Abstract

Specifying the requirements to be met by a new system is a fundamental activity in the early stage of the system development life cycle. Any errors occur at this point will affect the implementation phase of the system. The specification of requirements need not be a very large document or too technical but it must contain sufficient information to enable stakeholders understand the requirements of the system to be developed and respond to it. As an alternative for better understanding of the requirements, a requirements visualization tool, ReVis is proposed and developed. The tool is mainly developed to organize, visualize and verify the requirements gathered in requirements elicitation stage based on stakeholders. Node Link Diagram, TreeMaps, Dynamic Queries, Focus + Context, Overview + Detail, Colour, Zooming, Highlighting are the visualization techniques applied in this tool.
ACKNOWLEDGEMENTS

In the name Of Allah, the Most Gracious and the Most Merciful, Peace and blessings of Allah upon Prophet Muhammad S.A.W.

First and foremost praise be to Allah, for enabling me to complete my dissertation.

Special thanks and appreciation is given to my supervisor, Dr. Zarinah Mohd Kasirun on her time, assistance and guidance throughout the preparation of this dissertation.

Thanks to all my colleagues at Software Engineering Section, Universiti Kuala Lumpur and my friends for their valuable time, helps and knowledge.

My sincere thanks go to my best friend, Suzana Kassim for her continuous ideas, support and encouragement in all ways.

Grateful thanks to my parents and all family members for their constant source of strength.

Finally and most of all, my deep appreciation and thanks to my husband, Kamal and my two childrens, Muhammad Naz Han and Ulin Nuha for putting up with me through the long days and nights. You are my joy, strength and hope.
TABLE OF CONTENT

Abstract i
Acknowledgements ii
Table of Content iii
List of Figures v
List of Tables viii
List of Graphs ix

CHAPTER 1 – INTRODUCTION ................................................................. 1
1.1 OVERVIEW .................................................................................. 1
1.2 PROBLEM STATEMENT ............................................................... 2
1.3 RESEARCH OBJECTIVES ............................................................. 4
1.4 SCOPE OF RESEARCH ................................................................. 4
1.5 RESEARCH METHODOLOGY ....................................................... 4
1.6 DISSERTATION OUTLINE ............................................................ 7

CHAPTER 2 – LITERATURE REVIEW .................................................... 8
2.1 REQUIREMENT VISUALIZATION .................................................... 8
2.2 VISUALIZATION OVERVIEW ...................................................... 9
2.3 VISUALIZATION TECHNIQUES .................................................. 12
   2.3.1 Node-link diagram ............................................................... 12
   2.3.2 TreeMap .............................................................................. 13
   2.3.3 Dynamic Queries ............................................................... 14
   2.3.4 Overview + Detail ............................................................... 15
   2.3.5 Focus + Context ................................................................. 16
   2.3.6 Zooming ............................................................................ 17
   2.3.7 Highlighting ....................................................................... 18
   2.3.8 Colour ............................................................................... 19
2.4 MULTIPLE VISUALIZATION TECHNIQUES ................................... 20
2.5 EXISTING REQUIREMENTS VISUALIZATION TOOLS ................. 23
   2.5.1 A Tool Suite for Aspect-Oriented Requirement Engineering (AORE) ................................................................. 23
   2.5.2 DOORS: A Requirement Management Tool ......................... 26
   2.5.3 Rational RequisitePro, Rational Software, IBM ..................... 29
   2.5.4 An Approach to Visualize and Reconcile Use Case Descriptions from Multiple Viewpoints(RECOCASE) ............................................ 32
   2.5.5 Model-Driven Visual Requirements Engineering: ANTHENA 34
   2.5.6 Analyst Pro Version 5.0 ......................................................... 36
   2.5.7 Borland® Caliber® Analyst ................................................. 38
   2.5.8 CRADLE Version 5.6 ........................................................... 41
2.6 COMPARISON ON EXISTING REQUIREMENTS VISUALIZATION TOOL .................................................. 43
2.7 THE IMPORTANCE OF REQUIREMENTS VISUALIZATION AT REQUIREMENTS ELICITATION STAGE ................................................... 45

CHAPTER 3 – ANALYSIS AND DESIGN .................................................. 47
3.1 ANALYSIS ................................................................................. 47
3.2 REQUIREMENTS FOR ReVIS TOOL ............................................ 49
3.3 ReVIS TOOL INTERFACE DESIGN ............................................. 51
   3.3.1 Organize ............................................................................ 51
   3.3.2 Visualize ............................................................................ 52
   3.3.3 Verify ................................................................................. 55
3.4 MINIMUM SYSTEM REQUIREMENTS .......................................... 55
List of Figures

Figure 1.1 : ReVis: Research methodology ......................................................... 6
Figure 2.1 : A schematic diagram of the visualization process ....................... 10
Figure 2.2 : Reference Model for Visualization ................................................. 11
Figure 2.3 : Node Link Diagram ............................................................................. 12
Figure 2.4 : TreeMaps ......................................................................................... 13
Figure 2.5 : Dynamic Queries: DC Home Finder .............................................. 15
Figure 2.6 : Cone Tree ........................................................................................ 16
Figure 2.7 : Pad++ - Directory Browser .............................................................. 17
Figure 2.8 : Colour-based highlighting ............................................................... 19
Figure 2.9 : Set of twelve colours in labelling ............................................... 20
Figure 2.10 : SpaceTree ..................................................................................... 21
Figure 2.11 : AORE Process ............................................................................. 23
Figure 2.12 : The Tool Suite ............................................................................... 24
Figure 2.13 : EA-Miner Tool ............................................................................. 26
Figure 2.14 : ARCADE ..................................................................................... 26
Figure 2.15 : DOORS: Project Interface ............................................................ 28
Figure 2.16 : DOORS: User Requirements Interface ....................................... 28
Figure 2.17 : DOORS: System Requirements Interface ................................... 29
Figure 2.18 : Rational Requisite Pro: Attribute Matrix View ....................... 31
Figure 2.19 : Rational Requisite Pro: Traceability Matrix View .................. 31
Figure 2.20 : Line or Hasse Diagram ................................................................. 34
Figure 2.21 : Overall requirement handling process ....................................... 35
Figure 2.22 : System architecture and processes ............................................. 36
Figure 2.23 : ANTHENA Tool ......................................................................... 36
Figure 2.24 : Analyst Pro: Requirements Elicitation Stage ............................................. 37
Figure 2.25 : Analyst Pro: Requirements Traceability Stage: ......................................... 38
Figure 2.26 : Analyst Pro: Requirements Traceability Analysis: ....................................... 38
Figure 2.27 : Borland® Caliber® Analyst: Requirements Elicitation Stage ..................... 40
Figure 2.28 : Borland® Caliber® Analyst: Requirements Validation Stage .......................................................... 40
Figure 2.29 : Borland® Caliber® Analyst: Requirements Traceability Stage ..................... 41
Figure 2.30 : Cradle Hierarchy Diagram shows the links between items for a particular requirements .......................................................... 42
Figure 2.31 : Cradle Workbench shows the Requirement Coverage Metric .......................................................... 43
Figure 3.1 : ReVis Model........................................................................................................ 50
Figure 3.2 : New Stakeholder tab design ........................................................................... 51
Figure 3.3 : New Requirement tab design ........................................................................... 52
Figure 3.4 : Requirement Details tab design ........................................................................ 52
Figure 3.5 : ReVis (a) screen design that display Project with three stakeholders ........................................................................................................... 53
Figure 3.6 : ReVis screen design that show Project with the stakeholders and related requirements ........................................................................................................... 54
Figure 3.7 : ReVis screen design that show Project with the stakeholders and related requirements and requirement flows ........................................................................................................... 54
Figure 3.8 : Search screen design. The user enter the string and the result will be highlighted on the tree. ........................................................................................................... 55
Figure 4.1 : An example of GraphML file created in ReVis ............................................ 58
Figure 4.2 : The relationship among tables in Stakeholder database .................................. 64
Figure 4.3 : Code to generate GraphML file ........................................................................ 65
Figure 4.4 : TreePlus code for interactive graph visualization ............................................ 66
Figure 4.5 : Revis Tool ......................................................................................................... 67
Figure 4.6 : Input – Project .................................................................................................... 68
Figure 4.7 : Input – Stakeholders ................................................................. 69
Figure 4.8 : Input – Requirements ............................................................... 69
Figure 4.9 : Input – Requirements Flow ..................................................... 70
Figure 4.10 : Save And Visualize ............................................................... 71
Figure 4.11 : Save And Visualize Dialog Box ............................................. 72
Figure 4.12 : Open And Visualize Dialogue Box ......................................... 72
Figure 4.13 : First Visualization Screen ..................................................... 73
Figure 4.14 : Lecturer Requirement ............................................................. 74
Figure 4.15 : By single clicking Display Statistic of analysis, the node got
focus (indicated by green background and thick border) and currently visible
nodes and not currently visible nodes previewed on the Preview Panel on the right. 75
Figure 4.16 : The refresh of Preview Panel once the user double clicked
“Display statistic of analysis” node ............................................................. 76
Figure 4.17 : Visualization of Requirements for Online Assessment
System ........................................................................................................ 77
Figure 4.18 : Zooming Slider ...................................................................... 78
Figure 4.19 : Fill in button .......................................................................... 78
Figure 4.20 : Search Function ..................................................................... 79
Figure 4.21 : Home button .......................................................................... 80
Figure 4.22 : Close Visualization Home button .......................................... 80
Figure 4.23 : Verify : Edit and Delete button ............................................... 81
List of Tables

