4 mProjections: this module is the main part of whole project. It has 16 subroutines, and the key subroutine is ProjectjMap. This subroutine is used for checking the selection from the Query Form and calling the different sub or functions to implement the corresponding operations. The rest subroutines are: InstanceBasedQueryProject, InteractiveProjection,
QIBHideRows, QIBHideCols, QuerySPAndHideCols, QuerySPXorHideCols,
QuerySPOrHideCols, QuerySPNotHideCols, ShowAll, SPHideRows,
SPAndHideCols, SPOrHideCols, TestSPXorHideCols, TestSPNotHideCols.
ValidateProjectionParameters. Mostly they all called by ProjectjMap. The
relationship between them can be found through the ContextMap which is
converted by the source code of this module.

C. Classes:

C.1 CsuspendVisualInterface: this class is used for creating the instance of the
application and setting or getting the Timer or Status property for the
application.

Context Maps can be used for representing architectures, structures, and relationship for
information systems. It can also clearly demonstrate the consistency and completeness of
the VB code and CONTEXT+ user manual. In order to illustrate the usage of
ContextMap and implementation of the source code in this chapter, an efficient means
will be introduced to represent the source code and user manual: ContextMaps format
documentation. By this way, users can be able to understand the usefulness of
ContextMaps and comprehend the workflow of source code and user manual.

The work of conversion is pretty hard and challenging. As everybody knows,
constructing the schema of the ContextMap is critical and the most difficult part in the
heavy mapping field. Professor W.M. Jaworski provides a good schema as table 1
(Jaworski [2002]) and instructions so that all these processes can be preformed smoothly.
And also a group of students help me to convert part of the maps and verify the maps. It contains 53 ContextMaps for VB source code and 19 ContextMaps for CONTEXT+ user manual. All these maps are very important in this report; however, all of them can’t be showed here because of paper and space limitations. Here, only the map schema, top map and some small views will be displayed. The rest of them can be viewed on the website: http://groups.yahoo.com/group/ContextMaps/files/VBtoMAPS/.

<table>
<thead>
<tr>
<th>2</th>
<th>0 (Context)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 (Map/Code Purpose Description)</td>
</tr>
<tr>
<td></td>
<td>0 (Data Objects)</td>
</tr>
<tr>
<td></td>
<td>0 (Transition Description)</td>
</tr>
<tr>
<td></td>
<td>0 (Pre-conditions)</td>
</tr>
<tr>
<td></td>
<td>0 (States)</td>
</tr>
<tr>
<td></td>
<td>0 (SubRoutines/Sub-Maps)</td>
</tr>
<tr>
<td></td>
<td>0 (Actions)</td>
</tr>
<tr>
<td></td>
<td>0 (Post-conditions)</td>
</tr>
<tr>
<td></td>
<td>0 (Edited Source Code Statements)</td>
</tr>
<tr>
<td>2</td>
<td>0 (Source Code Statements)</td>
</tr>
<tr>
<td>2</td>
<td>1 (Content Source)</td>
</tr>
<tr>
<td>2</td>
<td>1 (Author)</td>
</tr>
<tr>
<td></td>
<td>Syntax and Patterns © by W. M. Jaworski, 1988-2002</td>
</tr>
<tr>
<td></td>
<td>Map © by &lt;map developer name&gt;</td>
</tr>
<tr>
<td></td>
<td>Map verification © by &lt;map verifier name&gt;</td>
</tr>
</tbody>
</table>

Table 1: ContextMap Schema for mapping VB code
Table 2: Top-Map Schema and part of map

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>4</th>
<th>(Context)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Public Sub Merge (joinSheets As Collection)</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td></td>
<td>CT source code</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td>1</td>
<td>Private Sub JoinMaps (joinSheets)</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td>map</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td>The code is used for adding new sheet named “Merge Result” which is used to save the result of merging multiple maps</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td>The whole code divided into 4 States in map</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td>1</td>
<td>Begin sub</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
<td>Browse to check each worksheets name under current active workbook</td>
<td></td>
</tr>
</tbody>
</table>

Syntax and Patterns © by W.M. Jaworski. 1988-2002
Map © by W.M. Jaworski

Zhou Kang

Editing by WMJ
Check whether the current worksheet exceeds the max count of all the worksheets. Check if there's any worksheet named "Merge Result" already.

If there is no worksheet named "Merge Result" in the current active workbook, add a new sheet named the "Merge Result".

```vb
O O O O 4 5 (Data Object)
  v 1 Dim WS_Count As Integer
  u 2 Dim I As Integer
  u 2 Dim rngNewSheet As Boolean
  / 2 Dim wks As Worksheet
  ü ü 2 Dim newWks As New Worksheet

G G G G 4 2 (Pre-condition)
  n 3 For I = 1 To ActiveWorkbook.Worksheets.Count
  n 2 If ActiveWorkbook.Worksheets(I).Name = "Merge Result" Then

4 3 (State)
  t 1 Start Program

4 2 If "Merge Result" sheet already exists, delete it. Create a new sheet called "Merge Result"

4 2 Call sub to start merging maps. End this program

S S S S 5 1 (SubRoutines/Sub-Maps)
  # # 3 Call JoinMaps(joinSheets)

S S S S 5 17 (Action)
  1 1 Public Sub Merge(joinSheets As Collection)
  2 1  figNewSheet = False
  1 1 sheets("Merge Result").Select
  2 1 Application.CutCopyMode = False
  3 1 Application.DisplayAlerts = False
  4 1 ActiveWindow.SelectedSheets.Delete
  5 1 Exit For
  1 1 Next I

1 6 2 sheets(joinSheets(I)).Select
  2 7 2 Set wks = ActiveSheet
  3 8 2 wks.UsedRange.SpecialCells(xlCellTypeVisible).Copy
  4 9 2 Set newWks = ActiveWorkbook.Worksheets.Add(after:=Worksheets(Worksheets.Count))
  5 # 2 newWks.Name = "Merge Result"
  6 # 2 With newWks
  7 # 2 .Paste
  8 # 2 End With

v # # 3 Call JoinMaps(joinSheets)
```
Table 3: ContextMap of Code Sub Merge

<table>
<thead>
<tr>
<th>Context</th>
<th>Transition Description</th>
<th>Data Object</th>
<th>Pre-condition</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Sub InteractiveProjection(ByRef wks As Worksheet, ByRef objSelected As Object)</td>
<td>go to the the program</td>
<td>Dim frmProjectionOptions As FProjection</td>
<td>On Error Resume Next</td>
<td>begin the function</td>
</tr>
<tr>
<td>Public Function Project</td>
<td>Map(ByRef wks As Worksheet, ByRef obj As Object, Optional ByVal SaveView As Boolean = False, Optional Optional ByVal CheckVisible As Boolean = False, Optional ByVal QueryColors As Boolean = False, Optional ByVal AutoByColors As ByVal PasteToNewSheet As Boolean = False, Optional ByVal PromptForConcepts As Boolean = False, _Optional ByVal JoinMaps As Boolean = False, Optional ByVal getCardinality As Boolean = False, Optional ByVal ApplyingColor As Optional ByVal ProjectionType As Long = pjtNORMAL, Optional ByVal SuppressUserInteraction As Boolean = False) As Boolean</td>
<td>there is error system is resumed</td>
<td>If Not .Canceled Then</td>
<td>if no error continue below functions</td>
</tr>
<tr>
<td>map</td>
<td>Execute the if clause actions</td>
<td></td>
<td></td>
<td>if user don't want to cancel the execution then continue below functions</td>
</tr>
</tbody>
</table>
Table 4: ContextMap of Code Sub InteractiveProjection

<table>
<thead>
<tr>
<th>Context</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 4</td>
<td>Public Function PromptWorkSheets(ByRef colSelectedSheets As Collection) As Boolean</td>
</tr>
<tr>
<td>9</td>
<td>CT source code</td>
</tr>
<tr>
<td>9</td>
<td>properties(SelectWorksheet)</td>
</tr>
<tr>
<td>9</td>
<td>map</td>
</tr>
<tr>
<td>8 2</td>
<td>(Map/code Purpose Description)</td>
</tr>
<tr>
<td>1</td>
<td>This map is used to represent the source code for selecting all the active worksheets in the current workbook</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(Transition Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>
Dim colAllSheets As Collection

**Pre-condition**

OnError Resume Next

For nSheetIndex = 1 To .Sheets.Count

If Not SelectWorkSheet(frmSelectWorksheet.listSheets, frmSelectWorksheet.canceled, frmSelectWorksheet.SelectedSheets) Then

Begin function and define the variable used in this program

Initialize some variable

Browse every worksheets in the current workbook, add their name to collection

Get the selected maps from the available maps list

End Function

Resume Next

**SubRoutines/Sub-Maps**

If Not SelectWorkSheet(frmSelectWorksheet.listSheets, frmSelectWorksheet.canceled, frmSelectWorksheet.SelectedSheets) Then

Public Function PromptWorkSheets(ByRef colSelectedSheets As Collection) As Boolean

Set colAllSheets = New Collection

With ActiveWorkbook

For nSheetIndex = 1 To .Sheets.Count

    colAllSheets.Add .Sheets.Item(nSheetIndex).Name

Next nSheetIndex

End With

Exit Function

PromptWorkSheets = True

**Comment**

**Content Source**

Zhou Kang

**Author**

Syntax and Patterns © by W.M. Jaworski, 1988-2002

Map © by ZhouKang

Map verification © by Yang Shentian

---

Table 5: ContextMap of Code Sub PromptWorkSheets
Table 6: ContextMap of Code Sub CheckBold
Table 7: ContextMap of User Manual “CONTEXT+ Toolbar Menu”

| 12  | 1   | (cTuple Id) | CONTEXT+ | Show Schema | Query | Prompt the Query Form
|-----|-----|-------------|----------|------------|-------|----------------------
|     | 9   | # | # | # | 12 | Id | Show Schema | Query | Prompt the Query Form |
|     | 9   | 12 | Id | Show Schema | Query | Prompt the Query Form |
| 12  | 3   | (Context) | Compute Card | ApplyColor | Function | Get the Help |
|     | 12 | 1 | OO Templates | ApplyColor | Function | Get the Help |
| 2   | Processes | 10 | Products | Get the Help | |
| 1   | Project Process | 1 | Phase Process | 1 | ID: Interaction Diagram | |
| v   | v   | 1 | v | 12 | v | 12 | 1 (OO Template) | 1 | Project Process | 1 | Phase Process | 1 | ID: Interaction Diagram | |
| XX | X  | 12 | 0 (cTuple Type) | 12 | 1 | (cTuple Id) | 12 | Id | Show Schema | Query | Prompt the Query Form |
| 8  | 9  | 10 | 11 | 12 | 12 | Id | Show Schema | Query | Prompt the Query Form |

Table 8: ContextMap of User Manual “CONTEXT+ Query Form”

| 12  | 1   | (cTuple Id) | CONTEXT+ | Show Schema | Query | Prompt the Query Form |
|-----|-----|-------------|----------|------------|-------|----------------------
|     | 9   | # | # | # | 12 | Id | Show Schema | Query | Prompt the Query Form |
|     | 9   | 12 | Id | Show Schema | Query | Prompt the Query Form |
| 12  | 3   | (Context) | Compute Card | ApplyColor | Function | Get the Help |
|     | 12 | 1 | OO Templates | ApplyColor | Function | Get the Help |
| 2   | Processes | 10 | Products | Get the Help | |
| 1   | Project Process | 1 | Phase Process | 1 | ID: Interaction Diagram | |
| v   | v   | 1 | v | 12 | v | 12 | 1 (OO Template) | 1 | Project Process | 1 | Phase Process | 1 | ID: Interaction Diagram | |
| XX | X  | 12 | 0 (cTuple Type) | 12 | 1 | (cTuple Id) | 12 | Id | Show Schema | Query | Prompt the Query Form |
| 8  | 9  | 10 | 11 | 12 | 12 | Id | Show Schema | Query | Prompt the Query Form |

![CONTEXT+ Query Form Diagram](image)

<table>
<thead>
<tr>
<th>Mode:</th>
<th>Visible Map:</th>
<th>Select Color:</th>
<th>Select Roles:</th>
<th>Select Maps:</th>
<th>Join Maps:</th>
<th>By Color</th>
<th>Map</th>
<th>Graph</th>
<th>XOR</th>
<th>Map &amp; Graph</th>
<th>Apply Color</th>
<th>Add Atom</th>
<th>Help</th>
<th>Display only Visible Maps</th>
<th>Perform Predefined Query</th>
<th>Perform Select Sets</th>
<th>Perform ApplyColor Function</th>
<th>Perform Select Maps</th>
<th>Perform JoinMaps Using:</th>
<th>Perform Four Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v</td>
<td></td>
<td>CS: Class Specification</td>
<td>CCCD: Class-Centric Class Diagram</td>
<td>STD: State Transition</td>
<td>ID: Interaction Diagram</td>
<td></td>
<td>Perform Four Operations</td>
<td></td>
</tr>
<tr>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Perform JoinMaps Using:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>12</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v</td>
<td></td>
<td>Perform Four Operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>12</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v</td>
<td></td>
<td>Perform JoinMaps Using:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>6</td>
<td>0</td>
<td>(Event)</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Perform JoinMaps Using:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v</td>
<td></td>
<td>Perform JoinMaps Using:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v</td>
<td></td>
<td>Perform JoinMaps Using:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>v</td>
<td></td>
<td>Perform JoinMaps Using:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

49
Chapter 6

Conclusion and Recommendation

6.1 General Conclusions

After studying this notation technology, many advantages of the Context Maps are displayed. The following conclusions are drawn from the study of this major report:

1. Context maps enable us to create virtual information maps for the knowledge-based system. Context Maps are a notational technology for representing systems architecture, structures, processes and reusable templates. The Context Maps notation allows easy recovery and modeling of generic schema for processes, objects and views of information systems.

