QUIM Editor: a Tool for Quality in Use Measurement

Jovan Strika

A Thesis

In

The Department

Of

Computer Science

Presented in partial fulfillment of the requirements
For the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

March 2003

0-612-77720-0

ABSTRACT

QUIM Editor: a Tool for Quality in Use Measurement

Jovan Strika

In this thesis is described the Quality in Use Integrated Map (QUIM) editor that aims to allow developers and managers to visually interact with the QUIM framework and in particular, the QUIM Map. QUIM (Quality in Use Integrated Map) framework is proposed by the human-centered software engineering group as an integrative framework for understanding and applying usability models consistently. At its current state of development, the map contains 10 quality in use factors, 27 criteria and more than 100 metrics. It also describes the relationship that exists between these factors, criteria and metrics.

The QUIM editor is an application written in Visual Basic, which interacts with a Microsoft Access database. Beside browsing and searching, QUIM editor also helps developers to customize the QUIM Map by extracting factors, criteria, and metric that are relevant to a specific project or application such as building a customized usability model for measuring trustfulness and satisfaction in Web applications. This customized model can then be used as "usability testing guidelines". QUIM editor is useful for anyone who wants to learn how usability can be measured, how it can specify or predict quality, as well as how usability should be integrated in the development lifecycle.

ACKNOWLEDGMENTS

Hereby, I would like to thank all persons who helped and encouraged me in writing this thesis.

I would also like to thank all my friends and family who gave valuable comments in improving the thesis. To my supervisor, Dr. Ahmed Seffah, I would like to express my gratitude for his support, patience, understanding, and constructively critical eye. And finally to my wife, Nela, who suffered through each paragraph along with me, I acknowledge a debt, an appreciation, that extends beyond any words at my command.

Table of Contents

List of Figures

1 Introduction

Nowadays, many products and systems are competing for the same market share. A software product is expected to be used for accomplishing certain tasks with efficiency, while increasing consumer performance, productivity and satisfaction. Presumably, all developed products are supposed to have a high degree of usability, but in fact they do not. The usability is generally implicit and taken for granted [1]. Usability is important for professionals that use interactive systems as tools in their daily work. The products and systems should be more than user friendly and should provide the functionality for which they are designed.

1.1 Usability as a Measurable Quality Factor

To be more explicit about how to achieve a usable product or system, it is necessary to clarify its meaning, i.e. define quality in use and usability. There are a number of definitions of usability that differ to a certain extent. In the discussion below, I have chosen the ISO definition(s) of usability. ISO is an international standard and widely used as opposed to other definitions of usability or related concepts. By further exploring the ISO standards, I have found that there are at least two different definitions of usability as well as a similar concept, quality in use.

During the 90's, based on the information presented in annex D of the ISO 9241-11 [2], it was concluded that the term usability refers to the capability of a product to be used easily. The ISO/IEC 9126 [3] view was derived from the ISO 8402 [4] which recognized and defined usability and part of product/system Quality through definition of quality as:

> *"The totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs"*

The ISO/IEC 9126 definitions acknowledged that the objective was to meet user needs. The expectations for quality were increased in consumer and professional markets. ISO/IEC 9126 tried to broaden the perception of quality by categorizing quality from a user perspective as functionality, reliability, usability, efficiency, maintainability and portability (Figure 1).

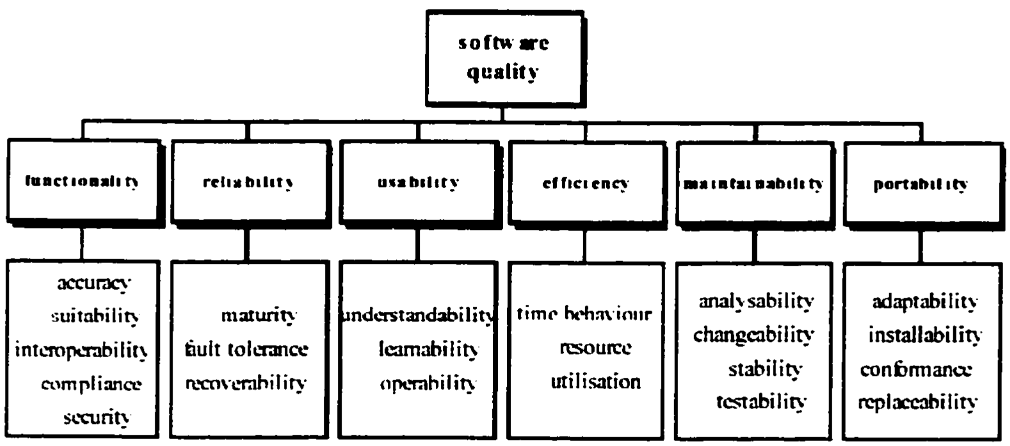

Figure 1 – ISO 9126 (adapted from [3])