Table 2.1  :  Comparison on each requirements tool in applying visualization techniques. .............................................................44
Table 3.1  :  Comparison on 4 Requirements Elicitation and Analysis Visualization Tools In Applying Visualization Technique ..........47
Table 3.2  :  Metis and Borland® Caliber® Analyst : Comparison on visual structures applied ......................................................................49
Table 3.3  :  Hardware and software requirement .............................................................55
Table 4.1  :  TreePlus API and Preferences class used in ReVis ............63
Table 5.1 :  Average Likert Scale Rating for Usability Issue in ReVis Tool, using the scale of 1 = Disagree Strongly, 6 = Agree Strongly .......................................................................................85
Table 5.2 :  Average Likert Scale Rating for Functionality Issue in ReVis Tool, using the scale of 1 = Disagree Strongly, 6 = Agree Strongly ............................................................................88
Table 5.3 :  Average Likert Scale Rating for Effectiveness Issue in ReVis Tool, using the scale of 1 = Disagree Strongly, 6 = Agree Strongly ............................................................................90
Table 5.4 :  Average Likert Scale Rating for Visualization Technique Rate, using the scale of 1 = Disagree Strongly, 6 = Agree Strongly .......................................................................................92
List of Graphs

Graph 2.1 : Breakdown of system errors ......................................................... 45
Graph 2.2 : Breakdown of system errors ......................................................... 46
Graph 5.1 : ReVis Tool : Usability Issue ......................................................... 86
Graph 5.2 : ReVis Tool : Functionality Study .................................................. 89
Graph 5.3 : ReVis Tool : Effectiveness Issue ................................................... 91
Graph 5.4 : ReVis Tool : Visualization Technique Rate ................................... 93
Graph 5.5 : ReVis Tool : Overall Result ........................................................... 94
CHAPTER 1 – INTRODUCTION

1.1 Overview

Requirements are statements that describe what a system must be able to do. Based on Kotonya and Sommerville (2003), “requirements are defined during the early stages of a system development as a specification of what should be implemented. They are descriptions of how the system should behave, application domain information, constraints on the system’s operation, or specifications of a system property or attribute.”

To specify the requirements to be met by a new system is a very challenging task. Any errors occur at this point will affect the implementation phase of the system. The specification of these requirements must contain sufficient information to enable stakeholders understand the business processes of the system to be developed and respond to it. It needs not be a very large document or too technical but it must reflect the real needs of the stakeholders. Requirements visualization is the field that can help the stakeholders to improve their understanding on the business processes of the system and yet they can easily verify it. Wikipedia (n.d) defined visualization as “any techniques for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an effective way to communicate both abstract and concrete ideas”. With requirement visualization each requirement can be effectively expressed to reduce the communication gap between the stakeholders and the developers and yet improve their understanding on the business processes of the system.
1.2 Problem Statement

Requirements specification has direct influence upon development, implementation and further progress. Primarily, the failure of the projects is caused by inadequate requirements specification.

Here are the problems arise in requirements specification that will affect the next stage of System Development Life Cycle.

1.2.1 The requirement specification delivered is not up to the quality required

Potter et al (1991) stated that, “Meeting user requirements of a software system has been a major challenge to software developers. Experience in a number of large projects reveals that a very large percentage of errors were made at earlier stages of their development. Therefore, it is a well accepted fact that it is crucial to express user requirements completely, correctly and unambiguously as possible”.

It is important to stress on the quality of requirement specification because the cost of fixing requirements error can be up to 100 times the cost of fixing a simple programming error (Kotonya and Sommerville 2003).

There are many consequences that will arise when requirements specification delivered is not up to the quality required.

a. Late delivery of the system and budget overrun.

b. Users’ dissatisfaction of the system due to its unreliability.

c. The costs of maintaining and changing the system are very high.
1.2.2 Problem of detailed requirements document

In order to develop the system, the developer will first understand the user needs which are a very challenging task. The developer can use many techniques to elicit the requirements. One of it is user’s requirements documents. It is not easy to analyze a very large user requirements documents. It is usually brought in different interpretation between the users and developers.

1.2.3 Lack of visualization tool in requirement elicitation stage

There are many requirements management tools that support all activities in requirement engineering process but only a few of them are specifically designed for requirements elicitation stage.

Many developers wrongly interpret the user requirements and then they come out with incorrect requirements specification. The verification of this requirements specification by the user is important to check level of understanding between users and developers before proceed to next stage.

1.2.3 No quick referencing to a particular requirement by a particular stakeholder

The existing requirements visualization tools in the market are mostly focusing on visualizing the requirements based on the requirements engineering processes. However, the tools are not mainly designed based on stakeholder preferences.
Due to the stated problems above, it is significant to build a requirement visualization tool that can organize, visualize and verify the requirements by using multiple visualization techniques which provide a referencing to a particular requirement by a particular stakeholder.

1.3 Research objectives
The objectives of this research are:

a. To develop a requirements visualization tool
b. To evaluate the usability, functionality and effectiveness of the tool

1.4 Scope of Research
The scope of the research is as follows:

a. Revis tool is expected to be used for organizing, visualizing and verifying requirements only.

b. Revis tool will organize the requirements based on the stakeholders.

c. Revis tool is targeted to be used by requirement engineers.

1.5 Research Methodology
For this research, the author has studied on requirements visualization, existing visualization techniques and the importance of requirements visualization at requirements elicitation stage. Besides that, the author also studied on existing requirement visualization tools and found out at which stage in requirement engineering processes the visualization tool is used and what visualization technique has been used. The purpose is to find the research gap. Based on that one of the stages in requirements engineering
processes will be chosen and existing visualization techniques will be applied.

Waterfall model is adopted to develop the tool. The model consists of a few stages: Analysis and Design, Implementation and Evaluation. Each stage has an outcome that feeds the subsequent stage.

In Analysis and Design stage, the author will discuss on the proposed visualization techniques and then come out with the tool architectural model and the proposed design. Next, in the implementation stage, the author will discuss on the development of the tool. After that, the evaluation of the tool will be discussed in Evaluation stage. Finally, the output of the research will be a requirements visualization tool, ReVis.

The research methodology diagram (Figure 1.1) is as follows:
Figure 1.1: ReVis: Research methodology
1.6 Dissertation Outline

This research consists of structured outlines as follows. Chapter 2 presents the overview of the requirements visualization, visualization techniques and requirements visualization tools that existed in requirements management environment. The analysis and design of the tool is discussed in Chapter 3. Chapter 4 and Chapter 5 describe the implementation and evaluation of the tool accordingly. Finally, this dissertation ends with Chapter 6 that discusses on a summary of ideas developed during the preparation of the dissertation and a proposal on relevant further research.
CHAPTER 2 – LITERATURE REVIEW

This chapter first reviews on requirements visualization, visualizations and its techniques. The next section discusses on existing requirements visualization tools techniques and comparison of each tool in applying visualization techniques. The chapter ends with the review of the importance of requirements visualization in requirements elicitation stage.

2.1 Requirement Visualization

For the past 25 years, software requirements have been identified to be a major problem (Lamsweerde 2000). Bell and Thayer (2000 cited in Lamsweerde 2000) through their empirical study perceived that problems with software requirements have a big impact on the quality of the software produced. The Standish Group (2000 cited in Lamsweerde 2000) reported that from a survey conducted over 8000 project by 350 US companies had revealed that one third of the projects were not completed and one half succeeded with partial functionalities, budget overruns and errors.

The major source of failure was because of poor requirements. European Software Institute (1996) on the European side had also reported a survey result from over 3800 organizations in 17 countries which indicated that most of the software problems are on requirements specification ( >50% ) and requirements management ( 50% ).

Visualization is seen as a technique to improve the requirements specification so that it is not likely to suffer with a lot of software requirements deficiencies such as requirements not complete and inconsistent. Dulac et al.(2002)
stated that the effective visualization can help express and describe the meaning and concepts and it is also possible to reduce the cognitive load of complex requirements specification by emphasizing on the related interactions and characteristics of the identified system.

2.2 Visualization Overview

Ward (2007) defined visualization as, “The graphical presentation of information, with the goal of providing the viewer with a qualitative understanding of the information contents. Information may be data, processes, relations or concepts. Graphical presentation may entail manipulation of graphical entities (points, lines, shapes, images, text) and attributes (colour, size, position, shape).”

Lamping (1997) defined visualization as, “A process of transforming information into a visual form enabling the user to observe the information”.

In computer science studies visualization uses computer graphics and image processing in which both are technologies. This means that visualization involves both visual computing and display, and human beings. Thus, human perception, cognitive capabilities, human variations and task characteristics need to take into consideration.

There are several advantages of visualization. Gray et al. (1997), highlighted some of the advantages below:

a. Ability to comprehend massive data.
b. Ability to reveal emergent properties.
c. Ability to see the problem exist on the data
d. Ease understanding for both large and small scale features of data.

e. Ease the hypothesis formation.

Visualization itself consists of a few stages. Ware (2000) listed that there are four basic stages in the process of data visualization, together with a number of feedback loops, which are the data collection and storage, the data processing and transformation, the graphics engine and the human information analyst. The following Figure 2.1 illustrates the four stages.

![Figure 2.1 : The visualization process diagram( Ware 2000 )](image)

The longest feedback loop involves gathering data itself as the analyst may choose to gather more data to follow up on an interesting lead. Another loop controls the computational preprocessing as the analyst may feel that if the data is subjected to a certain transformation prior to visualization, it can be persuaded to give up its meaning. Finally, the visualization process itself may be highly interactive.
Shneiderman (1992) developed a reference model for visualization in which
the core of this reference model is the mapping of Data into Visual Form.
Figure 2.2 below shows the diagram.