2. Although Context maps already have a set of predefined notation, they also allow the user to create their extendable notations to meet new needs. In the meantime, there are always many different ways of representing the same knowledge. This property shows the flexibility of Context maps.

3. Context Maps syntax is simple and robust. Context Maps models are 3P-able format, allow users to specify, query and control the model views. Different views are generated logically to be useful for compilers or end users.

4. By applying the CONTEXT+ tool, Contexts Maps are more efficient and convenient for users to retrieve and compare relevant information from the mass and complicated diagrams.
5. Using advanced query in CONTEXT+query tools, query on the high-level design schema is becoming possible. And directly querying the data allows query more easy and visible.

6. If the data of Context Maps reach the limit of each sheet in Excel, it can be stored in another sheet to extend size of Context Maps. And the CONTEXT+ tool has such capability to process multiple sheets.

7. The CONTEXT+ program can also be treated as standard knowledge based information to be converted into Context Maps. The structure of CONTEXT+ program becomes normalized.

6.2 Recommendations for Future Works

After studying Context Maps technology, further work is required to improve this technology and application program. The following are recommended for the future improvement:

1. When processing a large Context Maps sheet, it takes much time to get results by using the query tools. It is necessary to modify the program of CONTEXT+ tools to speed up the query.

2. Some functions of CONTEXT+ will be developed or improved. In output section, Output Graph will be done. This feature will be very important for Context Maps.

3. The tools of Context Maps should be expensed to be more convenient and intelligent to support the 3P-able format. Design a more friendly user interface needs to be done in the future.
4. Inputting the data in the map is very boring and there are some redundancies existing. Provide a guideline for designing the schema, composing the map and inputting the data.

5. Transferring the knowledge to the map is still done by manually. Maybe there are some automatic transferring tool can be developed in the near future.

6. Refine the ambiguity concepts, elements or set members within Context Map; then, building a kind of reliable criteria guides further work.
Bibliography

A. Printed Materials


B. Online Resources


2) Context Maps.

3) IHMC Concept Map Software,
   http://cmap.coginst.uwf.edu/download cmapChoice.mgi


2) Concordia University, Thesis preparation and thesis examination regulations,
   http://www-gradstudies.concordia.ca/SGS_WWW/publications.html
Appendix A---- CONTEXT+: User Manual

The CONTEXT+ application is written using VB language from Microsoft Office technologies, allowing it to run on the Microsoft platform. Make sure you check the system requirements (please refer to Chapter 4.2 Installation) before you run the application. The following will walk you through how to use this tool.

A.1 Start Program

1. Opening of the ContextTools.xls file will show the following popup dialog:

![MS Excel Standard Dialog Box]

Figure A-1: MS Excel Standard Dialog Box

2. Clicking of the Enable Macros button will launch the tool and display new menu named CONTEXT+ in EXCEL toolbar.

3. As shown in Figure A-2, five sub-menus appear when user clicks the CONTEXT+.
Figure A-2: Sub-Menus of the CONTEXT+ Menu.

Note: some of sub-menus functions will also appear on the RUN part of the Query form shown in Figure A-8.

A.2 CONTEXT+ Tool Functionality

A.2.1 Show Schema

This button will produce Zoom-in Map. which is a work frame of Context Map.

A.2.2 Computer Cardinality

1. Using this function, the criteria for this button is there should be two columns reserved or added just preceding the Concept column for computer cardinality.

2. In Context Map, the right column added is used to compute the number of not empty roles in this row and left one is used to compute the number of sub-concept value under this concept.
3. After the option button \( \text{Compute Cardinality} \) is selected from Query Form (Figure A-8) or \( \text{Compute Cardinality} \) is clicked from sub-menu bar in Context+, the program will implement computing function to position the results at these two columns.

![Diagram](image)

**Figure A-3: Syntax for Cardinality**

### A.2.3 Apply Color

The Apply Color function is applying different color in each cell for the spreadsheet based on the value of the cells. The criteria is following:

1. First, the font of the whole row is bolded and the light gray color is used as background color only if this row has concept value.
2. Second, two columns before Concept is used for Cardinality function, and the number is colored Red.

3. The rest of the Context Maps will apply different color at the background of the cells based on the cell’s value.

4. The detailed color code index is as following:

<table>
<thead>
<tr>
<th>Color</th>
<th>ColorIndex</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColorIndex = 3</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 7</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 38</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 46</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 45</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 44</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 40</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 12</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 43</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 6</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 36</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 10</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 50</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 4</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 35</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 14</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 42</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 8</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 34</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 47</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 41</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 33</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 37</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 47</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 13</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 54</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 39</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 16</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 48</td>
<td></td>
</tr>
<tr>
<td>ColorIndex = 15</td>
<td></td>
</tr>
</tbody>
</table>

Figure A-4: The Color Index
5. The following lists the different color applied on different value of cells.

"A. I. o" --- Dark Gray

"v. u" --- Light Gray

"V" --- Bright Green

"X. x, F" --- Light Purple

"E" --- Lime

"b" --- Violet

"O" --- Object

"R, m" --- Aqua

"c" --- Green

"d" --- Plum

"Y" --- Light Turquoise

"f" --- Red

"L" --- Pink

"G" --- Dark Green

"N" --- Rose

"S" --- Yellow

"t" --- Pale Blue

6. After the option button is selected from Query Form (Figure A-8) or is clicked from sub-menu bar in CONTEXT+, "Apply Color" function will apply different color in the map. Figure A-5 is Input Test Map for Apply Color Function, and Figure A-6 is Output Result Map for Apply Color Function.
Figure A-5: Input Test Map for Apply Color Function

![Input Test Map](image)

**Figure A-6: Output Result Map for Apply Color Function**

![Output Result Map](image)
A.2.4 Help

When this option button \( \text{Help} \) is clicked, the short version “Help” of Excel sheet will be displayed. This short version “Help” has brief description for each button on the Query Form.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible Map</td>
<td>The button selects the MAP space for Query operation. ON: only visible MAP is searched. OFF: whole MAP is searched, all the hidden rows and columns will display.</td>
</tr>
<tr>
<td>Selected Sets</td>
<td>The button selects the SETS space for Query operation. ON: all the Set values on the active sheet will be selected and put into another form to let user select any of them. OFF: all SET values on the active sheet is searched.</td>
</tr>
<tr>
<td>Selected Roles</td>
<td>The button selects the ROLES space for Query operation. ON: only selected ROLES is searched. OFF: all ROLES is searched.</td>
</tr>
<tr>
<td>Selected Maps</td>
<td>The button implements predefined query on multiple sheets. ON: all the available Maps are listed to let user select any of them. OFF: only active MAP is searched.</td>
</tr>
<tr>
<td>Join Maps</td>
<td>The button implements merging multiple maps into one map. ON: list all the available maps; and selected MAPS is merged. OFF: no maps will be merged.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{XOR} )</td>
<td>Implements disjunction operation based on the value of the columns related to the selected cells in the current active. Implement exclusion operation based on the value of the columns related to the selected cells in the current active.</td>
</tr>
<tr>
<td>( \text{XOR} )</td>
<td>Implements conjunction operation based on the value of the columns related to the selected cells in the current active. Implements negation operation based on the value of the columns related to the selected cells in the current active.</td>
</tr>
<tr>
<td>By Color</td>
<td>Implements ‘AND’, ‘XOR’, ‘OR’, ‘NOT’ operation based on predefined query. By default, one of the NOT, AND, OR, XOR is selected when Query Form shows up. The operation will perform relevant process on the columns related to the selected cells if ( \text{By Color} ) is not selected.</td>
</tr>
<tr>
<td>By Color</td>
<td>Operation only works when one or multiple predefined queries were selected. Under predefined queries, the AND operation is marked as Red color, XOR operation is marked as yellow color. OR operation is marked as Green color while NOT operation is marked as Blue color.</td>
</tr>
</tbody>
</table>

Figure A-7: Excel Version of Help Sheet
A.2.5 Query

1. After clicking Query from sub-menu in CONTEXT+, an interactive GUI Form will be popped up. Figure A-8.

2. In this form, there are four groups of buttons which can implement different functionalities.

3. The **Query** Section is the main and most important part. The **Mode** Section is used as input criteria for Query Section. The **Output** Section is the output format for the result of the Query Section. And **Run** Section has several independent functions which is as same as sub-menu in CONTEXT+.

![Figure A-8: Query Form for CONTEXT+ Menu](image)

A.2.5.1 Query Section Functionality

1. AND, XOR, OR and NOT operations:
- These four buttons perform operations according to different colors. Figure 5-10 is a Test Input Sheet.

- AND and XOR operations allow 2 - n rows selected by holding CTRL key. OR operation allows selected 1 - n rows. NOT operation allows only single row. Otherwise it will display error message.

- The restriction of these functions is that the selected cells must have value.

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>12</td>
<td>0</td>
<td>(Note)</td>
</tr>
<tr>
<td>22</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>6</td>
<td>0</td>
<td>(Event)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>(Phase Transition Criteria)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>(Review Criteria)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>L</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td>(Phase/State)</td>
</tr>
<tr>
<td>26</td>
<td>S</td>
<td>L</td>
<td>2</td>
<td>0</td>
<td>(Activity/State)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>S</td>
<td>1</td>
<td>0</td>
<td>(SubActivity/Step)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>A</td>
<td>A</td>
<td>2</td>
<td>0</td>
<td>(System)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>(Problem Statement Section)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>F</td>
<td>1</td>
<td>0</td>
<td>(System Specification Text)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>M</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>4</td>
<td>0</td>
<td>(Use Case Name)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>L</td>
<td>A</td>
<td>A</td>
<td>L</td>
<td>7</td>
<td>0</td>
<td>(Category)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>2</td>
<td>0</td>
<td>(Class)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>1</td>
<td>0</td>
<td>(Method)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>M</td>
<td>1</td>
<td>0</td>
<td>(Attribute)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>3</td>
<td>0</td>
<td>(Timing Constraint)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>4</td>
<td>0</td>
<td>(Pre-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>(Exception)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>L</td>
<td>1</td>
<td>0</td>
<td>(Scenario/State)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0</td>
<td>(Action)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>S</td>
<td>1</td>
<td>0</td>
<td>(Software Reaction)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>G</td>
<td>G</td>
<td>2</td>
<td>0</td>
<td>(Post-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>3</td>
<td>0</td>
<td>(Multiplicity)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A-10: Test Input Sheet for AND, XOR, OR, NOT Operations

**AND**

Implementing conjunction operation based on the values of the columns related to the selected cells in the current active sheet.

1. Tag the 2-n selected cells by holding CTRL key.
2. Check the "AND" button in Figure A-9.

3. Click "OK" button.

The result map, Figure A-11 after implementing AND operation on Figure A-10, will be displayed.

**XOR** ------ Implementing exclusion operation based on the value of the columns related to the selected cells in the current active sheet.