As a software quality attribute, usability is defined:

"A set of attributes of software that bear on the effort needed for use and on

the individual assessment of such use by a stated or implied set of users" [3]

The ISO/IEC 9126-1 in its 1999 revision defines usability as:

"The capability of the software to be understood, learned, used and liked by the user when used under specified conditions"

ISO/IEC 9126-1 distinguishes between two components in the software quality model: 1) external and internal quality, and 2) quality in use. It considers usability as a characteristic of external and internal quality that can be measured using external and internal attributes of the product that contribute to software product quality in use.

The ISO 9241-11 standard defines usability as:

"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [2]

In this standard, usability is influenced not only by the ease of software use, but also by the functionality, reliability and efficiency, as well as the suitability of the hardware, user and task.

This diversity of definitions and perceptions make usability a very confusing concept. However, the different standards and definitions are intended for different target groups, which may explain the discrepancies.

Managers believe that a highly usable product should meet business needs in terms of human productivity and consumer satisfaction. Developers think that a usable product should offer a wide variety of functionalities and provide a task-oriented help system. But for an end-user, a good product is the one that is easy to use and offers a useful functionality when it is needed and without any training. Experienced users are interested in completing a wider range and number of tasks with minimal obstruction. Therefore, a usable system would be easy to use over time with experience, but not necessarily easy to learn [2].

Different audiences perceive usability differently, as evidenced above, which is why the definition of usability evolved and is still an unclear concept for the majority of developers and users. Usability engineering and user centered design help designers to determine potential users of their product, and furthermore, to ensure their product meets the needs of the end-users. In parallel with a development cycle, usability engineers need to establish a usability cycle that contains a constant measurement of usability with respect to users.

There are conflicting views on how to effectively measure usability. ISO definitions of usability do not provide much help in this quest, even though they normally break down the concept into measurable elements, which include effectiveness, efficiency and satisfaction in use [5]. These are, in fact, very different concepts, and their grouping may create more difficulties than solutions for assessment.

1.2 QUIM Framework

Donyaee [6] established the foundation of Quality in Use Integrated Measurement (QUIM) framework. QUIM is based on the decomposition of the quality in use into factors attributes. Software consumers often express their needs in general qualitative terms, such as reliability and efficiency. It is, thus, necessary to decompose consumer-oriented attributes into technically-oriented attributes that are more meaningful to software producers. QUIM is a layered model for usability measurement with four layers referred to as factors, criteria, metrics and data (Figure 2). The relationship among these layers is a many-to-many relationship. "Quality in Use", the end-user perception of software quality, is at the top of the hierarchy. The end-user is mainly interested in software performance and the effects of its use, and typically is not concerned about the internal aspects of the product. The end-user just perceives observable, external attributes of the software. The quality in use factor attribute is a user-oriented characteristic of the user interface. It defines the quality of user interface in the user's language and, thus, is easily understandable to the user. However, it is not easy to specify and measure the usability factors. The factors could be divided into the sub-factors or criteria attributes. Criteria are more technical sub-factors or sub-characteristics of the user interface. Criteria define the quality in use in the language of user interface developer, and, thus, are less understandable to the end-user. Criteria are determined by one or more observable attributes called metrics. The metric can have numeric value based on interpretation and calculation of data attributes.