![Reference Model for Visualization](image)

Figure 2.2 : Reference Model for Visualization (Shneiderman 1992)

Figure 2.2 shows the series of data transformations that transform the data to
visual form and finally to human perceiver. The data transformations are Data
Transformation, Visual Mappings and Views. Data transformations map Raw
Data (in idiosyncratic format) into more structured relations called Data
Tables (metadata). Visual Mappings then transform Data Tables into Visual
Structures. Visual Structures are the structures with the combination of
substrates, marks and graphical properties that is used to encode
information. Finally View Transformations transform the Visual Structures
into Views by establishing graphical parameters such as clipping, zooming
and panning. This last stage increases the amount of information that can be
visualized.
2.3 Visualization Techniques

As visualization holds great promise for computational science and engineering, the review on visualization techniques is necessary. There are many techniques that can be used to transform data into visual form.

2.3.1 Node-link diagram

Node link diagram is an effective technique in visualizing tree by using connection. Shneiderman (1997) stated that it is a well-known technique for encoding network data and relationships between cases. Node link diagram is effective for trees that have uneven shape. Figure 2.3 below shows the node link diagram.

Figure 2.3 : Node Link Diagram (Shneiderman 1997)

The advantage of node link diagram is, it is easy to lay out and interpret since there can be no cycles and spatial axis which is normally used to separate levels in the tree. The disadvantage is as the trees get larger, they need an extreme ratio and they often contain considerable empty space as it requires a lot of space to visually organize the nodes.
2.3.2 TreeMap

TreeMap is a technique that uses enclosure to visualize trees. TreeMap enables a large amount of data to be displayed in a limited space. It maps hierarchical information onto a 2-D rectangular region in a space filling manner in which it makes maximum use of available display space (Johnson and Shneiderman 1991). With this technique, important information can be assigned with bigger display space while less important information can be assigned with less display space. The advantages of TreeMap are it presents overview of the entire large hierarchy of information and makes the navigation to any location in the space easier. Figure 2.4 illustrates the TreeMap technique.

Figure 2.4 : TreeMap (Johnson and Shneiderman 1991)
2.3.3 Dynamic Queries

The dynamic queries technique is a visual alternative to Structured Query Language (SQL) for querying database. It is a natural method for requesting data when the output is going to have visual form. Dynamic queries let users explore the database safely, playfully and rapidly specify the known-item queries (Shneiderman 1994).

Traditional dynamic queries in SQL require users to formulate queries in command language. They are powerful and expressive but most of the users face the difficulties to construct the queries. The problem is solved with these new dynamic queries technique in which it helps to formulate the queries and generate visual display of database search result.

Figure 2.5 shows the DC Home Finder dynamic query system that lets users construct the queries by adjusting the sliders.
2.3.4 Overview + Detail

Overview + Detail technique is used to handle a large scale of information. The technique is concerned with the use of interaction to display the whole data while tracking for the detailed analysis of the data (Card et al 1999). It provides multiple views, an overview for orientation, and a detailed view for further work. The advantages of overview + detail technique are it reduces searching for the data because the whole data can be shown at the same time or one at a time, allows detection of overall pattern and helps the user in choosing the next view and can access the details of the data rapidly. However, this design needs to consider the use of space.
2.3.5 **Focus + Context**

Focus + Context is another technique developed that allows users attempt to use more of the display resource to correspond to interest of the user’s attention. It is developed because of these three arguments made by the user: the need of concurrent overview (context) and detail (focus) information, the need of different information for overview and detail and the need of single or dynamic display as in human vision (Card et al 1999).

This technique overcomes the limitation of the computer display. More resources are available to the detailed processing and at the same time more relevant context are also available. Filtering, highlighting and distortion are all the areas that can be obtained in Focus + Context. Figure 2.6 shows Cone tree that applies Focus + Context.

![Figure 2.6: Cone Tree (Card et al 1999)](image)
2.3.6 Zooming

Zooming is a visualization technique that deals with viewing information at different scales to see the overview or details of the information. Bederson and Holland (1994) in their research on graphical interface based on zooming stated that,

“If the interface designers are to move beyond windows, icons, menus, and pointers to explore the larger space or interface possibilities, new interaction techniques must go beyond the desktop metaphor”.

The new technique mentioned in their research is zooming technique. It lets users pan around the interface and zoom out for an overview of the information and zoom in for more details information. Zooming makes better use of screen space. Figure 2.7 shows snapshots of Pad++ that applies zooming technique.

![Figure 2.7 : Pad++ - Directory browser (Bederson and Holland 1994)]
2.3.7 Highlighting

Highlighting is the process of emphasizing information. It is tightly coupled with the specific task and context. Highlighting is also a special kind of micro-macro reading that made individual items visually distinctive in some way (Card et al 1999). There are number of styles for visual highlighting. Robinson (2006) highlighted some of the styles below:

a. Colour-based highlighting causes the data elements filled with the selected colour or obviously outlined. It can also be extended by including the soft edges and effect to create the real world highlighting. Colour-based highlighting is the common highlighting style.

b. Leader lines are lines that are dynamically drawn based on cursor position over the data elements. They manipulate stroke style and colour.

c. Depth of field creates areas of contrasting sharpness to data elements by changing the areas in terms of in and out of focus.

d. Transparency highlights the interest object by dissolving the context (by increasing the transparency of the context).

e. Contour lines shaped the data items by changing number of contours, the stroke style, colour and the contours distance. The contoured data will be easily distinguished by the users because their appearances are more “higher” than those data that are not contoured.

Example of colour-based highlighting is shown in Figure 2.8.
2.3.8 Colour

Colour is used to convey more information. Colour can be a very powerful design element. It can create an emotional effect and also can convey a particular message. Many applications use colour techniques to attractively display information and improve task performance.

Colour in visualization can be applied in many application areas. Ware (2000) presented some of the areas below:

a. Colour specification interface in which the user can be given a set of controls to specify a point in three dimensional colour space, a set of colour name to choose from, or a palette of predefined color sample.

b. Colour for labeling in which the distinctness, unique hues, contrast with background and colour blindness are among the perceptual factors that need to be considered.
c. Colour sequence for map in which the Pseudocolouring technique is used to reveal the shape of surfaces.

d. Colour reproduction in which is the colour appearances from one display device such as a computer monitor are transferred to another device such as paper. Calibration, range scaling, rotation and saturation scaling are the methods applied for the transformations.

Figure 2.9 below shows the twelve colour used in labeling.

![Figure 2.9: Set of twelve colours in labeling (Ware 2000)](image)

2.4 Multiple Visualization Techniques

Beard et al (1990) in their research stated that multiple visualizations help in creating more powerful information exploration environment by combining the strength of each visualization technique.

Three researchers, Catherine Plaisant, Jesse Grosjean and Benjamin B. Bederson from University of Maryland have built one novel interface that coordinates multiple visualizations. The novel interface was named as SpaceTree.
SpaceTree is “a novel interface that combines the conventional layout of trees with a zooming environment that dynamically lays out branches of the three to best fit the available screen space. It also uses preview icons to summarize the topology of the branches when there is not enough space to show them in full” (Plaisant et al 2002).

As stated by Plaisant et al (2002), SpaceTree (Figure 2.10) or also known as a novel tree browser is built on the conventional node link diagram. It applies a dynamic rescaling of tree branches to best fit the space screen, optimizes the camera movement and uses preview icons to summarize the branches that cannot be extended. Space Tree includes search and filter functions.
SpaceTree applies two techniques: space filling and node link technique that combine TreeMap and node link diagram. It also combines dynamic queries and overview + detail technique. The details of the description are as follows (Plaisant et al 2002):

a. Navigation by users on the tree is done by either clicking on the nodes to expand the tree or using arrow keys to navigate from one node to another (siblings, ancestors and descendants). (Node link diagram technique and TreeMap technique)

b. Search and filter functions are supported where users can navigate through the highlighted path resulted from the search and filter process. (Dynamic Filtering & Queries technique)

c. Pruning of the tree can be conducted by using the slider. (Focus + Context and Dynamic Queries technique)

d. Zooming using data aware zooming controls can be done when users click on the node. (Zooming technique)

SpaceTree is the attempt made by these three researchers to enhance the use of traditional node link representation. The combination of multiple visualization technique (TreeMap to cover the limitation of node link diagram, Focus + Context, Dynamic Queries, filtering and zooming technique) gives more interactivity to the users.
2.5 Existing Requirements Visualization Tools

The following is the review on 8 existing requirements visualization tools:

2.5.1 A Tool Suite for Aspect-Oriented Requirement Engineering (AORE)

As stated by Chitchyan et al (2006), "Aspect-Oriented Requirement Engineering (AORE) has emerged as a new way to modularize and reason about crosscutting concerns during requirements engineering". Based on Chitchyan et al (2006), "AORE supports the identification of crosscutting, aspectual requirements and their effect on other requirements of the system". Figure 2.11 shows the AORE process.

![Figure 2.11: AORE Process (Chitchyan et al 2006)](image)

A tool suite for AORE has been developed consists of five integrated tools: WMATRIX, Early Aspect Mining (EA-Miner), Requirements Analysis Tool (RAT), Key Word In Phrase (KWIP) and Aspectual Requirements Composition and Decision Support (ARCADE) (Figure 2.12).
TWO types of roles in the tool suite (Chitchyan et al 2006):

a. Information Generator Tools

i.  **WMATRIX** is a collection of corpus-based natural language processing tool. It is a web-based tool. WMATRIX accepts the gathered information documents from users. It extracts verbs and nouns from text of the document and assigns the part-of-speech tag to each word in the document. Then the words are grouped with multi-word expression by using the semantic analysis and category of the words is selected.

b. Information Consumer Tools

There are four Information Consumer Tools:

i.  **EA-Miner tool** utilizes information produced by WMATRIX and deals with both aspectual and non-aspectual concerns.

ii.  **RAT** uses the statistical information produced by WMATRIX to find the requirements containing the statistical word. Both EA-Miner and RAT are for concern and relationship identification.
iii. **KWIP Tool** supports the manual screening out of redundant concerns and requirements.

iv. **ARCADE** is a XML representation that helps in specifying viewpoints, aspects and associated composition rules. The ARCADE mechanisms are to see how aspects requirements influence viewpoint requirements and to identify the overlapping between aspects requirements and viewpoints requirements.