1. Tag the 2-n selected cells by holding CTRL key.

2. Check the "XOR" button in Figure A-9.

3. Click "OK" button.

The result map Figure A-12 after implementing XOR operation on Figure A-10 will be displayed.

<table>
<thead>
<tr>
<th>3</th>
<th>7</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>12</td>
</tr>
<tr>
<td>22</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>6</td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>26</td>
<td>A</td>
<td>2</td>
<td>0</td>
<td>(System)</td>
</tr>
<tr>
<td>29</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>(Problem Statement Section)</td>
</tr>
<tr>
<td>30</td>
<td>F</td>
<td>1</td>
<td>0</td>
<td>(System Specification Text)</td>
</tr>
<tr>
<td>31</td>
<td>M</td>
<td>A</td>
<td>A</td>
<td>4</td>
</tr>
<tr>
<td>32</td>
<td>M</td>
<td>L</td>
<td>L</td>
<td>7</td>
</tr>
<tr>
<td>33</td>
<td>M</td>
<td>L</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>M</td>
<td>2</td>
<td>0</td>
<td>(Method)</td>
</tr>
<tr>
<td>36</td>
<td>A</td>
<td>A</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>G</td>
<td>G</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>M</td>
<td>M</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>51</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>52</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>53</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>54</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
</tr>
</tbody>
</table>

Figure A-11: Result for AND Operation
<table>
<thead>
<tr>
<th>4</th>
<th>5</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>12</td>
<td>0</td>
<td>(cTuple Type)</td>
</tr>
<tr>
<td>21</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>12</td>
<td>0</td>
<td>(Note)</td>
</tr>
<tr>
<td>22</td>
<td>E</td>
<td>0</td>
<td>(Event)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>A</td>
<td>2</td>
<td>(System)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>A</td>
<td>4</td>
<td>(Use Case Name)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>F</td>
<td>F</td>
<td>A</td>
<td>A</td>
<td>7</td>
<td>0</td>
<td>(Category)</td>
</tr>
<tr>
<td>33</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>6</td>
<td>0</td>
<td>(Class)</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>A</td>
<td>3</td>
<td>(Timing Constraint)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>G</td>
<td>4</td>
<td>(Pre-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>G</td>
<td>1</td>
<td>(Exception)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>L</td>
<td>1</td>
<td>(Scenario/State)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>S</td>
<td>2</td>
<td>(Action)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>S</td>
<td>1</td>
<td>(Software Reaction)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>G</td>
<td>2</td>
<td>(Post-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>3</td>
<td>0</td>
<td>(Multiplicity)</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>R</td>
<td>4</td>
<td>(Product)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>51</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure A-12: Result For XOR Operation

**OR** ----- Implementing disjunction operation based on the value of the columns related to the selected cells in the current active sheet.

1. Tag the 1-n selected cells by holding CTRL key.
2. Check the “OR” button in Figure A-9.
3. Click “OK” button.

The result map, Figure A-13 after implementing OR operation on Figure A-10, will be displayed.

<table>
<thead>
<tr>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>E</td>
<td>6</td>
<td>0</td>
<td>(Event)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td>(Phase/State)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>A</td>
<td>A</td>
<td>7</td>
<td>0</td>
<td>(System)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>A</td>
<td>1</td>
<td>0</td>
<td>(Problem Statement Section)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>F</td>
<td>1</td>
<td>0</td>
<td>(System Specification Text)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>M</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>4</td>
<td>0</td>
<td>(Use Case Name)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>L</td>
<td>A</td>
<td>A</td>
<td>7</td>
<td>0</td>
<td>(Category)</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>M</td>
<td>M</td>
<td>F</td>
<td>F</td>
<td>L</td>
<td>6</td>
<td>0</td>
<td>(Class)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>M</td>
<td>2</td>
<td>0</td>
<td>(Method)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>3</td>
<td>0</td>
<td>(Timing Constraint)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>G</td>
<td>G</td>
<td>G</td>
<td>4</td>
<td>0</td>
<td>(Pre-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>(Exception)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>L</td>
<td>1</td>
<td>0</td>
<td>(Scenario/State)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>S</td>
<td>2</td>
<td>0</td>
<td>(Action)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>S</td>
<td>1</td>
<td>0</td>
<td>(Software Reaction)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>G</td>
<td>2</td>
<td>0</td>
<td>(Post-condition)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>3</td>
<td>0</td>
<td>(Multiplicity)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>3</td>
<td>0</td>
<td>(Message)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>R</td>
<td>4</td>
<td>0</td>
<td>(Product)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td>(Figure)</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td>(Table)</td>
<td></td>
</tr>
</tbody>
</table>

Figure A-13: Result For OR Operation
Implementing negation operation based on the value of the columns related to the selected cell in the current active sheet.

1. Tag the only single cell.

2. Check the "NOT" button in Figure A-9.

3. Click "OK" button.

The result map Figure A-14 after implementing NOT operation on Figure A-10 will be displayed.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>9</th>
<th>11</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>6</td>
<td>0</td>
<td>(Event)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td></td>
<td>(Phase Transition Criteria)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td></td>
<td>(Review Criteria)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>L</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>0</td>
<td>(Phase/State)</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>S</td>
<td>L</td>
<td>2</td>
<td>0</td>
<td>(Activity/State)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>S</td>
<td>1</td>
<td>0</td>
<td>(SubActivity/Step)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>2</td>
<td>0</td>
<td>(System)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td>1</td>
<td>0</td>
<td>(Problem Statement Section)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td>1</td>
<td>0</td>
<td>(System Specification Text)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>A</td>
<td>4</td>
<td>0</td>
<td>(Use Case Name)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>7</td>
<td>0</td>
<td>(Category)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>F</td>
<td>6</td>
<td>0</td>
<td>(Class)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>M</td>
<td>2</td>
<td>0</td>
<td>(Method)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>M</td>
<td>1</td>
<td>0</td>
<td>(Attribute)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>A</td>
<td>3</td>
<td>0</td>
<td>(Timing Constraint)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>G</td>
<td>G</td>
<td>4</td>
<td>0</td>
<td>(Pre-condition)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>G</td>
<td>1</td>
<td>0</td>
<td>(Exception)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>L</td>
<td>1</td>
<td>0</td>
<td>(Scenario/State)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>S</td>
<td>S</td>
<td>2</td>
<td>0</td>
<td>(Action)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A-14: Result for NOT Operation

Note: AND, XOR and OR will hide the whole empty rows, however NOT won't.

The NOT operation will hide the columns with values. For the example shows above, the whole row 32 is empty.
2. **BY COLOR** operation:

- This button will be implemented with one of ‘AND’, ‘XOR’, ‘OR’, ‘NOT’ operations based on selected predefined - query.

- The predefined - query should be marked in three places in the Context Maps.
  1) Concept Value cells under Concept Set column, as showing Predefined Query (1) in the Figure A-15.
  2) Cells with “q” value relevant to the Concept Value columns with predefined queries, as showing Predefined Query (2) in the Figure A-15.
  3) Tagged cells with color background. [ ] stands for **AND** operation.

Yellow stands for **XOR** operation. [ ] stands for **OR** operation and [ ] stands for **NOT** operation, as showing Predefined Query (3) in the Figure A-15. And all these color are optional.

- Select at least one predefined query as described above. If only one predefined query is selected, this function will ignore the selected ‘AND’ or ‘XOR’ or ‘OR’ or ‘NOT’ operations from the QUERY Form, and only implement the predefined query. If multiple predefined queries are selected, the function will implement each predefined query first, then it will merge the result based on the selected ‘AND’ or ‘XOR’ or ‘OR’ or ‘NOT’ operations. Detailed implementation as following:
  1) Predefined queries in the Context Map
  2) Tagged the cells for the queries to implement.
  3) Click one of the ‘AND’, ‘XOR’, ‘OR’, ‘NOT’ operations.
  4) Click ‘By Color’ button.
5) Click ‘OK’.

Figure A-15: Input Test Sheet for Query By Color

- After predefined, there is a restriction when performing the BY COLOR operation. In each predefined-query column, only one colored cell can be allowed to tag. Otherwise, a selection error message will be displayed.

- For example, in Figure A-15, the column 1, 2, 9, 11 are four predefined queries. If two predefined queries R6:C11 and R8:C9 are tagged, first the function will process predefined query R6:C11: (b) WHERE Q1 = NOT v5; s1 XOR s2; s3 OR a7; a4 AND a5, the result will be saved in the memory and showing as following Figure A-16.
Figure A-16: Temporary Result Sheet For Implementing Predefined Query Q1 of "By Color" Function

Second the function will process the second predefined query R8:C9: (d)

WHERE Q3 = NOT v1; NOT v5; s2 XOR s3; a3 OR a4; a7 AND a8. The result will be saved in the memory also and shown as in Figure A-17.

Figure A-17: Temporary Result Sheet For Implementing Predefined Query Q3 of "By Color" Function
Finally, if AND operation is selected, it will process Q1 AND Q3, the result will be as following Figure A-18.

<table>
<thead>
<tr>
<th>4</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>(New Query Setup)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>q</td>
<td>q</td>
<td></td>
<td>(a) WHERE Q1 = NOT v5, s1 XOR s2, s3 OR a7, a4 AND a5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>cView</td>
<td>(b) WHERE Q1 = NOT v5, s1 XOR s2, s3 OR a7, a4 AND a5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>q</td>
<td>q</td>
<td></td>
<td>(c) WHERE Q3 = NOT v1, NOT v5, s2 XOR s3, a3 OR a4, a7 AND a8</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>10</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>11</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>12</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>13</td>
<td>L</td>
<td>l</td>
<td>l</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>15</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>16</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>17</td>
<td>s1</td>
<td>s2</td>
<td>s3</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>21</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>23</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>24</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
</tbody>
</table>

Figure A-18: Output Result Sheet for Q1 AND Q3.

If the NOT operation is selected, it will process the opposite way to AND operation. It will display all the columns not showing in Q1 and Q3. The result is shown in Figure A-19.

<table>
<thead>
<tr>
<th>4</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>(New Query Setup)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>q</td>
<td>q</td>
<td></td>
<td>(a) WHERE Q1 = NOT v5, s1 XOR s2, s3 OR a7, a4 AND a5</td>
</tr>
<tr>
<td>6</td>
<td>q</td>
<td>q</td>
<td></td>
<td>(c) WHERE Q3 = NOT v1, NOT v5, s2 XOR s3, a3 OR a4, a7 AND a8</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>9</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>10</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>11</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>l</td>
<td>l</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>t</td>
<td>f</td>
<td>t</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>16</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
<tr>
<td>17</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>m</td>
<td>m</td>
<td>m</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>M</td>
<td>N</td>
<td>N</td>
<td>12</td>
</tr>
<tr>
<td>21</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>4</td>
</tr>
<tr>
<td>25</td>
<td>f</td>
<td>f</td>
<td>f</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>27</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
</tbody>
</table>

Figure A-19: Output Result Sheet for Q1 NOT Q3
If XOR operation is selected, the result will be shown in Figure A-20.

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If OR operation is selected, the result will be shown in Figure A-21.

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A-20: Output Result Sheet for Q1 XOR Q3

Figure A-21: Output Result Sheet for Q1 OR Q3
A.2.5.2 Mode Section Functionality

Visible Maps------This button has to work with Query Section and display the map without hidden part.

1. Check the Query Section.
2. If Visible box is selected, the result map will only display visible part of current active sheet.
3. Otherwise, the result map will display all the hidden rows and columns.

Select Sets ------ This function will automatically list the all Set values the active sheet has, users can check all or some of these Set values they would like. Figure A-22 is a sample of Select Sets Form.

1. This button should be combined with Query Section.
   (a) Check the Query Section, click the Select Sets button, click the "OK" button from Query Form.
   (b) The user will see the prompt form as Figure A-22, and then the user can select all or some sets.
   (c) After clicking "OK" button, the result map will only remain the selected sets part of the map after implementation.

2. There is an exception when it works with `By Color` function. While the `By Color` button is selected and the `Select Sets` is not selected, it will automatically perform the Select Sets function without displaying the Select Sets Form. Under this condition, it checks the sets with light gray
background colors in concept column and only remains these sets in the result map.

![Select Sets Form](image)

Figure A-22: Select Sets Form

**Select Maps**

---

This function will list all the available worksheets for the current workbook and user can check all or some of these worksheets to perform "By Color" operation simultaneously. Figure A-23 is a sample of WorkSheets List Form.