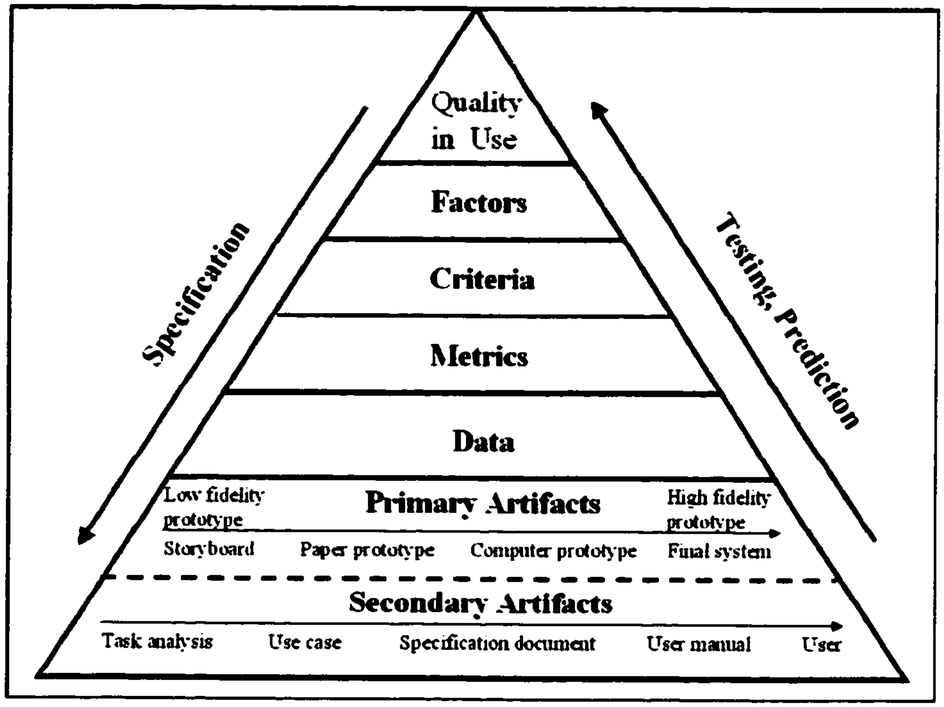

Figure 2 – Quality in Use Integrated Model [6]

Since QUIM is designed to be used over the entire software development process, inquiry and collection of data attributes may vary with product stage, expressed in QUIM framework as the Primary Artifact. For example, interviewing users based on "Low fidelity prototype" will most probably produce less compelling results than an interview based on "Final system". Different usability tools and collection methods, such as the "Task analyses" or "Questioners" are expressed as the "Secondary Artifacts".

To obtain the usability attribute values as numbers, the framework must be empirical; that is, measurement and metrics speak louder than words [6]. For instance, the fact that 40% of customers are able to find and order a product after testing an e-commerce web site, and analysis of such a measurement (if 40% is good or bad), carries much more weight than any speculation about improved web navigation systems. It is also useful to track the results of metric-oriented tests over time. This will inform developers on their progress towards the design improvement. Kitchenham notes that "quality is hard to define, impossible to measure, easy to recognize", but it is postulated that it is possible to measure with framework that can provide coherent and consistent measures over time that are understandable to both experts and non-experts in the area of usability [7].

1.3 Research Objectives and Methodology

To be useful, QUIM should be supported by an automated set of tools that improves the process of specifying and measuring usability during the whole software development process. As a first step toward an integrative and comprehensive quality in use measurement toolbox, the QUIM editor defined and developed in this thesis aims to:

- Help to build and maintain a measurement map with a set of factors, criteria, metrics and data. Today, several models for usability and quality in use measurement are available both in the software and Human Computer Interaction (HCI) communities [3]. However, in spite of their possible usefulness, they are not necessarily easy to understand, master and apply because of ambiguity and contradictions in their definitions [8];
- Help developers and quality insurance managers to develop specific usability models by customizing and/or extending existing models;

- Contribute to the establishment of an empirical database for analyzing and interpreting metrics. This database is essential for the development of new usability models, in particular predictive models;
- Facilitate the integration of usability and its measurement in the software development lifecycle, particularly in the software quality insurance models and the testing process;
- Produce a testing plan and benchmark reports that can be developed during the usability requirement process and used during the usability testing process.

Another important requirement for the QUIM editor is to support the following five categories of users (Figure 3):

- Consumers, who may be interested to get a better understanding of what can influence the usability of a software system;
- Developers, who may need more technical information on usability metrics and relationships among them;
- Usability managers, who may be able to define new usability factors, criteria and metrics, measurement methods and models;
- Development managers, who need to create specific usability models including relevant attributes and relationships among them;
- Testers, who need support, and guidelines for collections and interpretation of the

data and the metrics, during usability measurement.