From these five tools, EA-Miner and ARCADE apply dynamic queries technique. Figure 2.13 and Figure 2.14 show the interface of EA-Miner and ARCADE.
2.5.2 DOORS: A Requirement Management Tool

Dynamic Object Oriented Requirements System (DOORS) is “a multi-platform, enterprise-wide requirements management tool” (Telelogic 2006). DOORS functionalities are (Telelogic 2006):
a. to capture, link, trace, analyze and manage a wide range of information to ensure a project’s compliance to a specified requirements and standards.

b. to provide a communication of business needs, allow cross-functional teams to work together on the projects development to meet the needs and enables validation that the right system is being built.

DOORS Architecture consists of:

a. Project – a folder that contains all data for a particular project such as requirements, design, development, test, production and maintenance.

b. Folder – used to organize data. It may contain other folders, projects or modules and has access control.

c. Modules – to store the information and acts as containers for data sets.

There are FOUR classes of modules:

i. Formal – the most used type of module for structured sets of information

ii. Descriptive – unstructured source information

iii. Link – contains relationships between other data elements.

iv. Object – It can be a block of text, a graphic image or a spreadsheet. In formal modules, data is stored in objects.

Below in Figure 2.15, Figure 2.16 and Figure 2.17 are screenshots of DOORS.
Figure 2.15: DOORS: Project Interface (Telelogic 2006)

Figure 2.16: DOORS: User Requirements Interface (Telelogic 2006)
To visualize the complex requirements, DOORS apply several visualization techniques such as hierarchical tree, matrix and focus + context as shown above.

### 2.5.3 Rational RequisitePro, Rational Software, IBM

Rational RequisitePro is the requirement management tool provided by IBM. RequisitePro enables users to organize, prioritize, trace relationships and easily track changes to the requirements by integrating Microsoft Word with multi user database. It combines both document-centric and database-centric approaches.

Rational RequisitePro functionalities are (IBM 2003):

a. to maintains documents with the requirements linked to a database dynamically.
b. to identify the impact of change with traceability features and analysis queries.

c. to integrate requirements with other lifecycle artifacts and processes.

Rational RequisitePro methodology consists of:

a. Plan the project

It involves the activities of identifying the Requirement Type (FETA, FETA, UC, RMP, STRQ, SUPL), Requirement Attribute (Priority, Status), Document Type (eg: STRQ in document type STR) and Revision history.

b. Gather, Organize and Document Requirements

It involves the activities of representing the requirements in View, requirements in Hierarchy and requirements in a Document. This can be seen in Figure 2.18: An Attribute Matrix View.

Figure 2.18: Rational RequisitePro: Attribute Matrix View (IBM 2003)
c. Manage the Requirements Traceability relationships

It involves a traceability matrix. Traceability is a methodical approach used to manage change by linking requirements that related to each other. The traceability matrix can be seen in Figure 2.19: Traceability Matrix View and Traceability Tree View.

![Traceability Matrix View](image)

**Figure 2.19 : Rational RequisitePro: Traceability Matrix View (IBM 2003)**

d. Communicate the requirements

It supports a team communication such as a discussion channel via e-mail.

As what can be seen above, Rational Requisite Pro applies visualization techniques to gather, organize and manage the requirements. One of common visualization techniques used is tree. Besides that, it also applied iconic, colour and matrix or table.
2.5.4 An Approach to Visualize and Reconcile Use Case Descriptions from Multiple Viewpoints (RECOCASE)

The purpose of the RECOCASE is to capture use case descriptions from multiple viewpoints and automatically generates a visualization of the individual and shared viewpoints to assists identification and resolution of conflicts.

The six iterative phases of the RECOCASE viewpoint methodology are (Richards and Aquilera 2003):

- **a.** Requirement acquisition - a brainstorming session on use case diagram led by group facilitator (GF).

- **b.** Requirement translation - the process of parsing the sentences of the use case descriptions into noun and verb phrases by using natural language techniques.

- **c.** Concept generation - the process of generating concepts based on the words and phrases in the use case description and visualizing it into a diagram called a Line or Hasse Diagram (Refer Figure 2.6). Formal Concept Analysis (FCA) was applied.

- **d.** Concept comparison and conflict detection - the process that involves the selection of the sentences to be included, compares the viewpoints and detects the conflicts of the viewpoints. Use case sentences that have same concepts but have been expressed differently will share same node. All group members and the group facilitator involved.
e. Negotiation and Reconciliation - During the reconciliation process, each sentence in sentence number order will be reviewed by GF and group members. There are three terms used during the process. “Ignore” is used when none of the members in the group agree with the sentences. "Delay" is used when they do not agree or not sure of the sentences. “Done” is used when the sentences were already covered before. Sentences that are agreed to be included will be added to the final use case description.

f. Evaluation – Graph theory on lattices is used to evaluate if there are still any similarities on the final viewpoints and if a new round of negotiation is necessary.

Figure 2.20 below shows Line or Hasse Diagram that is used to visualize the concept generated based on words and phrases in the use case description.
In RECOCASE, node link diagram and colour techniques is used to visualize
the requirements into Line or Hasse Diagram. Besides that, RECOCASE also
uses graph or lattices during the evaluation phase.

2.5.5 Model-Driven Visual Requirements Engineering : ANTHENA

The ANTHENA describes the requirement handling process of a set of
research projects (ANTHENA IP) and how the process is supported by
modeling and visualization requirement structures. ANTHENA applies an
Active Knowledge Modeling (AKM) and the Dynamic Requirements
Definition System (DRDS).

The stakeholders register the requirements through web interface and then
the managers or developers utilize and integrate modeling tool for visual
classification, analysis, elicitation, and selections of requirements.

The main parts of the ANTHENA system are (Solheim et al 2005):

a. Web user interface - used to register the requirements.
b. Requirement Database – used for storing all the requirements
   information.
c. Visual model - Metis is chosen as visual modeling tool. The tool is
   used to visualize and organize the requirements.

Below are the ANTHENA overall requirement handling process that is shown
in Figure 2.21 and its system architecture in Figure 2.22.
Figure 2.21: Overall requirement handling process (Solheim et al 2005)

Figure 2.22: System architecture and processes (Solheim et al 2005)

Figure 2.23 below shows the screenshot of ANTHENA tool that uses visual modeling tool, Metis to visualize and organize the requirements. Metis applies tree structure, matrix and dynamic queries technique.
2.5.6 **Analyst Pro Version 5.0** (Analyst Pro 2006)

Analyst Pro (ARTS) is a user friendly requirement management tool developed by Goda Software, Inc. The tool is set up as an intelligent tool to deal effectively with the complexities of system development. It incorporates all features in requirement management including requirement specification and tracking, requirement repository, requirement traceability and importing and exporting with the external systems.

Figure 2.24, 2.25 and 2.26 below are the screenshots of the tool that show how requirements are visualized.
Figure 2.24: Analyst Pro: Requirements Elicitation Stage (Analyst Pro 2006)

Figure 2.25: Analyst Pro: Requirements Traceability Stage (Analyst Pro 2006)
Generally, this user friendly user visual tool visualize the requirements by applying hierarchical tree, overview + detail, colour and dynamic queries.

2.5.7 **Borland® Caliber® Analyst** (Borland 2006)

Borland® Caliber® Analyst is a requirement management tool developed by Technology Builders, Inc. Borland® Caliber® Analyst is an enterprise product suite that integrates two tools, Borland® Caliber® DefineIT™ and Borland® CaliberRM™. The first tool is a software requirement definition system that is used at early stage of a new project or enhancement activity. The outcome of the tool is the accurate and complete software requirement definition. The second tool is requirements management software. This tool helps to ensure that the applications developed will fulfill users' needs. The requirements
management software helps in managing expectation, collaboration, impact analysis and communication in software development life cycle.

The features in the tool provide many benefits. There are visual scenarios that display captured and specified requirements, industry-unique storyboard execution that is used for requirements validation, auto-generated test cases that publish and synchronize the requirements and traceability diagram for managing requirements changes.

The followings screenshots in Figure 2.27, 2.28 and 2.29 show the features in Borland® Caliber® Analyst.
Figure 2.28: Borland® Caliber® Analyst: Requirements Validation Stage

(Borland 2006)

Figure 2.29: Borland® Caliber® Analyst: Requirements Traceability Stage

(Borland 2006)
Borland® Caliber® Analyst applies multiple visualization techniques on each stage of requirements specification started from requirements elicitation, requirements validation and requirements validation. There are hierarchical tree, overview + details, diagram, and node link diagram.

2.5.8 CRADLE Version 5.6 (CRADLE 2006)

CRADLE Version 5.6 is a requirement management tool delivered by Structured Software System Ltd. It is fully a web based tool that incorporates the whole project stages, activities and deliverable in a framework.

CRADLE provides requirements capture facility in its requirements gathering stage. This facility able to examine the customer statements, get the requirements and make a cross reference back to the actual document. It is integrated with Word, Excel, PDF and all text formats.

CRADLE also supports tool to identify the duplication, omission, ambiguity and conduct coverage analyses. Then Non Functional Requirement Graphs (NFRGs), Hierarchy Diagrams are applied to visualize linkage in the requirements sets. The hierarchy diagrams apply a node link diagram which is a hierarchical tree. This can be seen in Figure 2.30.
Besides the node link diagram, CRADLE also applies a hierarchical tree, matrix, iconic and colour in its requirement coverage metric as shown in Figure 2.31: CRADLE Workbench shows the Requirement Coverage Metric.
2.6 Comparison on Existing Requirements Visualization Tool

Many of the tools described above are requirement management tools that apply visualization techniques in a few stages in requirement specification stages.