1. This button should work with "By Color" function.
2. The criterion for this button is that predefined query should be selected before processing this function.
3. When "Select Maps" button is selected, the following form will pop up.
4. After selecting worksheets, Excel will automatically open several new sheets corresponding to the number of sheets user selected under the same workbook and rename the new sheets as original sheets name plus “Result” suffix respectively. If there are already the same name sheets in the current active workbook, it will prompt a message to ask whether user would like to delete the old sheets. It should always click YES because Excel won’t allow two same name sheets existing in one workbook.

5. The following figures will demonstrate ‘Select Maps’ function. From Query Form, click “Select Maps”, “By Color” and “AND” buttons, then click “OK” button. It pops up a Worksheets List Form. Figure A-24 and Figure A-25 are two input worksheets selected from Worksheets List Form. After implementation, two new worksheets Figure A-26 and Figure A-27 are automatically added to save the corresponding result.
Figure A-24: Input Sheet1 For 'Select Maps' Function

Figure A-25: Input Sheet2 For 'Select Maps' Function
Figure A-26: Result For Input Sheet1 after Implementing 'Select Maps' Function

<table>
<thead>
<tr>
<th>6</th>
<th>8</th>
<th>9</th>
<th>11</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>q</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>q</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
</tr>
<tr>
<td>16</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>9</td>
</tr>
<tr>
<td>17</td>
<td>f</td>
<td>5</td>
<td>s1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>f</td>
<td></td>
<td>1</td>
<td>s3</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>12</td>
</tr>
<tr>
<td>21</td>
<td>m</td>
<td>4</td>
<td>a1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>m</td>
<td></td>
<td>4</td>
<td>a3</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>m</td>
<td>4</td>
<td>a5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>m</td>
<td>4</td>
<td>a7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>30</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
</tr>
</tbody>
</table>

Figure A-27: Result For Input Sheet2 after Implementing 'Select Maps' Function

<table>
<thead>
<tr>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>8</th>
<th>10</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>q</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td></td>
<td>12</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td></td>
<td>3</td>
<td>v1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>6</td>
<td>v2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>9</td>
<td>v3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
<td>v4</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
<td>v6</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>f</td>
<td>5</td>
<td>s1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>t</td>
<td>f</td>
<td>f</td>
<td>t</td>
<td>8</td>
<td>s2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>f</td>
<td></td>
<td>1</td>
<td>s3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>t</td>
<td>f</td>
<td>m</td>
<td>4</td>
<td>a1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>t</td>
<td>f</td>
<td></td>
<td>4</td>
<td>a2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>f</td>
<td>t</td>
<td>m</td>
<td>4</td>
<td>a3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>t</td>
<td>f</td>
<td></td>
<td>4</td>
<td>a4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>t</td>
<td>f</td>
<td>m</td>
<td>4</td>
<td>a5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>t</td>
<td>f</td>
<td></td>
<td>4</td>
<td>a7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>f</td>
<td></td>
<td>1</td>
<td>a8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>v</td>
<td>12</td>
<td>Syntax and Patterns © by W.M. Jaworski, 1988-2000</td>
</tr>
</tbody>
</table>
**Join Maps**

List all the available worksheets for the current workbook and choose some worksheet to merge. The criterion for this function is: comparing the values of concept column within each selected worksheet, join the columns with the same values of concept. Or add addition rows with different values of concept. Then save the result into separate sheet named “Merge Result”. This function can implement independently.

1. In Query Form, click the Join Maps box.
2. Click “OK” button. It pops up a Worksheets List Form as Figure A-23.
3. Select worksheets, which should be merged from Worksheets List Form.
4. Click “OK” button in Worksheets List Form.
5. Produce a new worksheet named “Merge Result” to save the result.

For example, Figure A-28, Figure A-29 and Figure A-30 are three input worksheets for Joining Maps. After implementation, an output worksheet, Figure A-31, will be automatically produced to save the corresponding result.

Notes: Look at all the sub-concept values under Concept {cView} in the figures.

![Figure A-28: Input Sheet Named “Merge1” for Joining Maps](image)
Figure A-29: Input Sheet Named “Merge3” for Joining Maps

Figure A-30: Input Sheet Named “Merge2” for Joining Maps
Figure A-31: Output Sheet Named “Merge Result” for Joining Maps

This function “Joining Maps” can also work with “By Color” function while “Select Maps” is selected. Under this circumstance, “Select Maps” function will implement first, then “By Color” function, finally it will join the results and save into a new sheet Named “Merge Result”. There is an example.

1. Predefine query on Figure A-24 Input Sheet1 and Figure A-25 Input Sheet2.

2. From Query Form, buttons “Select Maps”, “Join Maps”, “By Color” and “AND” are selected.

3. Click “OK” button, pops up WorkSheets List Form.

4. Select “Sheet1” and “Sheet2”, then click “OK” button in WorkSheets List Form.
5. Automatically add the two new worksheets ‘Sheet1 Result’ and "Sheet2 Result" like Figure A-26 and Figure A-27.

6. Finally produce a new worksheet named "Merge Result", Figure A-32, after joining "Sheet1 Result" and "Sheet2 Result".

|     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |
|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1   | A | A | A | A | A | A | A | A | A | A  | A  | A  | A  | A  | A  | [New Query Setup] |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| 2   | q | q |   |   |   |   |   |   |   | (b) WHERE Q1 = NOT v5, s1 XOR s2, s3 OR a7, a4 AND a5 |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| 3   | q |   |   |   |   |   |   |   |   | (d) WHERE Q3 = NOT v1, NOT v5, s2 XOR s3, a3 OR a4, a7 AND a1 |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| 4   | V | V | V | V | V | V | V | V | V | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  | V  |
| 5   |   |   |   |   |   |   |   |   |   |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6   |   |   |   |   |   |   |   |   |   |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7   |   |   |   |   |   |   |   |   |   |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8   |   |   |   |   |   |   |   |   |   |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9   |   |   |   |   |   |   |   |   |   |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 10  | L | L | L | L | L | L | L | L | L |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 11  | F | F | F | F | F | F | F | F | F |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 12  | T | F | F | F | F | F | F | F | F |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 13  | F | F | F | F | F | F | F | F | F |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 14  | E | E | E | E | E | E | E | E | E |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 15  | M | M | F | F | F | F | F | F | F |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 16  | M | M | T | T | T | T | T | T | T |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 17  | M | M | T | T | T | T | T | T | T |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 18  | M | M | T | T | T | T | T | T | T |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 19  | M | M | T | T | T | T | T | T | T |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 20  | M | M | T | T | T | T | T | T | T |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 21  | M | M | T | T | T | T | T | T | T |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 22  | A | A | A | A | A | A | A | A | A |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 23  | V | V | V | V | V | V | V | V | V |     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Figure A-32: Result Sheet named "Merge Result" for 'Select Maps', 'By Color', 'Join Maps'

A.2.5.3 Output Section Functionality

- **Output**
  - Output is Context Map format.
  - Output is Graph only.
  - Output is Context Maps plus Graph.

Figure A-33: Output from Query Form
The output format can be selected. However, in this report, all outputs are in Context Map format. The Graph format is recommended for future work.

A.2.5.4 Run Section Functionality

**Schema** ------ Output Map is produced as Zoom-in Map, which is a work frame of Context Maps. This function is as same as 5.2.1 Show Schema.

**Cardinality** ------ Compute the number of non-empty roles of each row and the number of sub-concept value of each Concept set. Detail to see 5.2.2 Compute Cardinality.

**Apply Color** ------ This function is applying different colors for the map in spreadsheet based on the value of each cell. This function is as same as 5.2.3 Apply Color.

**Help** ------ This function is supplying a short version “Help”, which has brief description for each button on the Query Form. This is also as same as 5.2.4 Help.
Appendix B---- Program Source Code

The program was coded by using VB language, the project consists of three parts: user forms, modules and class modules

A user form contains user interface controls, such as command buttons and text boxes

A module is a set of declarations followed by procedures—a list of functions that a program implements.

A class module defines an object, its properties, and its methods. A class module acts as a template from which an instance of an object is created at run time.

A. User Form Source Code

The source code for User Form includes as following created forms:

• frmProjection
• frmSelectConcepts
• frmSelectWorksheets

A.1 frmProjection

Option Explicit

Private m_Canceled As Boolean
Private m_SaveView As Boolean
Private m_NewSheet As Boolean
Private m_Type As Long
Private m_PromptForConcepts As Boolean
Private m_CheckVisible As Boolean
Private m_QueryColor As Boolean
Private m_AutoByColor As Boolean
Private m_SelectMaps As Boolean
Private m_JoinMaps As Boolean
Private m_Cardinality As Boolean
Private m_AppliedColor As Boolean

Public Property Get Canceled() As Boolean
  On Error Resume Next
  Canceled = m_Canceled
End Property

Public Property Let Canceled(BlgCancel As Boolean)
  On Error Resume Next
  m_Canceled = BlgCancel
End Property
Public Property Get SaveView() As Boolean
  On Error Resume Next
  SaveView = m_SaveView
End Property

Public Property Get CheckVisible() As Boolean
  On Error Resume Next
  CheckVisible = m_CheckVisible
End Property

Public Property Get SelectMap() As Boolean
  On Error Resume Next
  SelectMap = m_SelectMaps
End Property

Public Property Get QueryColor() As Boolean
  On Error Resume Next
  QueryColor = m_QueryColor
End Property

Public Property Get AutoByColor() As Boolean
  On Error Resume Next
  AutoByColor = m_AutoByColor
End Property

Public Property Get JoinMap() As Boolean
  On Error Resume Next
  JoinMap = m_JoinMaps
End Property

Public Property Get Cardinality() As Boolean
  On Error Resume Next
  Cardinality = m_Cardinality
End Property

Public Property Get ApplyColor() As Boolean
  On Error Resume Next
  ApplyColor = m_ApplyColor
End Property

Public Property Get NewSheet() As Boolean
  On Error Resume Next
  NewSheet = m_NewSheet
End Property

Public Property Get PromptForConcept() As Boolean
  On Error Resume Next
  PromptForConcept = m_PromptForConcepts
End Property

Public Property Get ProjectionType() As Long
  On Error Resume Next
  ProjectionType = m_Type
End Property

Private Sub chkAutoByColor_Click()
  If Not chkSelectMaps.Value Then
    If chkAutoByColor.Value Then
      If chkJoinMaps.Value Then
        MsgBox "You selected By Color. Join Maps can't be selected at this time";
        chkJoinMaps.Value = False
      End If
    End If
  End If
End Sub
End If
End If
End If
End Sub

Private Sub chkJoinMaps_Click()
If Not chkSelectMaps.Value Then
If chkJoinMaps.Value Then
    If chkAutoByColor.Value Then
        MsgBox "You selected Join Maps. By Color can't be selected at this time"
        chkAutoByColor.Value = False
    End If
End If
End If
End Sub

Private Sub chkSelectMaps_Click()
If chkSelectMaps.Value = False Then
    If chkAutoByColor.Value And chkJoinMaps.Value Then
        MsgBox "You can't select Join Maps and By Color at the same time"
        chkAutoByColor.Value = False
    End If
End If
End Sub

Private Sub cmdCancel_Click()
On Error Resume Next
m_Canceled = True
Call Me.Hide
End Sub

Private Sub cmdOK_Click()
On Error Resume Next

m_PromptForConcepts = chkPromptForConcepts.Value
m_CheckedVisible = chkVisible.Value
m_AutoByColor = chkAutoByColor.Value
m_SelectMaps = chkSelectMaps.Value
m_JoinMaps = chkJoinMaps.Value
m_Cardinality = optCardinality.Value
m_ApplyColor = optApplyColor.Value

If optTypeAnd.Value Then
    m_Type = ptAND
ElseIf optTypeOr.Value Then
    m_Type = ptOR
ElseIf optTypeXor.Value Then
    m_Type = ptXOR
Else
    m_Type = ptNORMAL
End If