Therefore, there are three ways to understand and use QUIM editor as a software measurement tool:

1. **Exploring and Understanding Existing Usability Models**. Most of the existing qualities in use models are described in ways that are difficult to understand and learn. QUIM editor offers different interactive features for exploring a model, such as definitions, goal-driven browsing and keyword searching. It also provides several features for visual and graphical exploration of the relationships among sets of factors, criteria and metrics.

2. **Establishing, Managing and Formalizing the Practices and Knowledge related to Quality in Use Measurement**. QUIM editor may help quality assurance managers, to create a repository of factors, criteria and metrics, as well as to define the relationships among them that can emerge in day-to-day practice or from existing usability models. This repository is referred to as a 'knowledge map' in this thesis, as it not only includes the factors and their relationships, but also the practices, such as how to measure and how to interpret measures. QUIM editor offers features for checking and validating the integrity of the quality in use measurement map (preventing duplicate and unrelated attributes).

3. **Building a new Quality in Use Model**. QUIM editor provides a number of features for creating and customizing a specific model from the measurement knowledge. Among others, it includes a wizard, which may help developers and

testers unfamiliar with usability measurements, to create a step-by-step specific measurement model, such as a model for measuring safety in critical systems or Web sites accessibility.

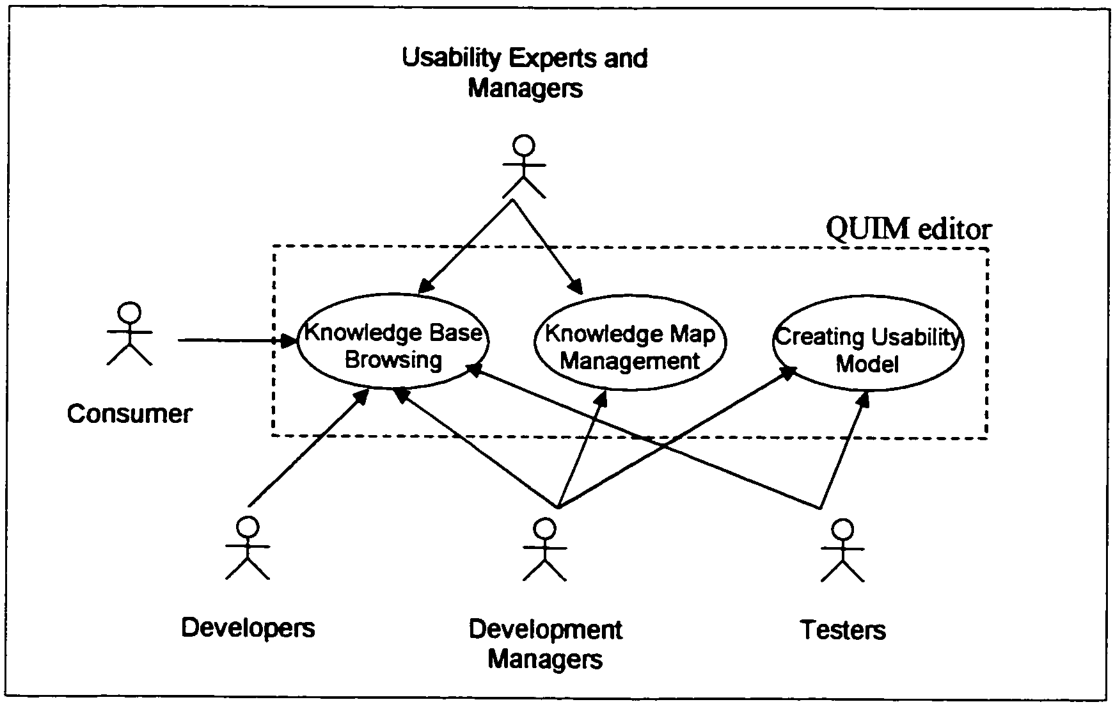

Figure 3 – QUIM editor use cases

This thesis presents the features and architecture of QUIM editor. The QUIM knowledge Map structure and content are presented in Padda Harkikat's thesis (manuscript in preparation).