Table 2.1 compares the tools based on the stages in which requirements visualization applies and what techniques is used in the tool.
<table>
<thead>
<tr>
<th>REQUIREMENT STAGE</th>
<th>TOOLS</th>
<th>EA Miner / ARCADE</th>
<th>DOORS</th>
<th>REQUISITE PRO</th>
<th>RECOCASE</th>
<th>METIS</th>
<th>ANALYST PRO</th>
<th>BORLAND</th>
<th>CRADLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENT ELICITATION</td>
<td></td>
<td></td>
<td></td>
<td>Tree</td>
<td>Matrix</td>
<td>Focus + Context</td>
<td>Tree</td>
<td>Dynamic Queries</td>
<td>Matrix</td>
</tr>
<tr>
<td>REQUIREMENT ANALYSIS</td>
<td></td>
<td></td>
<td>Node Link Diagram</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQUIREMENT VALIDATION</td>
<td></td>
<td></td>
<td>Graph Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQUIREMENT TRACEABILITY</td>
<td></td>
<td></td>
<td>Tree</td>
<td>Matrix</td>
<td>Iconic</td>
<td>Colour</td>
<td>Tree</td>
<td>Overview + Detail</td>
<td>Dynamic Queries</td>
</tr>
<tr>
<td>REQUIREMENT VERIFICATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQUIREMENT SPECIFICATION</td>
<td></td>
<td></td>
<td>Dynamic Queries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQUIREMENT TRACKING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of each requirements tool in applying visualization techniques
Based on the comparison above, many of the requirement visualization tools use tree techniques to visualize the requirements. 6 tools out of 9 tools use tree technique to visualize requirements elicitation, requirements analysis and requirements traceability stage. However, the tools not only apply the tree techniques but they are also applied multiple visualization techniques to improve their requirements visualization.

From 8 tools, only 2 tools give a specific focus on requirements elicitation stage.

2.7 The importance of Requirements Visualization at Requirements Elicitation Stage

Rajagopal et al (2005) stated that “requirements elicitation stage is both the hardest and most critical part of software development, since errors at this beginning stage propagate through the development process and the hardest to repair later”. Studies by Beichter et al (1984 cited in Rajagopal et al 2005) show that inadequate system specification brings to 70% of system error and the design errors brings to 30% of system errors. Graph 2.1 illustrates it.

Graph 2.1 : Breakdown of system errors (Beichter et al 1984 cited in Rajagopal et al 2005)
The SEI National Software Capacity Study (1990) also indicates that inadequate system specification is a major factor that contributes to the system errors. Graph 2.2 illustrates it.

Graph 2.2 : Breakdown of system errors (SEI 1990)

From the above studies, it shows that the incorrect requirements specification will lead to the development of deficient system, increases the cost of its development or even causes the project to fail. All these are related with the requirements gathered during requirements elicitation stage.

Therefore, it is crucial for the stakeholders to understand and satisfy the requirements of the planned system. This means that the requirements must be described in a form that the stakeholders can clearly understand and easily verify it. Visualization techniques appear as a useful tool to help in these processes.
CHAPTER 3 – ANALYSIS AND DESIGN

This research used Waterfall Model as a methodology to develop the proposed tool, ReVis. Chapter 3 will focus on Analysis and Design of the proposed tool.

3.1 Analysis

As in Chapter 2: Table 2.1 shows that 4 out of 8 tools visualize the requirement elicitation and analysis stage. Below in Table 3.1 is an extract of the table.

<table>
<thead>
<tr>
<th>REQUIREMENT STAGE</th>
<th>TOOLS</th>
<th>DOORS</th>
<th>REQUISITE PRO</th>
<th>METIS</th>
<th>BORLAND® CALIBER® ANALYST</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUIREMENT ELICITATION</td>
<td></td>
<td>Tree</td>
<td>Tree</td>
<td>Tree</td>
<td>Diagram ( Scenario Diagram )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Matrix</td>
<td>Dynamic Queries</td>
<td></td>
<td>Overview + Detail</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Focus + Context</td>
<td>Matrix</td>
<td></td>
<td>Dynamic Queries</td>
</tr>
<tr>
<td>REQUIREMENT ANALYSIS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Tree</td>
</tr>
</tbody>
</table>

Table 3.1: Comparison on 4 Requirements Elicitation and Analysis Visualization Tools in applying visualization techniques.

The analysis on the tools was made and the author found out that:

a. All four tools are managing and visualizing functional requirements.
b. Two out of four tools, METIS and Borland® Caliber® Analyst are specifically managing and visualizing requirements in requirements elicitation stage. While another two tools, Requisite Pro and DOORS are combining two stages which are requirements elicitation and requirements analysis.

c. As the research objective is to visualize requirements in the requirement elicitation stage, METIS and Borland® Caliber® Analyst are chosen.

d. METIS uses tree which is similar to folder tree, dynamic queries and matrix. Borland® Caliber® Analyst uses a folder tree, dynamic queries, overview + detail and a diagram called Scenario / Activity Diagram to visualize the requirements.

e. As the research objective is also to organize and visualize the requirements based on stakeholders, none of these two tools (METIS and Borland® Caliber® Analyst) visualize requirements according to stakeholders.

f. No verification is made at this stage after the requirements were visualized.

For more details, the author had made a comparison on both tools based on the visualization techniques applied on them.

Table 3.2 shows the comparison.
Table 3.2: Metis and Borland® Caliber® Analyst: Comparison on visual structures applied

Due to that, the author summarized that the three visual structures when combine together will give a novel interface as what applied by Borland® Caliber® Analyst.

3.2 Requirements for ReVis Tool

After the analysis made, ReVis is proposed. ReVis will be a tool to organize, visualize and verify the requirements at a very early stage of requirements elicitation stage based on the stakeholders. Figure 3.1 shows ReVis Model.
There are 3 requirements in ReVis Model:

a. **Organize**

ReVis shall be able to organize the requirements by allowing:

- Add project, stakeholder, requirement and requirement flow.
- Delete project, stakeholder, requirement and requirement flow.

b. **Visualize**

ReVis shall be able to visualize the requirements by applying multiple visualization techniques: Node Link Diagram, TreeMaps, Dynamic Queries, Focus + Context, Overview + Detail, zooming, highlighting and colour.

c. **Verify**

ReVis shall be able to allow stakeholders to verify the requirements by allowing:

- Edit project, stakeholders, requirements and requirements flow.
- Delete project, stakeholder, requirement and requirement flow.
The main point here is all requirements gathered during the requirements elicitation stage shall be organized, visualized and verified by ReVis.

3.3 ReVis Tool Interface Design

As mentioned in previous section, ReVis is divided into three requirements.

3.3.1 Organize

The user will first enter the requirements details such as stakeholder, requirements of each stakeholders and requirements flows. Below are the three screen designs for the purpose of it.

Figure 3.2 : New Stakeholder tab design
3.3.2 Visualize

The user can click the menu, Visualize and ReVis to visualize the details entered. This is the main part of the tool. The stakeholders and details of requirements are visualized by applying SpaceTree interface that contains Node link diagram, TreeMaps, Dynamic and Queries, Focus + Context and
zooming technique. Nodes are used to represent stakeholder, requirement and requirement flow. The user can double click the node to see the details.

Below are the screen designs (Figure 3.5, Figure 3.6, Figure 3.7 and Figure 3.8) with the details visualization techniques that will be applied:

Figure 3.5: ReVis (a) screen design that display Project with three stakeholders.
Figure 3.6: ReVis screen design that show Project with the stakeholders and related requirements.

Figure 3.7: ReVis screen design that show Project with the stakeholders and related requirements and requirement flows.
Figure 3.8: Search screen design. The user enters the string and the result will be highlighted on the tree.

### 3.3.3 Verify

After the requirements are visualized, the stakeholders will be asked to verify it. The stakeholders can edit any incorrect requirements.

### 3.4 Minimum System Requirements

The minimum system requirement of ReVis 1.0 is as stated in Table 4.1.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Pentium III Processor</td>
<td>Microsoft Visual Studio 2005 (Ms C#.Net)</td>
</tr>
<tr>
<td>128MB Memory</td>
<td>Microsoft SQL Server 2005</td>
</tr>
</tbody>
</table>

Table 3.3: Hardware and software requirement
CHAPTER 4 : IMPLEMENTATION

The chapter discusses on the third stage of Waterfall model: implementation of the system. It focuses on the development of the process model of the system, the tool used to develop and the interactive graph visualization applied.


There are four important components involved in implementing the system. There are Microsoft Visual C#.NET as the interface, Microsoft SQL Server Express Edition as the database, GraphML file as the data file and TreePlus as a component use to visualize the data into interactive graph visualization.

4.1.1 Microsoft Visual C#.NET

C# (pronounced C-sharp) is a new object oriented language from Microsoft and part of Visual Studio .NET platform. C# is derived from C and C++ and also borrows a lot of concepts from Java. C# is the only language designed specifically for .NET and suitable for all applications.

For the development of the system, Visual C#.NET 2005 is used as an interface of the system as it gives many advantages such as manage complexity of a program and provide simplified development and deployment of the system.
4.1.2 Microsoft SQL Server 2005 Express Edition

SQL Server 2005 Express Edition is the next version of MSDE. It is a simple but powerful development environment for building data-driven applications. SQL Server 2005 Express Edition is a free, lightweight, and embeddable version of SQL Server 2005. It is easy for new developers to use immediately. The database is easily managed with SQL Server 2005 Management Studio Express. The design schemas, add data and query local databases are all inside the Visual Studio 2005 environment.

This Microsoft SQL Server Express Edition is used to create a database for ReVis. All data gathered during requirement elicitation stage will be stored in this database. For that purpose, the database named Stakeholder is created to store all the data.

4.1.3 GraphML

Before visualizing the requirements, the data stored in Stakeholder database must be converted or formatted into GraphML file format. Figure 4.1 shows the gathered requirements formatted into GraphML file.