'Save the frmProjection Setting
SaveAlphaSetting REG_KEY_INTERACTIVE_PROJECTION_TYPE, CStr(m_Type)
SaveAlphaSetting REG_KEY_INTERACTIVE_PROJECTION_PROMPT_FOR_CONCEPTS, CLng(m_PromptForConcepts)
SaveAlphaSetting REG_KEY_Check_Visible_OUTPUT, CLng(m_CheckVisible)
SaveAlphaSetting REG_KEY_Query_Color_OUTPUT, CLng(m_QueryColor)
SaveAlphaSetting REG_KEY_Auto_ByColor_OUTPUT, CLng(m_AutoByColor)
SaveAlphaSetting REG_KEY_SelectMap_OUTPUT, CLng(m_SelectMaps)
SaveAlphaSetting REG_KEY_JoinMap_OUTPUT, CLng(m_JoinMaps)
SaveAlphaSetting REG_KEY_Cardinality_OUTPUT, CLng(m_Cardinality)
SaveAlphaSetting REG_KEY_ApplyColor_OUTPUT, CLng(m_ApplyColor)
SaveAlphaSetting REG_KEY_INTERACTIVE_PROJECTION_OUTPUT, ipoNONE

m_Canceled = False
Call Me.Hide

End Sub

Private Sub OptHelp_Click()

'O Option Button Help click, showing the help sheet
Sheet10.Activate
m_Canceled = True
Call Me.Hide

End Sub

Private Sub UserFormInitialize()

On Error Resume Next

Select Case GetAlphaSetting(REG_KEY_INTERACTIVE_PROJECTION_TYPE, REG_DEF_INTERACTIVE_PROJECTION_TYPE)
Case pjAnd
 optTypeAnd.Value = True
Case pjOr
 optTypeOr.Value = True
Case pjXor
 optTypeXor.Value = True
Case Else
 optTypeNormal.Value = True
End Select

chkPromptForConcepts.Value = CBool(GetAlphaSetting(REG_KEY_INTERACTIVE_PROJECTION_PROMPT_FOR_CONCEPTS, REG_DEF_INTERACTIVE_PROJECTION_PROMPT_FOR_CONCEPTS))

chkVisible.Value = CBool(GetAlphaSetting(REG_KEY_Check_Visible_OUTPUT, REG_DEF_Check_Visible_OUTPUT))

chkAutoByColor.Value = CBool(GetAlphaSetting(REG_KEY_Auto_ByColor_OUTPUT, REG_DEF_Auto_ByColor_OUTPUT))

chkSelectMaps.Value = CBool(GetAlphaSetting(REG_KEY_SelectMap_OUTPUT, REG_DEF_SelectMap_OUTPUT))

chkJoinMaps.Value = CBool(GetAlphaSetting(REG_KEY_JoinMap_OUTPUT, REG_DEF_JoinMap_OUTPUT))

optCardinality.Value = CBool(GetAlphaSetting(REG_KEY_Cardinality_OUTPUT, REG_DEF_Cardinality_OUTPUT))

optApplyColor.Value = CBool(GetAlphaSetting(REG_KEY_ApplyColor_OUTPUT, REG_DEF_ApplyColor_OUTPUT))

76
End Sub

A.2 frmSelectConcepts

* Elide the detail source code for this FORM.

A.3 frmSelectWorksheets

* Elide the detail source code for this FORM.

B. Modules Source Code
The source code for Modules includes as following created modules:

- mApplyColor
- mAutoByColor
- mMerge
- mProjection

B.1 mApplyColor
'Purpose of this model is applying different background color according to the cells' value

Option Private Module

Public Sub ApplyColor()

Dim mySheet As Excel.Range

'Check if it's concept set value
CheckBold

'Define the color index
AutoRAqua = RGB(60, 195, 238)
AutoN = RGB(230, 160, 162)
AutoELime = RGB(153, 192, 46)
AutotGreen = RGB(0, 251, 0)
AutoFRed = RGB(255, 0, 0)
AutoLightOrg = RGB(222, 150, 51)
AutoPink = RGB(255, 0, 255)
AutoYBarkblue = RGB(107, 7, 184)
AutoPalexblue = RGB(153, 204, 255)
AutoLightPink = RGB(255, 166, 205)
AutosYellow = RGB(238, 207, 0)
AutoVDarkYellow = RGB(226, 178, 0)
AutoSBrightYellow = RGB(255, 255, 0)
AutoAGray = RGB(128, 128, 128)
AutoXLightPurple = RGB(204, 137, 255)
AutoLPurple = RGB(255, 0, 255)
AutoGDarkGreen = RGB(0, 95, 0)
AutobDarkPurple = RGB(128, 0, 128)
AutoBrightBlue = RGB(0, 204, 255)
AutoUPale = RGB(192, 192, 192)
AutoDRed = RGB(194, 61, 87)
AutoCGreen = RGB(62, 193, 104)
AutoOgray = RGB(123, 123, 123)

'wmj
AutoAqua = RGB(60, 186, 196)
AutoELime = RGB(153, 178, 51)
AutoGreen = RGB(0, 251, 0)
AutoRed = RGB(255, 0, 0)
AutoLightOrg = RGB(222, 144, 51)
AutoPink = RGB(255, 0, 255)
AutoPapaya = RGB(153, 204, 255)
AutoLightPink = RGB(255, 166, 205)
AutoYellow = RGB(238, 192, 65)
AutoBrighYellow = RGB(255, 255, 0)
AutoGray = RGB(128, 128, 128)
AutoLightPurple = RGB(204, 137, 255)
AutoPurple = RGB(255, 0, 255)
AutoDarkGreen = RGB(0, 95, 0)
AutoBrighBlue = RGB(0, 204, 255)

Set mySheet = ActiveSheet.UsedRange

With mySheet
    Set c = .Find("A", lookat:=xlWhole, MatchCase:=True)
    If Not c Is Nothing Then
        firstAddress = c.Address
        Do
            c.Interior.Color = AutoAGray
            Set c = .FindNext(c)
            Loop While Not c Is Nothing And c.Address <> firstAddress
        End If
    End If

    Set c = .Find("", Lookln:=xlValue, lookat:=xlPart)
    If Not c Is Nothing Then
        firstAddress = c.Address
        Do
            c.Interior.colorIndex = 15
            Set c = .FindNext(c)
            Loop While Not c Is Nothing And c.Address <> firstAddress
        End If
    End If

** Elide the detail code for applying different background color

End With
For Each c In mySheet.Range(Cells(1, ConceptCol - 2), Cells(mySheet.rows.count, ConceptCol - 1))
    If IsNumeric(c.Value) Then
        c.Font.colorIndex = 3
    End If
Next
End Sub

Private Sub CheckBold()

Dim wks As Worksheet
Dim rng As Range
Dim RowCount As Long
Dim ColCount As Long
Dim rowIndex As Long
Dim colIndex As Long

78
Set wks = ActiveSheet

With wks.Cells.SpecialCells(xlLastCell)
  RowCount = .Row
  ColCount = .Column
End With

Search for the Concept Column within whole sheet

For rowIndex = 1 To RowCount
  For colIndex = ColCount To 1 Step -1
    If IsConceptLabel(wks.Cells(rowIndex, colIndex).Value) Then
      wks.rows(rowIndex).Font.Bold = True
      Exit For
    End If
  Next colIndex
  Next rowIndex
End Sub

B.2 mAutoByColor

*Purpose of this model is using for implementing ByColor function *based on predefined query.

Const BLOCK_ALLOCATION = 100

Dim flgArray() As Boolean
Dim addSheet() As Integer
Dim addSheetCount As Integer
Dim currColumn As Long
Dim currRow As Long
Dim flgFirst As Long
Dim flgPredefined As Boolean

Private Sub AutoQueryByColor(wks As Worksheet, ByVal pnRows_IN As Long, ByVal pnColMax_IN As Long, pnColumns_OUT() As Long)

Dim I, J, RowCount, ColCount As Integer
Dim totalRow As Integer
Dim lngColumns() As Long
Dim lngRows() As Long
Dim lngColumns() As Long
Dim nRowLast, nColLast, notCol As Long
Dim flgFunction As Boolean
Dim flgAnd As Boolean
Dim flgXor As Boolean
Dim flgOr As Boolean

Set wks = ActiveSheet
totalCount = 0

ReDim lngRows(1 To pnRows_IN)

*Check the Red color which means there needs AND operation
For l = 1 To pnRows_IN
  If wks.Cells(l, currColumn).Interior.colorIndex = 3 Then
    totalCount = totalCount + 1
lngRows(totalCount) = I
flgFunction = True
End If

Next I

' Save the result of And operation into array
If flgFunction = True Then

    ReDim Preserve lngRows(1 To totalCount)
    SPAndHideCols wks, lngRows, pnColMax_IN, pnColumns_OUT()
    flgAnd = True

End If

totalCount = 0

flgFunction = False
ReDim lngRows(1 To pnRows_IN)

' Check the Yellow color which means there needs XOR operation
For I = 1 To pnRows_IN

    If wks.Cells(I, currColumn).Interior.colorIndex = 6 Then
        totalCount = totalCount + 1
        lngRows(totalCount) = I
        'pnColMax_IN = j
        flgFunction = True
    End If

Next I

' Save the result of XOR operation into array
If flgFunction = True Then

    ReDim Preserve lngRows(1 To totalCount)
    SPXorHideCols wks, lngRows, pnColMax_IN, pnColumns_OUT()
    flgXor = True

End If

flgFunction = False
totalCount = 0

ReDim lngRows(1 To pnRows_IN)

' Check the Green color which means there needs OR operation
For I = 1 To pnRows_IN

    If wks.Cells(I, currColumn).Interior.colorIndex = 4 Then
        totalCount = totalCount + 1
        lngRows(totalCount) = I
        flgFunction = True
    End If

Next I

' Save the result of OR operation into array
If flgFunction = True Then

    ReDim Preserve lngRows(1 To totalCount)
    SPOrHideCols wks, lngRows, pnColMax_IN, pnColumns_OUT()
    flgOr = True

End If
totalCount = 0
flagFunction = False

ReDim lngRows(1 To pnRows_IN)

'Check the Blue color which means there needs NOT operation
For I = 1 To pnRows_IN
    If wks.Cells(I, currColumn).Interior.colorIndex = 41 Then
        totalCount = totalCount + 1
        lngRows(totalCount) = I
        notCol = I
        flagFunction = True
    End If
Next I

'Save the result of NOT operation into array
If flagFunction = True Then
    ReDim Preserve lngRows(1 To totalCount)
    SNotHideCols wks, lngRows, pnColMax_IN, panColumns_OUT()
End If

End Sub

Private Sub SAndHideCols(wks As Worksheet, ByRef pnRows_IN() As Long, ByVal pnColMax_IN As Long, panColumns_OUT() As Long)

Dim nColCurr As Long
Dim nColumnsIndex As Long
Dim nColumnsUpperBound As Long
Dim nRowsIndex As Long
Dim nRowsUpperBound As Long
Dim nRowsLowerBound As Long
Dim bFoundBlank As Boolean

nColumnsUpperBound = BLOCK_ALLOCATION
ReDim panColumns_OUT(1 To nColumnsUpperBound)
nColumnsIndex = 0
nRowsUpperBound = UBound(pnRows_IN)
nRowsLowerBound = LBound(pnRows_IN)

With wks
    'find next stating Column which is not related to the concept
    For nColCurr = addSheetCount + 1 To pnColMax_IN
        bFoundBlank = False
        For nRowsIndex = nRowsLowerBound To nRowsUpperBound
            If Len(wks.Cells(pnRows_IN(nRowsIndex), nColCurr)) = 0 Then
                bFoundBlank = True
                Exit For
            End If
        Next nRowsIndex
        If bFoundBlank Then
            'Hide the column
            flgArray(flgFirst, nColCurr) = False
        Else
            'All cells full
            nColumnsIndex = nColumnsIndex + 1
            If nColumnsIndex > nColumnsUpperBound Then
                'store the Column number into an array
                nColumnsIndex = nColumnsIndex + 1
            End If
        End If
    Next nColCurr
End With

End Sub
nColumnsUpperBound = nColumnsUpperBound + BLOCK_ALLOCATION
ReDim Preserve panColumns_OUT(1 To nColumnsUpperBound)
End If
    panColumns_OUT(nColumnsIndex) = nColCurr
    flgArray(flgFirst, nColCurr) = True
End If
Next nColCurr
'Resize the array Column
If nColumnsIndex > 0 Then
    ReDim Preserve panColumns_OUT(1 To nColumnsIndex)
Else
    Erase panColumns_OUT
End If
End With

End Sub

Private Sub SPXorHideCols(wks As Worksheet, ByRef panRows_IN(,) As Long, ByVal pnColMax_IN As Long, panColumns_OUT(,) As Long)

Private Sub SPNotHideCols(wks As Worksheet, ByRef panRows_IN(,) As Long, ByVal pnColMax_IN As Long, panColumns_OUT(,) As Long)