1.4 Thesis Organization

The basic concepts and features provided by the QUIM editor are described in Chapter 2.

An illustrative example is used to demonstrate the facilitated exploration and understanding of an existing usability model by using the QUIM editor.

Chapter 3 describes the management of the measurement knowledge map as an iterative process of creating and refining the knowledge map. The rationale for QUIM knowledge map management is based on data mining, measurement process and knowledge management research.

Creation, adaptation and refining of a usability model are discussed in Chapter 4, using an illustrative example. It explains how QUIM editor may help in decision making, in particular, choosing relevant usability attributes, defining priorities, and generating a model graph, reports and testing form to be used in the measurement process. In addition, advantages of the QUIM over existing models like MUSiC [36] and SUMI (Software Usability Measurement Inventory) [30] measurement frameworks, are summarized.

Chapter 5 summarizes the research and development issues related to the QUIM editor such as comparison with existing software tools and milestones in development processes.

The lessons learned and the future avenues toward an integrative framework for knowledge-based measurement are described in the conclusions.

1.5 Basic Terminology

Knowledge map represents a collection of QUIM elements and their relationships that emerged from empirical studies. A usability attribute that exists in a form of QUIM knowledge map element like factor, criteria, metric or data, is referred to as "attribute" in this thesis. This collection of usability attributes can outline one or many usability models or standards. For description of attributes used in this thesis please refer to the Appendix. In the QUIM terminology, a 'model' refers to the existing or custom usability model, as a sub-view of the knowledge map.

2 QUIM Editor Basic Concepts and Features

Software usability has recently been the subject of many international standards, directives and theoretical and empirical research, like the ISO Standard 9126 [3] that relates to software quality, ISO 9241 [2] that concerns ergonomic requirements for the use of computer equipment, and the European Council directive 90/270/EEC [9] on minimum safety and health requirements for work with computer equipment. At the same time, many practical techniques for measuring usability have been proposed for the interactive software development lifecycle [10], [11], [12]. Poorly indexed and cross-referenced documents exist worldwide. Standards development equally depends on individuals involved in research and on technical issues. Experts outside the standardization process have strong and valuable views, but fail to contribute because the process of standardization is confusing and bureaucratic. It is unclear how all the existing lists, rules and criteria are related (if at all) and whether one list may be more useful than others. To understand the various checklists and relationships among them, the concept of usability needs to be broken down in a way that allows comparisons from both theoretical and practical viewpoints.

QUIM is designed to primarily address these issues by segregating usability standards, measurable metrics and methods from numerous sources in one centralized knowledge base. Each usability model or standard can be presented as a QUIM measurement knowledge map for interactive browsing of all elements and their definitions, methods, rules of inspection and interpretation. Basic function of the QUIM editor is to make that knowledge easily accessible for any type of user.

2.1 Exploring and Learning Existing Quality in Use Models

Abstract definition of efficiency, effectiveness and satisfaction is given in the ISO 9241-11 [2] standard. "Efficiency" is defined as the resources expended in relation to the accuracy and completeness with which users achieve goals. "Effectiveness" is defined as the accuracy and completeness with which users achieve specified tasks. "Satisfaction" is a subjective measure and concerns the comfort and acceptability of use by end users. This definition approaches usability from a theoretical viewpoint and may not be very practical. Nielsen [10] has a slightly different definition that is classified in elements that are more specific. Nielsen only regards the expert users when talking about efficiency, although learnability is also directly related to efficiency. Memorability mainly relates to casual users, and errors deal with those errors not covered by efficiency, which have more catastrophic results [13]. Shneiderman offers a nearly identical definition that differs only in terminology [14]. Shneiderman does not refer to his definition as a definition of usability, but as "five measurable human factors, central to evaluation of human factors goals" [14]. Many inconsistencies are identified in the existing quality in use and usability models [15]. In addition, relationships among factors, criteria and metrics have not been clearly defined. Enormous research is required from any user willing to explore and learn about these models.

In QUIM editor usability models are accessible to user. Browsing through selected usability models in a form of a knowledge map enables users to learn how to choose and apply appropriate cost-effective methods and tools in commercial projects at different stages of development. Since all models are based on the QUIM framework structure, users can easily compare how each model relates to their business goals (Figure 4).