GraphML is a comprehensive and easy-to-use file format for graphs. It is a markup language to describe the structural properties of the graph. Its main features include support of directed, undirected and mixed graphs, hierarchical graphs, hypergraphs and graphical representations. The GraphML syntax is defined by the GraphML Schema.(Refer Appendix A).

Figure 4.1 below shows an example of GraphML file created in ReVis.
Figure 4.1: An example of GraphML file created in ReVis
This GraphML file above consists of graphml element and a variety of sub elements: graph, node and edge. These are the main structural elements of GraphML.

4.1.3.1 The header

The first line of the file is an XML process instruction which defines that the file adheres to the XML 1.0 standard. The encoding of the file is UTF-8 which is the standard encoding for XML documents.

The second line contains the graphml element, <graphml> which is the root-element element of any GraphML files. This graphml element belongs to the namespace http://graphml.graphdrawing.org/xmlns. The namespace is defined as a default namespace by adding the XML attribute xmlns=http://graphml.graphdrawing.org/xmlns to the file.

4.1.3.2 The Graph

The graph is denoted by a graph element, <graph>. Nested inside a graph elements are a node element, <node> and an edge element, <edge>.

A graph element, <graph> contains two basic attributes, id and edgedefault. This can be seen through the statement: <graph id="G" edgedefault="directed">. id is a XML Attribute that is used to specify an identifier for the graph. The identifier, “G” is used to reference the graph. edgedefault is a XML Attribute that is used to declare a default direction of the graph element. The two possible values are directed and undirected. The “directed” is used in this file.
A node element, `<node>` contains a XML attribute id that is used to specify the identifier of the node. The identifier defined must be unique within the entire file. As shown in Figure 4.1 above, the number is used as an identifier because of huge data.

An edge element, `<edge>` contains two endpoints that is defined using XML attributes `source` and `target`. Example: `<edge source="0" target="1"/>` where 0 and 1 are node identifier.

### 4.1.3.3 A GraphML-Attribut and Value

A GraphML-Attribute is defined by a `key` element that specifies the `identifier`, `name`, `type` and `domain` of the attribute. From Figure 4.1, the identifier is specified by the XML-Attribute id and is used to refer to the GraphML-Attribute inside the file. `nodename` is used as the id.

The name of GraphML-Attribute is defined by the XML-Attribute `attr.name`. The purpose of it is to allow the applications identify the meaning of the attribute. `attr.name` must be unique but it is not used inside the file.

The type of GraphML attribute is defined by the XML-Attribute `attr.type`. The type can be either string, Boolean, int, long, float or double. For this file, string is used as all data are in string. For the domain, possible values are graph, node, edge and all and it specifies for which graph elements the GraphML-Attribute is declared.
A GraphML-Attribute value for a graph element is defined by a data element nested inside the element for the graph element. The data element has an XML-Attribute key that refers to the identifier of the GraphML-Attribute. The value is the text content of the data element that must be of the type declared in key definition which is string.

4.1.4 TreePlus

Once the GraphML file is ready, the TreePlus component is used for visualization. TreePlus was built as a UserControl in C# with Piccolo.NET. Piccolo.NET is a C# based engine that supports 2D visualizations and it is designed to support zoomable information spaces.

TreePlus was inspired by the success of the earlier tool developed, SpaceTree (as discussed in Chapter 2) and TaxonTree. As a User Control in C#, TreePlus is pluggable into any .NET application, provides an application programming interface (API) and fires events to communicate with the containing application. TreePlus consists of 15 classes and about 7000 lines. (Refer Appendix B)

TreePlus is also known as an interactive graph visualization component and based on tree-style layout. It is created by Bongshin Lee from University Of Maryland, USA in 2006. TreePlus enables users to iteratively explore a graph by starting at a node and then incrementally expanding and exploring the graph.
As stated by Bongshin Lee (2006), TreePlus transforms a graph into a tree plus cross links (i.e. the additional links that are not represented by the spanning tree) and uses visualization, animation and interaction techniques to reveal the graph structure while preserving the readability of the labels.

Based on Chapter 3: Analysis and Design, TreePlus was chosen as a suitable component to visualize the requirements as designed. It includes all visualization techniques proposed: Node Link Diagram, TreeMaps, Dynamic and Queries, Focus + Context, Overview + Details, zooming and highlighting.

The implementation involved 3 libraries. There are:

a. TreePlusControl.dll contains TreePlus API (Application Programming Interface) and preferences class that support rapid creation of graph visualization system. Table 4.1 shows TreePlus API and Preferences class that had been used in ReVis.

<table>
<thead>
<tr>
<th>TreePlus API</th>
<th>Preferences Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void Home()</td>
<td>public int LevelSeparation{ }</td>
</tr>
<tr>
<td>public void Search()</td>
<td>public int BorderWidth{ }</td>
</tr>
<tr>
<td>public void ShowGraph()</td>
<td>public int LevelSeparation{ }</td>
</tr>
<tr>
<td>public void FitinWindow()</td>
<td>public int TickSize{ }</td>
</tr>
<tr>
<td>public void Zoom()</td>
<td>public int MaxWidth{ }</td>
</tr>
<tr>
<td></td>
<td>public string OrderBy{ }</td>
</tr>
<tr>
<td></td>
<td>public string SortOrder{ }</td>
</tr>
<tr>
<td></td>
<td>public bool ShowLegend{ }</td>
</tr>
</tbody>
</table>
Table 4.1: TreePlus API and Preferences class used in ReVis

b. GraphLibrary.dll that contains data structures and basic operations related to graphs. It also contain some utility classes for specifying preferences and non-graph related operations

c. UMD.HCIL.Piccolo.dll and UMD.HCIL.PiccoloX.dll. These two namespaces are needed as TreePlus was created with Piccolo.NET. Piccolo.dll and PiccoloX.dll are the two root namespaces in for all Piccolo classes. It contains the core scenegraph classes and Activities, Event, Nodes and Util namespaces that are used to build Piccolo applications.

4.2 System Implementation

4.2.1 The Stakeholder database

There are four tables implemented in Stakeholder database. There are projects table to save all project details, stakeholders table to save all stakeholder details, requirements table to save all requirements details and reqdetails table to save all requirements flow details. Figure 4.2 shows the relationship among the tables.
4.2.2 The GraphML file

After obtaining all information and stored in database, GraphML file can be generated. The code to implement this conversion is as shown in Figure 4.3.
private void WriteXML()
{
    saveFileDialog1.Filter = "GRAPHML|*.graphml";
    saveFileDialog1.Title = "ReVis - Save And Visualize";
    saveFileDialog1.ShowDialog();

    if (saveFileDialog1.FileName != "")
    {
        try
        {
            //pick whatever filename with .xml extension
            string filename = saveFileDialog1.FileName;
            XmlDocument xmlDoc = new XmlDocument();
            XmlTextWriter xmlWriter = new XmlTextWriter(filename, System.Text.Encoding.UTF8);
            xmlWriter.Formatting = Formatting.Indented;
            xmlWriter.WriteProcessingInstruction("xml", "version='1.0'
            encoding='UTF-8'");
            xmlWriter.WriteDocType("graphml", null, null, null);
            xmlWriter.WriteStartElement("graphml");
            xmlWriter.WriteStartElement("key");
            xmlWriter.WriteAttributeString("id", "nodename");
            xmlWriter.WriteAttributeString("for", "node");
            xmlWriter.WriteAttributeString("attr.name", "nodename");
            xmlWriter.WriteAttributeString("attr.type", "string");
            xmlWriter.WriteEndElement();
            xmlWriter.WriteStartElement("graph");
            xmlWriter.WriteAttributeString("id", "G");
            xmlWriter.WriteAttributeString("edgedefault", "directed");
            foreach (DataRow dtRow in dTable.Rows)
            {
                //<node id="0">
                xmlWriter.WriteStartElement("node");
                xmlWriter.WriteAttributeString("id", dtRow["id"].ToString());
                xmlWriter.WriteStartElement("data");
                xmlWriter.WriteAttributeString("key", "nodename");
                xmlWriter.WriteString(dtRow["elementname"].ToString());
                xmlWriter.WriteEndElement();
                //</node>
                xmlWriter.WriteEndElement(); //node
            }
            foreach (DataRow dtRow in dTable.Rows)
            {
                if (dtRow["id"].ToString() != "0")
                {
                    xmlWriter.WriteStartElement("edge");
                    xmlWriter.WriteAttributeString("source", dtRow["ownerid"].ToString());
                    xmlWriter.WriteAttributeString("target", dtRow["id"].ToString());
                    xmlWriter.WriteEndElement();
                }
            }
            xmlWriter.WriteEndElement(); //graph
            xmlWriter.Close();
        }
        catch (Exception ex)
        {
            WriteError(ex.ToString());
        }
        else
        {
            MessageBox.Show("Enter Filename");
        }
    }
}

Figure 4.3: Code to generate GraphML file
4.2.3 TreePlus

As the GraphML file is ready, all the data then can be visualized. The TreePlus user control is implemented. Figure 4.4 shows the TreePlus code that provides the interactive graph visualization.

```csharp
private void InitializeControl()
{
    treePlusControl1 = new TreePlusControl.TreePlusControl();
    Rectangle desiredBounds = new Rectangle(0, 20, 1024, 768);
    treePlusControl1.Bounds = desiredBounds;
    treePlusControl1.AllowDrop = true;
    treePlusControl1.DragEnter +=
        new DragEventHandler(TreePlusControl_DragEnter);
    treePlusControl1.DragDrop +=
        new DragEventHandler(TreePlusControl_DragDrop);
    treePlusControl1.DragLeave +=
        new EventHandler(TreePlusControl_DragLeave);
    Controls.Add(treePlusControl1);
}

public void ShowGraph(string filename)
{
    graph = new GraphLibrary.Graph();
    graph.LoadGraph(filename);
    treePlusControl1.Visible = true;
    treePlusControl1.DataGraph = graph;
    pref = new Preferences();
    pref.MaxWidth = 1000;
    pref.FocusDesc = "Focus";
    pref.OutDesc = "Outgoing";
    pref.InDesc = "Ingoing";
    pref.BorderWidth = 3;
    pref.TickSize = 2;
    pref.LevelSeparation = 30;
    pref.OrderBy = "Link Direction";
    pref.SortOrder = "Ascending";
    pref.ShowLegend = false;
    pref.ShowTooltip = false;
    treePlusControl1.Preferences = pref;
    treePlusControl1.ShowGraph(rootNodeId);
}
```

Figure 4.4: TreePlus code for interactive graph visualization
4.2.4 ReVis Tool

ReVis applies several visualization techniques through the TreePlus component. Figure 4.5 demonstrates the visualization techniques applied in ReVis: Dynamic queries, Zooming, Node link diagram, TreeMaps, Focus + Context, Highlighting, Colour and Overview + Details.