Private Sub SPOrHideCols(wks As Worksheet, ByRef panRows_IN(,) As Long, ByVal pnColMax_IN As Long, panColumns_OUT(,) As Long)

*Elide the detail source code for these three function

Public Sub startMultiQuery(wks As Worksheet, objSelection As Object, ByVal flgSelect As Boolean, ByVal sheetNo As Long, ByVal flgConcept As Boolean, ByVal colConcept As Long, ByVal rows As Long, ByVal cols As Long, ByVal operationType As Long, cols_OUT(,) As Long)

    Dim I, J, K, colNoStart, colNoEnd As Long
    Dim objArea As Range
    Dim currSelection As Range
    Dim selectedCol As Long
    Dim flgBlank As Boolean
    Dim Temp As Integer
    Dim queryColCount As Long
    Dim colCollection(,) As Long
    Dim BlankProjection As Boolean

    Set wks = ActiveSheet
    addSheetCount = 0
    flgFirst = 0

    Set objArea = objSelection
    selectedCol = objArea.Areas.count
    ReDim flgArray(1 To selectedCol, 1 To cols)
    ReDim colCollection(1 To selectedCol)

    'check the selection is valid for the criteria which means the selected column has to have Red or Yellow or Blue or Green 'Color

    If Not valBycolorParameter(wks, objArea, colConcept) Then
        Exit Sub
    End If

    queryColCount = 1

    For Each currSelection In objArea.Areas

    Next

For Each currSelection In objArea.Areas

    Next

End Sub
If (objArea.Cells.Columns.count <> 1) Or (objArea.Cells.Rows.count <> 1) Then
    MsgBox "Each selection must be a single cell.", vbInformation + vbOKOnly, PROC_DISPLAY_NAME
    frmProjection.Canceled = True
    Exit Sub
End If
currColumn = currSelection.Column
currRow = currSelection.Row
If flgPredefined = True Then
    For I = 1 To cols
        If LCase(wks.Cells(currRow, I)) = "q" Then
            currColumn = I
            Exit For
        End If
    Next I
End If
flgFirst = flgFirst + 1
colCollection(flgFirst) = currColumn
Call AutoQueryByColor(wks, rows, cols, cols.OUT())
flgArray(qryColCount, currColumn) = True
Next

'Hide columns for 4 operations
For J = 1 To cols
    flgBlank = False
    Temp = 0
    If selectedCol > 1 Then
        For I = 1 To selectedCol
            If J = colCollection(I) Then
                flgBlank = True
                Exit For
            End If
        Next I
    Else
        Select Case operationType
            Case pjtOR
                If flgArray(I, J) Then
                    flgBlank = True
                    Exit For
                Else
                    flgBlank = False
                    wks.Columns(J).Hidden = True
                Exit For
            End If
            Case pjtAND
                If Not flgArray(I, J) Then
                    flgBlank = False
                    Exit For
                Else
                    flgBlank = True
                End If
            Case pjtXOR
                flgBlank = True
                If flgArray(I, J) Then Temp = Temp + 1
                If Temp = selectedCol Or Temp = 0 Then flgBlank = False
            Case pjtNORMAL
                flgBlank = True
                If flgArray(I, J) Then
                    flgBlank = False
                    Exit For
                Else
                    flgBlank = True
                End If
        End Select
    Else
        Exit For
    End If
Next J
If wks.Cells(1, J).Value = "Q" Then
    flgBlank = False
    Exit For
End If

End Select
End If
Next J
If flgBlank = False Then
    wks.Columns(J).Hidden = True
End If
Else
    If J = currColumn Then
        wks.Columns(J).Hidden = False
    ElseIf Not flgArray(1, J) Then
        wks.Columns(J).Hidden = True
    End If
End If
Next J

' check the gray color for the concept value
If Not flgConcept Then
    flgBlank = False
    For I = 1 To rows
        If IsConceptLabel(wks.Cells(I, colConcept)) Then
            For J = 1 To colConcept
                If wks.Cells(I, J).Interior.colorIndex = 15 Or InStr(wks.Cells(I, colConcept).Value, "Query") > 0
                    Then
                        flgBlank = False
                        Exit For
                    Else
                        flgBlank = True
                    End If
                Next J
            Else
                If flgBlank = True Then wks.rows(I).Hidden = True
            Else
                If flgBlank = True Then wks.rows(I).Hidden = True
            End If
        Next I
    End If
End If

ReDim cols_OUT(0)
    J = 0
    For I = 1 To colConcept - 3
        If Not wks.Columns(I).Hidden Then
            J = J + 1
            ReDim Preserve cols_OUT(0 To J)
            cols_OUT(J) = I
        End If
    Next I
    ReDim Preserve cols_OUT(0 To J)
End Sub

Public Function PromptWorkSheets(ByRef colSelectedSheets As Collection) As Boolean

Dim nSheetIndex As Long
Dim colAllSheets As Collection
On Error Resume Next
Set colAllSheets = New Collection
With ActiveWorkbook
    Set colSelectedSheets = Nothing
End With
PromptWorkSheets = True
With ActiveWorkbook
    Set colSelectedSheets = colAllSheets
End With
End Function
For nSheetIndex = 1 To .sheets.count
    colAllSheets.Add (.sheets.item(nSheetIndex)).Name
Next nSheetIndex
End With

If Not SelectWorkSheet(colAllSheets, colSelectedSheets) Then
    Exit Function
End If
PromptWorkSheets = True
End Function

Private Function valBycolorParameter(ByRef wks As Worksheet, ByRef objSelected As Object, ByVal
    ConceptColumn As Long) As Boolean

    Dim nAreaCount As Long
    Dim rg As Range
    Dim rngCellCurr As Range
    Dim nColInstance As Long
    Dim colorIndex As Long
    Dim strQuery As String
    On Error Resume Next
    figPredefined = False
    Set rg = objSelected
    nAreaCount = rg.Areas.count
    'check that the selection consists of non empty cells in the same column

    For Each rngCellCurr In rg.Areas
            MsgBox "Each selection must be a single cell.", vbInformation + vbOKOnly.
        PROC_DISPLAY_NAME
        Exit Function
        End If
        If nColInstance = 0 Then
            nColInstance = rngCellCurr.Column
            If rngCellCurr.Column = ConceptColumn Then
                nColInstance = 0
            End If
        ElseIf nColInstance = rngCellCurr.Column Then
            MsgBox "The selected cells must all be in the same column.", vbInformation + vbOKOnly.
        PROC_DISPLAY_NAME
        Exit Function
        End If

        'check interior color
            .Interior.ColorIndex
        strQuery = UCase(wks.Cells(rngCellCurr.Row, ConceptColumn))
        If rngCellCurr.Column = ConceptColumn Then
            If InStr(strQuery, "AND") <> 0 Or InStr(strQuery, "XOR") <> 0 Or InStr(strQuery, "OR") <> 0 Or InStr(strQuery, "NOT") = 0 Then
                figPredefined = True
            End If
            ElseIf colorIndex <> 41 And colorIndex <> 3 And colorIndex <> 4 And colorIndex <> 6 Then

            If InStr(strQuery, "AND") = 0 And InStr(strQuery, "XOR") = 0 And InStr(strQuery, "OR") = 0 And InStr(strQuery, "NOT") = 0 Then

                MsgBox "Modified later The selected cells must cannot be empty.", vbInformation + vbOKOnly.
            PROC_DISPLAY_NAME
            Exit Function

85
End If
End If
Next rngCellCurr
Set rngCellCurr = Nothing
Set rng = Nothing
valBycolorParameter = True
End Function

B.3 mMerge

Purpose of this module is implementing merging selected sheets into one sheet called Merge Result. It can either run by itself or work with Select Maps Function

Public Sub Merge(joinSheets As Collection)

Dim WS_Count As Integer
Dim I As Integer
Dim flgNewSheet As Boolean
flgNewSheet = False
For I = 1 To ActiveWorkbook.Worksheets.count
If ActiveWorkbook.Worksheets(I).Name = "Merge Result" Then
    sheets("Merge Result").Select
    Application.CutCopyMode = False
    Application.DisplayAlerts = False
    ActiveWindow.SelectedSheets.Delete
    Exit For
End If
Next I

Dim wks As Worksheet
Dim newWks As New Worksheet
sheets(joinSheets(1)).Select
Set wks = ActiveSheet
wks.UseRange.SpecialCells(xlCellTypeVisible).Copy
Set newWks = ActiveWorkbook.Worksheets.Add(after:=WorkSheets(WorkSheets.count))
newWks.Name = "Merge Result"
With newWks
    .Paste
    Call JoinMaps(joinSheets)
End Sub

Private Sub JoinMaps(mergeSheets As Collection)

Dim resultNO, wksNO, I, J, K As Long
Dim sheetNo As Integer
Dim resultRows As Long
Dim resultCols As Long
Dim resultColConcept As Long
Dim rstRowIndex As Long
Dim rstRowIndexDes As Long
Dim wksRowIndex As Long
Dim wksRowIndexDes As Long
Dim wksRows As Long
Dim wksCols As Long
Dim wksColConcept As Long
Dim Temp As Long
Dim insertRow As Long
Dim wks As Worksheet

86
Dim resultWks As Worksheet
Dim resultSet() As String
Dim wksSet() As String
Dim_flgSet Equal As Boolean
Dim_flgAtomEqual As Boolean

Set resultWks = Worksheets("Merge Result")
For sheetNo = 2 To mergeSheets.count
  'get the rows and columns number, the concept value column no.
  If Not FindDataExtent(resultWks, resultRows, resultCols, resultColConcept) Then
    Call MsgBox(ERM_UNKNOWN_DATA_EXTENT, vbInformation + vbOKOnly, "Map Projection")
  End If
  Set wks = Worksheets(mergeSheets(sheetNo))
  If Not FindDataExtent(wks, wksRows, wksCols, wksColConcept) Then
    Call MsgBox(ERM_UNKNOWN_DATA_EXTENT, vbInformation + vbOKOnly, "Map Projection")
  End If

  'saved the concept set value and its row number into array
  ReDim wksSet(2, 0)
  Temp = findConceptSet(wks, wksSet(), wksRows, wksColConcept)
  'get the visible row number, col number and concept column no from the Select Maps result sheet
  wksRows = 0
  wksCols = 0
  For I = 1 To wks.UsedRange.rows.count
    If Not wks.Rows(I).Hidden Then
      wksRows = wksRows + 1
    End If
  Next I

  For I = 1 To wks.UsedRange.Columns.count
    If Not wks.Columns(I).Hidden Then
      wksCols = wksCols + 1
      If IsConceptLabel(wks.Cells(1, I)) Then
        wksColConcept = 1
        wksCols = wksCols - 3
        Exit For
      End If
    End If
  Next I

  'move the columns in merge result sheets
  resultWks.Range(Cells(1, resultCols + 1), Cells(resultRows, resultCols + 4)).Select
  Selection.Cut Destination:=Range(Cells(1, resultCols + wksCols + 1), Cells(resultRows, resultCols + wksCols + 4))

  'find set value and row number
  ReDim resultSet(2, 0)
  Temp = findConceptSet(resultWks, resultSet(), resultRows, resultColConcept + wksCols)
  insertRow = 0

  'compare concept set value
  For wksNG = 1 To UBound(wksSet, 2)
    _flgSetEqual = False
    For resultNO = 1 To UBound(resultSet, 2)
      If wksSet(1, wksNG) = resultSet(1, resultNO) Then
        _flgSetEqual = True
        Exit For
      End If
    Next resultNO
    wksRowIndex = wksSet(2, wksNG)
If wksNO = UBound(wksSet, 2) Then
    wksRowIndexDes = wks.UsedRange.rows.count + 1
Else
    wksRowIndexDes = wksSet(2, wksNO + 1)
End If

Equal

If flgSetEqual Then
    rstRowIndex = resultSet(2, resultNO) + insertRow
Else
    rstRowIndexDes = resultWks.UsedRange.rows.count + 1
End If