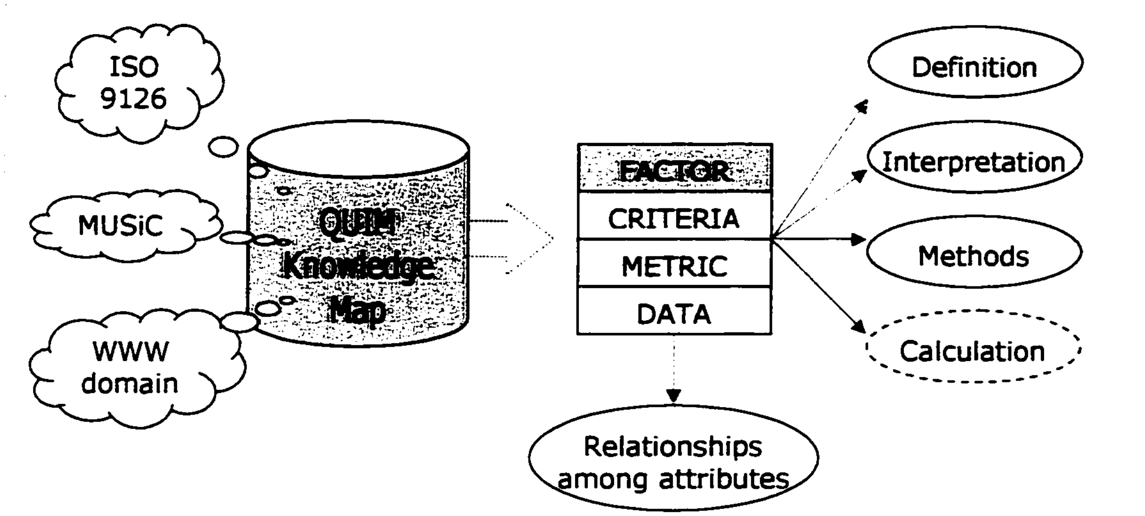

Figure 4 – QUIM Knowledge Base Browsing

During the design of the QUIM editor user interface, we took into consideration that browsing of QUIM database should be primarily used by product end-users (customers) or developers that are not familiar with the usability, and also by usability experts and development managers during knowledge map management and model design. We try to achieve simplicity and, as much as possible, user interface that is easy to master. The QUIM editor main application window that emerged from numerous low and high fidelity prototypes, is presented in Figure 5.

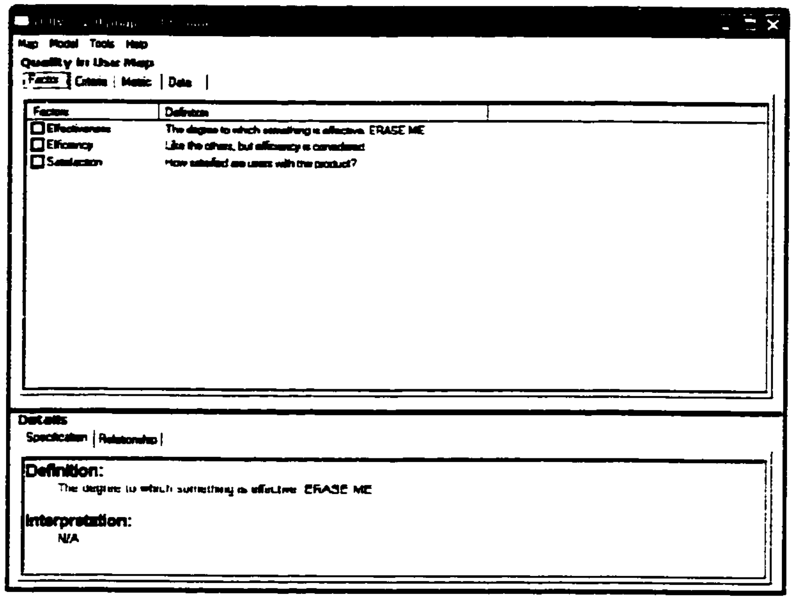

Figure 5 – QUIM editor interface

The upper frame of the QUIM editor displays four layers of QUIM knowledge map used to decompose the usability attributes: factors, criteria, metric and data. Since the QUIM editor automatically loads the most recently used usability knowledge map, user can start exploring usability attributes by selecting Quality in Use Map layers "tabs". Each layer will list all usability attributes previously established by quality insurance manager. Each attribute has a "Name", small description called "Definition" and, in front of the "Name", a check box used during creation of custom usability model process (see below). Details of an attribute, selected by user as an attribute name, are displayed in the lower frame. Specification and relationship of the selected attribute are presented in the details frame. Characteristics of usability attributes are defined in its specification that contains the following information:

Attribute name: This is unique name of factor, criteria, metric or data that is defined by standard or can be established by usability expert within his proposed model;

Definition: Since name can be understood in many different ways, definition offers detail description of attributes, context of use and methods for collection. As presented earlier, different models can have different definition for same or similar attributes;

Interpretation: This information is crucial for the measurement of usability. It explains how to calculate it, how to interpret it, and how it affects the product's quality. Software attributes related to human-centered design fall into two categories: process-oriented and product-oriented. Process-oriented attributes, which specify procedures and processes to be followed, should be evaluated by product users. Product-oriented attributes that specify required attributes of the user interface, can be evaluated by developers and/or usability experts;

Calculation: In case of data and metric, which may have numerical values as a result of usability measurement, this should provide a formula used to calculate the final result. Variables used in the formula could be a result of other usability attributes. For example, values of three data attributes may be calculated to obtain the final result for one metric attribute. Evaluation of usability can be done afterwards and during design. The usefulness of all the guidelines, heuristics and other aids is related to the kind of conducted evaluation.

In the example presented in Figure 5, after selecting factor "Effectiveness" only the definition of "Effectiveness" will be displayed in the Details frame, under the tab "Specification", since the attribute interpretation is not available. If an attribute specification exists, it is displayed in a scrollable text box, from which users can "Copy-Paste" content into other applications using the standard Windows shortcuts "Ctrl+C" and "Ctrl+P".

Since the development of the QUIM framework is in continuous progress, screenshots and examples of knowledge base attributes and their specification used in this thesis might differ from the latest installation version of the QUIM editor.

2.2 Understanding and Visualizing Complex Usability Models

During research, users have a tendency to visualize relationships of decomposed usability attributes through a hierarchical "tree" structure, often presented in a form of document tables. As presented earlier, QUIM is a model with four layers of attributes called factors, criteria, metric and data (Figure 2). The relationship between these layers is an N-M (many-to-many) relationship. The following scenarios may exist for each layer:

- Factor attributes are subdivided into sub-factors called criteria. Therefore there is only a relationship with lower layer. One factor attribute can be related to several criteria attributes;

- Criteria attribute related to one factor can also be related to other factors, but in order to interpret it, several metrics may be necessary. Thus, each criteria can have multiple relationships with upper layer (factors) and multiple relationships with lower layer (metric);

- Metric attribute, similar to criteria, can have multiple relationships with upper layer (criteria) and lower layer (Data). If the metric attribute is not calculable using a formula, its value may be obtained from the user directly;
- Data attributes have only relationships with metrics. Some data attributes, used as variables in a usability measurement calculation, can affect more than one metrics attributes.

Due to the complexity of relations, there is a necessity for establishment of a presentation method that would clarify related attributes of each attribute within the model, as well as the complete model map structure. With the QUIM editor, users can explore attributes relationships using metric and graph approach.

2.2.1 Dependencies Matrix

QUIM editor introduces a matrix based classification scheme to assess relationships of individual usability attributes within a quality in use model. It allows a user to predict how the different values for data used to determine a metric or a combination of metrics could affect system's quality in use. The tab "Relationship" within the "Details" frame displays all the related attributes to the selected one. When a user selects an attribute from "Quality in Use Map", the left matrix box displays all attributes affected by selected attribute. The right matrix box displays attributes that can affect selected attribute or that are used as measurement variables to determine a value of selected attribute.