Figure 4.5: ReVis Tool
The followings are the details of ReVis tool:

4.2.4.1 Organize

ReVis organizes the requirements into 4 layers. There are project, stakeholder, requirements of stakeholder and requirements flow. The following screenshots show the input process of ReVis:

Figure 4.6 shows the interface to input the first layer, Project. The interface is used to add new project, edit and delete the existing project. When the user clicks the Add button, pop up screen appears and ask user to enter project name and description.

![Figure 4.6: Input – Project](image)

Figure 4.7 shows the second entry. The user enters the stakeholders of the system created before. Add, edit, delete give same functionalities to add new stakeholder of the system or edit and delete the existing stakeholders.
Figure 4.7: Input – Stakeholders

Figure 4.8 shows the interface to input stakeholder requirements.

Figure 4.8: Input – Requirements
Figure 4.9 shows the last entry which is the requirements flow. Pop up screen will appear when the user click Add, Edit and Delete button. Requirements flow is used to enter the steps or processes of stakeholder requirements.

![Figure 4.9: Input – Requirements Flow](image)

4.2.4.2 Visualize

Once the input is ready, user can start visualizes the requirements. This can be done through Visualize menu. There are three submenus in it: Save and Visualize, Open and Visualize and Close Visualization.

Two submenus: Save and Visualize and Open and Visualize are used to visualization the requirements. Close Visualization is used to close the current visualization.
To start visualize, the user can click Save and Visualize menu. Then, pop up screen will appear to allow user to choose which project to save. After choosing the project, the user is given two selection buttons: Save and Save and Visualize. The function of Save button is to save the project with its requirements. While, the function of Save and Visualize button is to save the project with its requirements and then visualize it. Figure 4.10 shows the interface to save the project and all the requirements.

![Figure 4.10: Save And Visualize](image)

When the user clicks the Save or Save and Visualize button, Save Dialog Box appears. The user needs to enter file name and it will be save as GraphML file. Below in Figure 4.11 shows the interface to save the project into GraphML file.
The user can also click Open And Visualize to visualize the requirements. But this menu is only used to visualize the requirements that already been save as GraphML file. Figure 4.12 shows the Open And Visualize Dialog Box.
Once the project is saved or opened, the visualization of the requirements starts.

Figure 4.13 shows the first visualization screen for Online Assessment System. Two levels displays: Level 1: Project name and Level 2: Stakeholder.

Next, the user can start navigating the tree by double clicking on the nodes. For example, when the user double-clicks on the Lecturer node, list of requirements for lecturer expanded. Figure 4.14 shows list of Lecturer’s requirements.
To bring focus to the node, the user can single clicking the node and it previews all currently visible and not currently visible adjacent nodes to the node on the Preview Panel. Refer Figure 4.15.
Figure 4.15: By single clicking Display Statistic of analysis, the node got focus (indicated by green background and thick border) and currently visible nodes and not currently visible nodes previewed on the Preview Panel on the right.

When the user double clicks the “Display statistic of analysis” node, all currently not visible adjacent nodes will move from the Preview Panel to their position in the tree which is “Display statistic of analysis” node. The Preview Panel is then refreshed to reflect that all nodes are now visible. Refer Figure 4.16.
Figure 4.16: The refresh of Preview Panel once the user double clicked “Display statistic of analysis” node.

Figure 4.17 shows the whole visualization of requirements for Online Assessment System.
Figure 4.17: Visualization of requirements for Online Assessment System

Noted that in ReVis, the requirements are visualized as a tree-style layout. Each requirement is represented as a node. The nodes are classified into 4 levels. There are:

A. Level 1: Project name
B. Level 2: Stakeholder
C. Level 3: Requirements
D. Level 4: Requirements Flow

Besides the nodes, Preview Panel is provided on the right to preview all adjacent nodes connected to the node (Refer Figure 4.17 E). Other activities such as zooming and panning, and searching are provided on top (Refer Figure 4.17 F).
Figure 4.18 shows the zooming activity. The user can zoom the tree by using the slider.

There is a Fit In button that allows the user to fit the visualization in the current screen size. Besides that, the visualization itself can be moved when the user holds and moves the mouse. Figure 4.19 shows the Fit In button.
Next, the tool also provides searching function. User can search for any keyword in the system by entering the keyword and press Search button. The tool will display the result as shown in Figure 4.20.

![Search function](image)

**Figure 4.20 : Search function**

Next in Figure 4.21 shows the function of Home button. User clicks Home button to return to the first visualization screen.
Figure 4.21: Home button

Finally, to close the visualization screen, ReVis provides Close Visualization button. It is shown in Figure 4.22.

Figure 4.22: Close Visualization button
4.2.4.3 Verify

User can also use the tool to verify the requirements with the stakeholders. Any invalid requirements can be edited or deleted by selecting Verify menu as shown in Figure 4.23. The process will iterate until the requirements are agreed by all stakeholders.

Figure 4.23: Verify - Edit and Delete button
CHAPTER 5 : EVALUATION

This chapter concentrates on result and analysis that has been made upon evaluation of ReVis tool. Three sequences of processes involved in the evaluation: a questionnaire, system testing and result analysis.

Specifically, ReVis Tool is developed to fulfill the first research objective and an experiment to evaluate the tool is then designed to further answer the next research objectives: To evaluate the usability, functionality and effectiveness of the tool.

5.1 Preparing questionnaire

Steps taken to prepare the questionnaire were:

i. Write a draft questionnaire

ii. Pilot the draft

iii. Refine the draft into final version.

The draft questionnaire for this tool was designed based on research objectives and it was divided into a few issues. The first issue was on familiarization with information visualization techniques and any visualization tools. The second issue was the usability issue of the tool to test on the visualization techniques applied and the easiness of the tool. The functionality issue was designed to evaluate the overall functionality of the tool. Next, the effectiveness issue was also designed to evaluate on how the tool helps in organizing, verifying and tracking the requirements. Rate of visualization technique applied was also one of the issues. The answers of each question in the draft were in 6 Likert Scale format.
Once completed, the draft was piloted by 3 lecturers from Software Engineering Section, Universiti Kuala Lumpur and 2 programmers from IT Section, Universiti Kuala Lumpur. They were chosen because of their experience and knowledge on requirements engineering and system development. The purposes were to record which questions they struggled with, to check whether the questions were interpreted as intended and to record the time taken to complete the questionnaire. As a result, all of them agreed with the draft and suggested the duration of the testing is within 30 – 45 minutes. Finally the draft was refined into final version. This is the version or the questionnaire that was distributed to the participants in system testing (Refer Appendix C).

5.2 System Evaluation

5.2.1 Participants

20 participants were recruited. They were 10 requirements engineers from IT Section, GIATMARA Sendirian Berhad and 10 lecturers from Software Engineering Section, Universiti Kuala Lumpur. The evaluation was conducted at Lab 1905, Universiti Kuala Lumpur with 5 participants for each session. Four sessions were scheduled.

5.2.2 Procedure

The evaluation session was administered by the author. Participants first listened to the demonstration on ReVis tool given by the author. The demonstration started with the brief explanation on the research and the tool developed. Then the participants were shown by the author on how to use the tool. One proposed system, Online Assessment System was being used
in the demonstration and they were allowed to ask any questions. After the demonstration the participants evaluated the tool. During the evaluation, the participants were given a tutorial that can help them to navigate the visualization in the tool (Refer Appendix D). Once participants completed the evaluation, they answered the questionnaire. Each session lasted up to 45 minutes.

5.3 Result, Analysis and Findings

From the survey, 100% of participants were familiar with the information visualization techniques and also familiar with at least one visualization tool.