For J = wksRowIndex To wksRowIndexDes - 1
    flgAtomEqual = False

    If Not wks.rows(J).Hidden Then
        K = rstRowIndex
        Do While K < rstRowIndexDes
            Debug.Print wks.Cells(K, resultColConcept + wksCols)
            Debug.Print wks.Cells(J, wksColConcept)
        If (resultWks.Cells(K, resultColConcept + wksCols) = wks.Cells(J, wksColConcept)) And resultWks.Cells(K, resultColConcept + wksCols + 1) = wks.Cells(J, wksColConcept + 1) Then
            flgAtomEqual = True
        End If
        If flgAtomEqual Then
            wks.Activate
            wks.Range(Cells(J, 1), Cells(J, wksColConcept + 3)).SpecialCells(xlCellTypeVisible).Select
            Selection.Copy
            resultWks.Activate
            resultWks.Cells(K, resultCols + 1).Select
            K = K + 1
            ActiveSheet.Paste
            Exit Do
        End If
        K = K + 1
    Loop

    If Not flgAtomEqual Then
        resultWks.Activate
        resultWks.rows(rstRowIndexDes).Select
        Selection.Insert Shift:=xlDown
        wks.Activate
        wks.Range(Cells(J, 1), Cells(J, wksColConcept + 1)).SpecialCells(xlCellTypeVisible).Select
        Selection.Copy
        resultWks.Activate
        resultWks.Cells(K, resultCols + 1).Select
        ActiveSheet.Paste
        insertRow = insertRow + 1
        rstRowIndexDes = rstRowIndexDes + 1
    End If
    Next J
End If

rstRowIndex = resultSet(2, resultNO - 2) + insertRow
resultWks.Activate
For I = rstRowIndex To wksRowIndexDes - 1
    resultWks.rows(rstRowIndex).Select
Sub Selection.Insert shift:=xlDown
   Next I
   wks.Activate
   wks.Range(Cells(wks.RowIndex, 1), Cells(wks.RowIndexDes - 1, wksColConcept + 1)).SpecialCells(xlCellTypeVisible).Select
   Selection.Copy
   resultWks.Activate
   resultWks.Cells(rstRowIndex, resultCols + 1).Select
   ActiveSheet.Paste
   insertRow = insertRow + wksRowIndexDes - wksRowIndex
   rstRowIndexDes = rstRowIndexDes + wksRowIndexDes - wksRowIndex
End If
Next wksNo
Next sheetNo
resultWks.Columns.ColumnWidth = 2
End Sub

Private Function findConceptSet(ByRef wks As Worksheet, ByRef setArray() As String, ByVal rowNo As Long, ByVal cCol As Long) As Long
Dim index As Long
Dim count As Long
   count = 0
For index = 1 To rowNo
   If IsConceptLabel(wks.Cells(index, cCol)) Then
      If Not wks.row(index).Hidden Then
         count = count + 1
         ReDim Preserve setArray(2, UBound(setArray, 2) + 1)
         setArray(1, count) = wks.Cells(index, cCol)
         setArray(2, count) = index
      End If
   End If
Next index
   findConceptSet = count
End Function

B.4 mProjection

'Purpose of this module is retrieve the control value from the frmProjection form, and running And, Xor, Or, Not operation

'Public UserMap As Worksheet
Public Const pjtNORMAL As Long = 0
Public Const pjtAND As Long = 1
Public Const pjtOR As Long = 2
Public Const pjtXOR As Long = 3
Public Const pjtANDNOT As Long = 4
Public Const pjtORNOR As Long = 5
Public Const pjtAutoColor As Long = 10
Public Const relopEQUAL = 0
Public Const relopGREATEREQUAL = 1
Public Const relopLESSEQUAL = 2
Public Const PROC_DISPLAY_NAME As String = "JMap Projection"

Public Sub InteractiveProjection(ByRef wks As Worksheet, ByRef objSelected As Object)
   Dim frmProjectionOptions As frmProjection
   On Error Resume Next
   Set frmProjectionOptions = New frmProjection
   Load frmProjectionOptions
   With frmProjectionOptions
      .Show
      If Not .Canceled Then
         'Code here
   End With
End Sub
Private Function ValidateProjectionParameters(ByRef wks As Worksheet, ByRef objSelected As Object, ByRef ProjectionType_INOUT As Long) As Boolean

Dim nAreaCount As Long
Dim rng As Range
Dim rngCellCurr As Range
Dim nColInstance As Long
On Error Resume Next

Select Case ProjectionType_INOUT
    Case pjXOR, pjAND, pjOR, pjNORMAL
    Case Else
        ProjectionType_INOUT = pjNORMAL
End Select

'check if user select more than one area
Set rng = objSelected
nAreaCount = rng.Areas.count
Select Case nAreaCount
'Check the selection validation
    Case Is < 1
        MsgBox "This macro does not support the specified selection.", vbInformation + vbOKOnly.
        PROC_DISPLAY_NAME
        Exit Function
    Case Is = 1
        If ProjectionType_INOUT <> pjOR Then
            ProjectionType_INOUT = pjNORMAL
        End If
    Case Is > 1
        MsgBox "This macro does not support multi selection. Please try either the 'And' or the 'Or' projection.", vbInformation + vbOKOnly, PROC_DISPLAY_NAME
        Exit Function
End If
End Select

'Check that the selection consists of non empty cells in the same column
For Each rngCellCurr In rng.Areas
    If (rng.Cells.Columns(count <> 1) Or (rng.Cells.rows.count <> 1)) Then
        MsgBox "Each selection must be a single cell.", vbInformation + vbOKOnly, PROC_DISPLAY_NAME
        Exit Function
    End If
End If

'Same Column
If nColInstance = 0 Then
    nColInstance = rngCellCurr.Column
End If
If nColInstance <> rngCellCurr.Column Then
    MsgBox "The selected cells must all be in the same column.", vbInformation + vbOKOnly.
    PROC_DISPLAY_NAME
    Exit Function
End If
Public Function ProjectMap(ByVal wks As Worksheet, ByVal obj As Object, Optional ByVal SaveView As Boolean = False, Optional ByVal PasteToNewSheet As Boolean = False, Optional ByVal PromptForConcepts As Boolean = False, _
    Optional ByVal CheckVisibles As Boolean = False, Optional ByVal QueryColors As Boolean = False, _
    Optional ByVal AutoByColors As Boolean = False, Optional ByVal SelectMaps As Boolean = False, _
    Optional ByVal JoinMaps As Boolean = False, Optional ByVal getCardinality As Boolean = False, _
    Optional ByVal ApplyingColor As Boolean = False, _
    Optional ByVal ProjectionType As Long = pjNORMAL, Optional ByVal SuppressUserInteraction As Boolean = False) As Boolean

    Dim nRowCurr As Long
    Dim nColCurr As Long
    Dim nColConcept As Long
    Dim strViewName As String
    Dim nColMax As Long
    Dim nRowMax As Long
    Dim nColLeftOut As Long
    Dim strLeftOut As String
    Dim nLeftOut As Long
    Dim nColConcepts() As Long
    Dim nSelectRows() As Long
    Dim nSelectCols() As Long
    Dim nRowSelection As Range
    Dim nColSelection As Range
    Dim nRowCurr As Range
    Dim nColCurr As Range
    Dim svi As CSuspendVisualInterface
    Dim selectedWorksheets As Collection
    Dim resultSheets As Collection
    Dim nIndex As Long
    Dim nParentConceptRow As Long
    Dim asParentConcepts() As String
    Dim bBlankProjection As Boolean
    Dim c As Range

    'Instance a class
    Set svi = New CSuspendVisualInterface
    svi.StatusBar = "Projecting..."
    If Not AutoByColors And Not JoinMaps Then
      If Not ValidateProjectionParameters(wks, obj, ProjectionType) Then
        Exit Function
      End If
    End If
    Set mgSelection = obj

    'get the rows and columns count number, and concept column number
    If Not FindDataExtent(ActiveSheet, nRowMax, nColMax, nColConcept) Then
      Call MsgBox("ERM_UNKNOWN_DATA_EXTENT", vbInformation + vbOKOnly, "jMap Projection")
      GoTo PROC_EXIT
    End If

    'check the visible button from the frmProjection
    If CheckVisibles <> True Then
      ShowAll wks
    End If
    With wks
        nRowCurr = mgSelection.Row
    End With

    Dim nr As Long
    Dim nc As Long
    Dim strViewName As String
    Dim nColLeftOut As Long
    Dim strLeftOut As String
    Dim nLeftOut As Long
    Dim nColConcepts() As Long
    Dim nSelectRows() As Long
    Dim nSelectCols() As Long
    Dim nRowSelection As Range
    Dim nColSelection As Range
    Dim nRowCurr As Range
    Dim nColCurr As Range
    Dim svi As CSuspendVisualInterface
    Dim selectedWorksheets As Collection
    Dim resultSheets As Collection
    Dim nIndex As Long
    Dim nParentConceptRow As Long
    Dim asParentConcepts() As String
    Dim bBlankProjection As Boolean
    Dim c As Range

    'Instance a class
    Set svi = New CSuspendVisualInterface
    svi.StatusBar = "Projecting..."
    If Not AutoByColors And Not JoinMaps Then
      If Not ValidateProjectionParameters(wks, obj, ProjectionType) Then
        Exit Function
      End If
    End If
    Set mgSelection = obj

    'get the rows and columns count number, and concept column number
    If Not FindDataExtent(ActiveSheet, nRowMax, nColMax, nColConcept) Then
      Call MsgBox("ERM_UNKNOWN_DATA_EXTENT", vbInformation + vbOKOnly, "jMap Projection")
      GoTo PROC_EXIT
    End If

    'check the visible button from the frmProjection
    If CheckVisibles <> True Then
      ShowAll wks
    End If
    With wks
        nRowCurr = mgSelection.Row
    End With
'Check whether the ByColor button from the frmProjection is true
    If AutoByColors Then
        If SelectMaps Then
            'doing Select Maps
            Set selectedWorkSheets = New Collection
            Set resultSheets = New Collection
            Call PromptWorkSheets(selectedWorkSheets)
            If selectedWorkSheets.count > 0 Then
                For nIndex = 1 To selectedWorkSheets.count
                    Set wks = selectedWorkSheets(nIndex)
                    sheets(selectedWorkSheets(nIndex)).Select
                    Set wks = ActiveSheet
                    ShowAll wks
                    If Not FindDataExtent(wks, nRowMax, nColMax, nColConcept) Then
                        Call MsgBox(ERM_UNKNOWN_DATA_EXTENT, vbInformation + vbOKOnly, "jMap Projection")
                        GoTo PROC_EXIT
                    End If
                    svi.StatusBar = "MultiSheets ByColor ...."
                    Call startMultiQuery(wks, Selection, SelectMaps, nIndex, False, nColConcept, nRowMax, nColMax,
                        ProjectionType, alngColumns())
                    Call resultSheets.Add(wks.Name)
                    Next nIndex
                End If
            'Select Maps and Join Maps work togetehr
            If JoinMaps Then
                svi.StatusBar = "Merging Selected Sheets ...."
                Call Merge(resultSheets)
            End If
            Else
                'Select Maps only with ByColor
                svi.StatusBar = "MultiSheets ByColor ...."
                Call startMultiQuery(wks, obj, SelectMaps, nIndex, PromptForConcepts, nColConcept, nRowMax, nColMax,
                    ProjectionType, alngColumns())
            End If
            ElseIf SelectMaps = False And JoinMaps = True Then
                'Join Maps only without select Maps and ByColor
                Set selectedWorkSheets = New Collection
                Call PromptWorkSheets(selectedWorkSheets)
                svi.StatusBar = "Merging Selected ByColor ...."
                Call Merge(selectedWorkSheets)
            Else
                'Implementing AND, XOR, OR, NOT only with ByColor
                Select Case ProjectionType
                    Case pjAND, pjOR
                        'Add all the rows to the array
                        ReDim alngRows(1 To rngSelection.Areas.count)
                        nRangeCurr = 1
                        For Each rngCurr In rngSelection.Areas
                            alngRows(nRangeCurr) = rngCurr.Row
                            nRangeCurr = nRangeCurr + 1
                            Next rngCurr
                        'Project
                        If QueryColors <> True Then
                            If ProjectionType = pjAND Then
                                SPAndHideCols wks, alngRows, nColMax, alngColumns()
                            ElseIf ProjectionType = pjXOR Then
                                SXorHideCols wks, alngRows, nColMax, alngColumns()
                            Else
                                SPOrHideCols wks, alngRows, nColMax, alngColumns()
                            End If
                        Else
                            ElseIf ProjectionType = pjAND Then
                                SPAndHideCols wks, alngRows, nColMax, alngColumns()
                            ElseIf ProjectionType = pjXOR Then
                                SXorHideCols wks, alngRows, nColMax, alngColumns()
                            Else
                                SPOrHideCols wks, alngRows, nColMax, alngColumns()
                            End If
                        Else
                            ElseIf ProjectionType = pjAND Then
                                SPAndHideCols wks, alngRows, nColMax, alngColumns()
                            ElseIf ProjectionType = pjXOR Then
                                SXorHideCols wks, alngRows, nColMax, alngColumns()
If ProjectionType = pjtAND Then
    QuerySPAndHideCols wks, alngRows, nColMax, alngColumns()
Else
    QuerySPORHideCols wks, alngRows, nColMax, alngColumns()
End If

End If

Case pjtXOR
    ReDim alngRows(1 To rngSelection.Areas.count)
    nRangeCurr = 1
    For Each rngCurr In rngSelection.Areas
        alngRows(nRangeCurr) = rngCurr.Row
        nRangeCurr = nRangeCurr + 1
    Next rngCurr
    If QueryColors <> True Then
        Call `TestSPXorHideCols(wks, alngRows, nColMax, alngColumns())
    Else
        Call QuerySPXorHideCols(wks, alngRows, nColMax, alngColumns())
    End If
Case Else
    If QueryColors <> True Then
        Call `TestSPNotHideCols(wks, nRowCurr, nColMax, alngColumnists())
    Else
        Call QuerySPNotHideCols(wks, nRowCurr, nColMax, alngColumnists())
    End If
End Select
End If

svi.StatusBar = "Projecting ...."

'check the Select Sets value from frmProjection form
If PromptForConcepts Then
    ReDim asParentConcepts(1 To rngSelection.Areas.count)
    nRangeCurr = 1
    For Each rngCurri In rngSelection.Areas
        If Not ConceptFromInstance(rngCurri.Row, wks, nColConcept, nParentConceptRow) Then
            MsgBox "Unable to locate the parent concept for the selected instance(s). Please verify jMap integrity before using this macro.", vbInformation + vbOKOnly, "jMap Projection"
        Exit Function
    End If
    asParentConcepts(nRangeCurr) = wks.Cells.item(nParentConceptRow, nColConcept).Value
    nRangeCurr = nRangeCurr + 1
    Next rngCurri
End If

'Get the rows
If Not SelectMaps Then
    If Not SPBubbleRows(wks, nColConcept, nRowMax, alngColumnists(), asParentConcepts()) Then
        Exit Function
    End If
Else
    'Hiding empty rows
    If ProjectionType <> pjtNORMAL Or AutoByColors Then
        If Not JoinMaps Then
            Call SPHideRows(wks, nColConcept, nRowMax, alngColumnists(), bBlankProjection)
        End If
    End If
End If
End With

'Calculate Cardinality
If getCardinality Then
    Call ComputeCardinality(ActiveSheet)
End If

'Applying Color
If ApplyingColor Then
    svi.StatusBar = "Applying Color...."
    ApplyColor
End If

If bBlankProjection Then
    If Not SuppressUserInteraction Then
        If vbYes = MsgBox("The selected instance(s) have produced a blank Projection. Do you want to undo the projection?", vbYesNo) Then
            ShowAll wks
        End If
    End If
Else
    With wks
        Call Activate
        ActiveWindow.ScrollColumn = 1
    End With
End If

If SaveView Then
    strViewName = Trim$(InputBox("Please enter a name for the Map Projection: ": "Add Map Projection to View Manager": _
    .Cells(nRowCurr, nColMax + 4).Text = " View")
    If Len(strViewName) > 0 Then
        Call ActiveWorkbook.CustomViews.Add(strViewName, True, True)
    End If
End If

If PasteToNewSheet Then
    Copy the current jMap projection
    Selection.CurrentRegion.Select
    Selection.SpecialCells(xlCellTypeVisible).Select
    Selection.Copy
    'Deselect the selected cells
    .Cells(1, 1).Select
    'Create a new Workbook
    Dim wkbNew As Workbook
    Set wkbNew = Workbooks.Add
    'Using the new workbook
    With wkbNew.Worksheets(1)
        Paste the projection into it
        .Paste
        Make it look nice by adjusting column sizes
        .Columns.AutoFit
        'Fix the concept column so instances look good
        Call FindConceptColumn(wkbNew, Worksheets(1), nColConcept)
        .Columns(nColConcept).ColumnWidth = 1
        Fix text and cell formatting
        Selection.WrapText = False
        Selection.Orientation = 0
        Selection.ShrinkToFit = False
        Selection.MergeCells = False
        'Deselect the sheet
        .Cells(1, 1).Select
    End With
    'Deselect the sheet and turn off copy mode
    Application.CutCopyMode = False
    'Fill in the groupings
    Call GroupjMap(ActiveSheet)
End If
End With

If Not SuppressUserInteraction Then
    Call NotifyMacroFinished("Finished the Projection.", PROC_DISPLAY_NAME, svi.StartTime)
End If
End If

ProjectMap = True

PROC_EXIT:
    On Error Resume Next
    Erase alngColumns
    Erase alngRows
End Function

Public Sub SPHideRows(wks As Worksheet, ByVal colConcept As Long, ByVal nRowMax As Long, alColumns() As Long, ByRef BlankProjection_OUT As Boolean)
    'Purpose: hideBlankRow used by both projection and projectionSave

    Dim nRowCurr As Long
    Dim nColCurr As Long
    Dim blnFoundNonBlank As Boolean
    Dim nColumnsStillVisible As Long
    Dim svi As CSuspendVisualInterface
    On Error Resume Next
    Set svi = New CSuspendVisualInterface
    svi.StatusBar = "Projection: Hiding Rows."
    nColumnsStillVisible = UBound(alColumns)
    If Err.Nr <> 0 Then
        ActiveSheet.rows("1." & CStr(nRowMax)).Hidden = True
        BlankProjection_OUT = True
        Exit Sub
    Else
        BlankProjection_OUT = False
    End If
    On Error GoTo 0
    With wks.Cells
        'find blank row then hide the entire row
        For nRowCurr = 1 To nRowMax
            blnFoundNonBlank = False
            If Left(wks.Cells(nRowCurr, colConcept).Value, 1) = "Q" Then
                blnFoundNonBlank = True
            Else
                For nColCurr = 1 To nColumnsStillVisible
                    If Len(wks.Cells(nRowCurr, alColumns(nColCurr)).Value) > 0 Then
                        blnFoundNonBlank = True
                    Exit For
                End If
                Next nColCurr
            If Not blnFoundNonBlank Then
                wks.rows(nRowCurr).Hidden = True
            End If
        End If
        Next nRowCurr
    End With
End Sub

Private Function SPPromptForRows(wks As Worksheet, ByVal colConcept As Long, ByVal nRowMax As Long, alColumns() As Long, ParentConcepts() As String) As Boolean
'Purpose: hideBlankRow used by both projection and projectionSave
Dim nIndex As Long
Dim nRowCurr As Long
Dim nColCurr As Long
Dim blnFoundNonBlank As Boolean
Dim nColumnsStillVisible As Long
Dim colConceptsRelated As Collection
Dim colConceptsSelected As Collection
Dim sConceptCurr As String
Dim nConceptStart As Long
Dim svi As C SUS pendVisualInterface
On Error Resume Next
Set svi = New C SUS pendVisualInterface
svi.StatusBar = "Projection: Hiding Rows"
    nColumnsStillVisible = UBound(alColumns)
If Err.Number <> 0 Then
    ActiveSheet.rows("1:" & CStr(nRowMax)).Hidden = True
    Exit Function
End If
Set colConceptsRelated = New Collection
With wks.Cells
For nRowCurr = 1 To nRowMax
    If IsConceptLabel(item(nRowCurr, colConcept)) Then
        blnFoundNonBlank = False
        For nColCurr = 1 To nColumnsStillVisible
            If Len(item(nRowCurr, alColumns(nColCurr)).Value) > 0 Then
                blnFoundNonBlank = True
            Exit For
        Next nColCurr
        If blnFoundNonBlank Then
            Call colConceptsRelated.AddItem(nRowCurr, colConcept), UCase$(item(nRowCurr, colConcept))
        End If
    End If
Next nRowCurr
End With wks.Cells
svi.ReEnableInterface
If Not SelectConcepts(colConceptsRelated, colConceptsSelected) Then
    Exit Function
End If
svi.ReDisableInterface
With wks.Cells
    'find the first concept
    nRowCurr = 1
    Do Until IsConceptLabel(item(nRowCurr, colConcept))
        nRowCurr = nRowCurr + 1
    Loop
Do Until nRowCurr > nRowMax
    'nRowCurr points to a concept row
    Err.Clear
    sConceptCurr = colConceptsSelected.item(UCase$(item(nRowCurr, colConcept)))
    If Err.Number = 0 Then
        'Skip the current concept
        nRowCurr = nRowCurr + 1
        'find blank row then hide the entire row
        Do Until nRowCurr > nRowMax _ Or IsConceptLabel(item(nRowCurr, colConcept))
            blnFoundNonBlank = False
            For nColCurr = 1 To nColumnsStillVisible
                If Len(item(nRowCurr, alColumns(nColCurr)).Value) > 0 Then

96
blnFoundNonBlank = True
    Exit For
End If
Next nColCurr
If Not blnFoundNonBlank Then
    wks.rows(nRowCurr).Hidden = True
End If
nRowCurr = nRowCurr + 1
Loop
'
Else
' Find the start of the next concept
nConceptStart = nRowCurr
nRowCurr = nRowCurr + 1
Do Until nRowCurr > nRowMax
    Or IsConceptLabel(item(nRowCurr, colConcept))
    nRowCurr = nRowCurr + 1
Loop
'
If nConceptStart = nRowCurr - 1 Then
    wks.rows(nConceptStart).Hidden = True
Else
    wks.rows(CStr(nConceptStart) & ".* & CStr(nRowCurr - 1)) Hidden = True
End If
End If
Loop
End With
SPPromptForRows = True
End Function

Public Sub ShowAll(ByRef wks As Worksheet)
    On Error Resume Next
    UnHideAll wks
End Sub

Private Sub SPPAndHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub SPOrHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub SPXorHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub QuerySPXorHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub TestSPNotHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub QuerySPPAndHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub QuerySPOrHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

Private Sub QuerySPNotHideCols(wks As Worksheet, ByVal panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)
Private Sub TestSPXorHideCols(wks As Worksheet, ByRef panRows_IN As Long, ByVal pnColMax_IN As Long, panColumns_OUT As Long)

- Elide the detail source code for these functions

C. Classes Source Code
The source code for Class includes as following:

- CSuspendVisualInterface

Option Explicit

Private mblnScreenUpdating As Boolean
Private mlngMousePointer As Long
Private mvarInitialStatusBar As Variant
Private msgStartTime As Single

Private Sub Class_Initialize()
On Error Resume Next
    With Application
    ' Save the current state
    mlngMousePointer = .Cursor
    mblnScreenUpdating = .ScreenUpdating
    mvarInitialStatusBar = .StatusBar
    ' Set the wait state
    .Cursor = xlWait
    If CBool(GetAlphaSetting(REG_KEY_SCREEN_UPDATING_OFF, REG_DEF_SCREEN_UPDATING_OFF)) Then
        .ScreenUpdating = False
        End If
        .StatusBar = "Processing..."
        msgStartTime = timer
    End With
End Sub

Private Sub Class_Terminate()
On Error Resume Next
    With Application
    .Cursor = mlngMousePointer
    .ScreenUpdating = mblnScreenUpdating
    If 0 = StrComp(CStr(mvarInitialStatusBar & ",", "FALSE", vbTextCompare)) Then
        .StatusBar = False
    Else
        .StatusBar = mvarInitialStatusBar
    End If
    End With
End Sub

Public Sub ReEnableInterface()
On Error Resume Next
    With Application
    .Cursor = xlDefault
    .ScreenUpdating = True
    End With
End Sub

Public Sub ReDisableInterface()
On Error Resume Next
    DoEvents
With Application
    .Cursor = xlWait
    .ScreenUpdating = False
End With
End Sub

Public Property Get StatusBar() As String
    On Error Resume Next
    StatusBar = Application.StatusBar
End Property

Public Property Let StatusBar(ByVal NewValue As String)
    On Error Resume Next
    Application.StatusBar = NewValue
End Property

Public Property Get StartTime() As Single
    On Error Resume Next
    StartTime = msngStartTime
End Property