The following discussions are the major discussion that will answer the research objectives.
### 5.3.1 Usability Issue

<table>
<thead>
<tr>
<th>Usability Element</th>
<th>Average Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Over all the system was easy to use</td>
<td>5.3</td>
</tr>
<tr>
<td>b. I felt comfortable using the system</td>
<td>5.05</td>
</tr>
<tr>
<td>c. The information was layered effectively</td>
<td>5.2</td>
</tr>
<tr>
<td>d. Information was easy to read</td>
<td>5.3</td>
</tr>
<tr>
<td>e. The colours used were appropriate</td>
<td>4.9</td>
</tr>
<tr>
<td>f. Highlighting on screen was helpful</td>
<td>4.85</td>
</tr>
<tr>
<td>g. It easy to find the information needed</td>
<td>5.1</td>
</tr>
<tr>
<td>h. Arrows representing the information flow were helpful</td>
<td>5.25</td>
</tr>
<tr>
<td>i. It is clear how screen elements (eg: zoom, search, etc.) work</td>
<td>5.2</td>
</tr>
</tbody>
</table>

Table 5.1: Average Likert Scale Rating of Usabilities Issue in ReVis Tool, using the scale of 1 = Strongly Disagree, 6 = Strongly Agree
Graph 5.1: ReVis Tool: Usability Issue

<table>
<thead>
<tr>
<th>Element</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
<th>35</th>
<th>40</th>
<th>45</th>
<th>50</th>
<th>55</th>
<th>60</th>
<th>65</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>System was easy to use</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>I felt comfortable using the system</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>The information was layered effectively</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Information was easy to read</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>The colour used was appropriate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Highlighting on screen was helpful</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Information found was easy to find</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Arrows representing the information flow</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Other than colour were appropriate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

ReVis Tool: Usability Issue

- Over all the system was easy to use
- I felt comfortable using the system
- The information was layered effectively
- Information was easy to read
- The colour used was appropriate
- Highlighting on screen was helpful
- Information found was easy to find
- Arrows representing the information flow
- Other than colour were appropriate
Table 5.1 shows the average ratings of usability elements by 20 participants. The ratings were placed in 1-6 Likert Scale. The average of the Usability Issue is measured as:

Total average rating / 9 elements = 5.13 (5 = Agree)

The detailed result of each usability element is shown in Graph 5.1. 75% of participants agreed that the system was easy to use and arrows representing the information flow were helpful. 40% of participants were strongly agreed on how the screen elements work. The graph also shows that 25% of participants strongly agreed with the highlighting on screen but there was still 40% of participants who tend to agree with the issue. Meanwhile, 5% of participants tend to disagree with the colour used in the tool.

Therefore, from the result above, it can be concluded that the tool is usable and accepted.
### 5.3.2 Functionality Issue

<table>
<thead>
<tr>
<th>Functionality Element</th>
<th>Average Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. All features used for entering requirements (eg: button, menu etc.) are working correctly</td>
<td>5.25</td>
</tr>
<tr>
<td>b. All features used for visualizing requirements (zooming, searching, panning etc) are working correctly</td>
<td>5.4</td>
</tr>
<tr>
<td>c. All data are saved and retrieved accordingly</td>
<td>5.2</td>
</tr>
<tr>
<td>d. Smooth navigation</td>
<td>5.25</td>
</tr>
</tbody>
</table>

Table 5.2: Average Likert Scale Rating for Functionality Issue in ReVis Tool, using the scale of 1 = Strongly Disagree, 6 = Strongly Agree
Graph 5.2: ReVis Tool: Functionality Issue

- All features used for entering requirements (e.g., buttons, menus etc.) are working correctly.
- All features used for visualizing requirements (e.g., zooming, searching, panning etc.) are working correctly.
- All data are saved and retrieved according to their requirements.
- Smooth navigation.

Percentage of Participants

<table>
<thead>
<tr>
<th>Functionality Issue</th>
<th>All features for entering requirements</th>
<th>All features for visualizing requirements</th>
<th>All data are saved and retrieved correctly</th>
<th>Smooth navigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disagree Strongly</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Disagree</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tend to Disagree</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tend to Agree</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Agree</td>
<td>45</td>
<td>60</td>
<td>60</td>
<td>45</td>
</tr>
<tr>
<td>Agree Strongly</td>
<td>40</td>
<td>40</td>
<td>30</td>
<td>40</td>
</tr>
</tbody>
</table>
Table 5.2 shows that the average rating for each element is above 5.1.

The average of the Functionality Issue is measured as:

Total average rating / 4 elements = **5.28** (5 = Agree)

Meanwhile, Graph 5.2 shows that 60% of participants agreed and 40% strongly agreed on all features used in visualizing the requirements.

Therefore, it can be concluded that the features, the data retrieval and navigation in the tool are well functioned.

### 5.3.3 Effectiveness Issue

<table>
<thead>
<tr>
<th>Effectiveness Issue</th>
<th>Average Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. The system helps organize the requirements elicited from the stakeholders</td>
<td>5</td>
</tr>
<tr>
<td>b. The system helps in the verification of the stakeholder needs</td>
<td>4.9</td>
</tr>
<tr>
<td>c. The system helps in tracking the requirements source</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Table 5.3: Average Likert Scale Rating for Effectiveness Issue in ReVis Tool, using the scale of 1 = Strongly Disagree, 6 = Strongly Agree
<table>
<thead>
<tr>
<th>Element</th>
<th>Disagree Strongly</th>
<th>Disagree</th>
<th>Tend to Disagree</th>
<th>Tend to Agree</th>
<th>Agree</th>
<th>Agree Strongly</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. The tool helps organize the requirements elicited from the stakeholders</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>60</td>
<td>20</td>
</tr>
<tr>
<td>b. The tool helps in the verification of the stakeholder needs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>60</td>
<td>15</td>
</tr>
<tr>
<td>c. The tool helps in tracking the requirements source</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Graph 5.3: ReVis Tool: Effectiveness Issue
The average rating values for each element in Effectiveness Issue showed a small difference from one element to another as shown in Table 5.3. The average of the Effectiveness Issue is measured as:

\[
\text{Total average rating} / 3 \text{ elements} = 4.9 \ (4 = \text{Tend To Agree})
\]

Meanwhile, from Graph 5.3, it shows that 100% of participants agreed that the tool helps organize the requirements. The same percentage of participants also agreed that the tool helps in verifying the stakeholder needs. Meanwhile, only 5% of participants tend to disagree that the tool helps in tracking the requirements source.

Therefore, it can be concluded that the tool is effective in organizing, verifying and tracking the requirements.

### 5.3.4 Visualization Technique Rate

<table>
<thead>
<tr>
<th>Visualization Technique Rate</th>
<th>Average Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>How would you rate the visualization technique used in this tool?</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.4: Average Likert Scale Rating for Visualization Technique Rate in ReVis Tool, using the scale of 1 = Disagree Strongly, 6 = Agree Strongly
Graph 5.4 : ReVis Tool : Visualization Technique Rate

Table 5.4 shows the average rating for visualization technique rate where all ratings were placed in 1-5 Likert Scale (1 = Extremely Poor, 2 = Below Average, 3 = Average, 4 = Above Average, 5 = Excellent). The average rating of visualization techniques used in this tool is 5. The participants felt that the visualization techniques used were excellent.

It is also shown in Graph 5.4 whereby the percentage of participants who strongly agree and agree was not much difference and none of the participants felt the techniques used were poor.
5.3.5 Overall Result

In Graph 5.5 above, the data collected was divided into two groups: Agree and Disagree. The Likert Scale points that fall under Agree were point 4, 5 and 6. The Likert Scale points that fall under Disagree were point 1, 2 and 3. From the graph, it shows that 100% of participants agreed on functionality issue. 99% agreed on usability issue and 98% on effectiveness issue.

The following are the findings that are based on the result found in this project:

a. All participants were agreed with all usability and functionality elements evaluated. Only minor improvement need to be done on usability issue which is on the colour used and highlighting on screen.

b. All participants were agreed with all elements evaluated on effectiveness issue and visualization technique. They felt that the tool helps to organize the requirements elicited. They also felt the visualization techniques applied was excellent to represent the requirements. Furthermore, the participants also agreed that the tool
helps in verifying the stakeholder needs and tracking the requirements.

As conclusion, ReVis Tool achieves all research objectives and it seems to be a requirement visualization tools that can help the requirement engineers to organize, visualize and verify the requirements.
CHAPTER 6: CONCLUSION

This last chapter concludes the project on A Visualization Tool for Requirements Specification. This chapter also presents the discussion on project limitation and recommendation for future work.

6.1 Conclusion

There are many problems arose in the requirement specification stage. Among the problems are lack of quality requirement specification and a very detail of requirement document that brought in different interpretation between user and developer. All these problems will lead to incorrect requirement specification and hence affect the next stage in Software Development Life Cycle. Due to that, it is important to describe the requirements in a form that the stakeholders can clearly understand and then can easily verify them. Visualization techniques appear as a good solution to help in these processes.

ReVis Tool is developed to serve the needs mentioned above. The tool aims on requirement elicitation stage since there are only few available requirement visualization tools being used during elicitation stage. This requirement visualization tool is built to help the requirement engineers organize, visualize and verify the requirements gathered during the stage.
In ReVis Tool, the requirements are organized based on stakeholders. Then, the requirements are visualized using Node Link Diagram, Treemaps, Dynamic and Queries, Overview + Detail, Focus + Context, Highlighting, Zooming and Colouring techniques. Here, the techniques were applied using TreePlus user control. The requirements verification can proceed once visualized.

The tool has been evaluated based on its usability, functionality, effectiveness and visualization technique applied. The percentage of participants agreed on its usability, functionality and effectiveness were 98%, 100% and 99% accordingly. The visualization techniques were rated as excellent. This shows that the tool is suitable to be used as requirement visualization tool. Therefore, the objectives of the research achieved.

### 6.2 Future Work

There are a number of enhancements that if pursued, might improve the effectiveness of ReVis Tool.

a. Enhance the searching feature. The keywords found should be highlighted instead of showing the tracking.

b. Add more advance functions. The changes of the requirements can be done directly on the visualization screen and automatically save on database. Filtering on the tasks that not yet completed can also be added. The completed and verified requirements can be converted into other documents.

c. The visualization of the requirements should be displayed directly without need to be saved first on other file format.
d. Deal with requirement conflict.

e. Enhance the tool to be online requirement visualization tool so that the verification of the requirements can be done remotely by all stakeholders.
REFERENCES


Bongshin Lee (2006) Interactive Visualizations for Trees and Graphs, Ph.D, University Of Maryland

Borland®Caliber®Analyst,


Cradle, http://www.threesl.com [viewed on October 2006]

Customer System Usability Questionnaire,
http://oldwww.acm.org/perlman/question.cgi [viewed on February 2008]


Ware, C. (2000) Information Visualization: Perception For Design. San Francisco: Morgan Kaufmann
APPENDIX B : TREEPLUS API