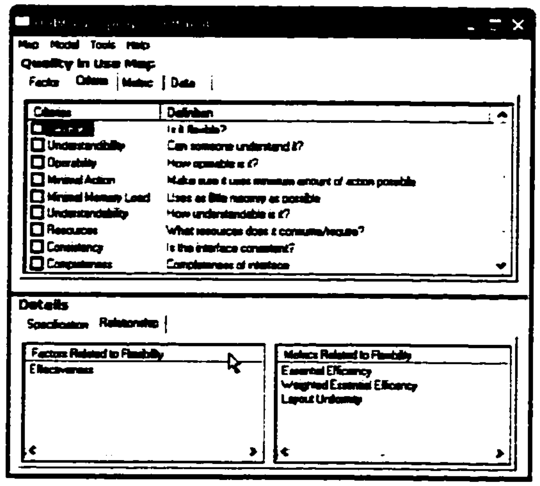

Figure 6 – Attribute relationships

For example, Figure 6 represents a case where the criteria attribute "Flexibility" is selected. Under the relationship tab in the box on the left side, where factors related to the "Flexibility" are displayed, the "Effectiveness" is displayed, indicating that only the factor "Effectiveness" will be affected by the criteria "Flexibility". On the right side, where metrics that affected "Flexibility" are displayed, the "Essential Efficiency", "Weighted Essential Efficiency" and "Layout Uniformity" are displayed.

The matrix approach is ideal for close-up attributes investigation. It can also be used to explore a model, since each attribute listed in the matrix is interactive. By double-clicking any related attribute, the QUIM browser will automatically jump to a proper attribute layer, auto select that attribute in the Quality in Use map and generate its related attributes in the matrix boxes.

2.2.2 Model Graph

Map graph is another way to visualize relationships among attributes within different models. This method is used to remove inconsistencies as well as to clarify definition of the relationships. It is an approach that can display a whole model map or just a map focused on one individual attribute. There is a new model wizard in the QUIM editor that allows a user to extract one or more attributes from the knowledge map and generate interactive model graph.

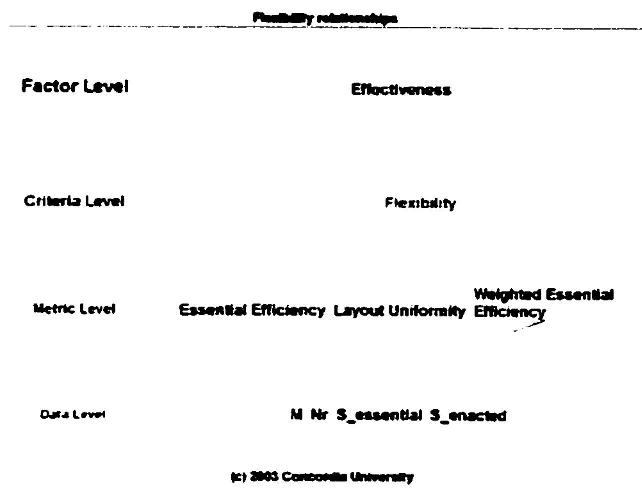

Figure 7 – Attribute Graph

Figure 7 illustrates a QUIM map of the attribute "Flexibility" in a graphical conceptual form. High-level goal is decomposed into a vertical direction with its related factors and criteria until they reach their metrics. From such an image, it is clear to a user that the quality factor "Effectiveness", as a user oriented attribute, is affected by its relevant metrics "Essential Efficiency", "Layout Uniformity" and "Weighted Essential

Efficiency". In contrast to the hierarchical tree structure, the QUIM graphic map can present that data attributes "M" and "Nr" are used for calculation of the "Essential Efficiency" and "Weighted Essential Efficiency" metrics. The other two data attributes, "S_essential" and "S_enacted", affect the metric "Layout Uniformity" and "Weighted Essential Efficiency". Using the model graph, users and developers may better understand how each attribute influences the usability of a software system or characteristic of the user interface. The model graph may also help quality engineers to clarify relationships between factor and data.

2.2.3 Graphical Dynamic Quality Analysis (GDQA)

The Graphical Dynamic Quality Analysis (GDQA) method, developed by the usability team at Concordia University, has been used for building high quality graphical presentation of quality requirements [16]. It assumes that each quality requirement can express a multi-variable function, which is independent from others in terms of their objectives. Each quality factor may have other dependencies, such as sharing primitive measures or indirect measures that are calculated and predicted to achieve Quality Prediction. While QUIM defines the components of quality system, GDQA analyses integration of these components into systematic structure. Figure 8 represents an example of GDQA model.

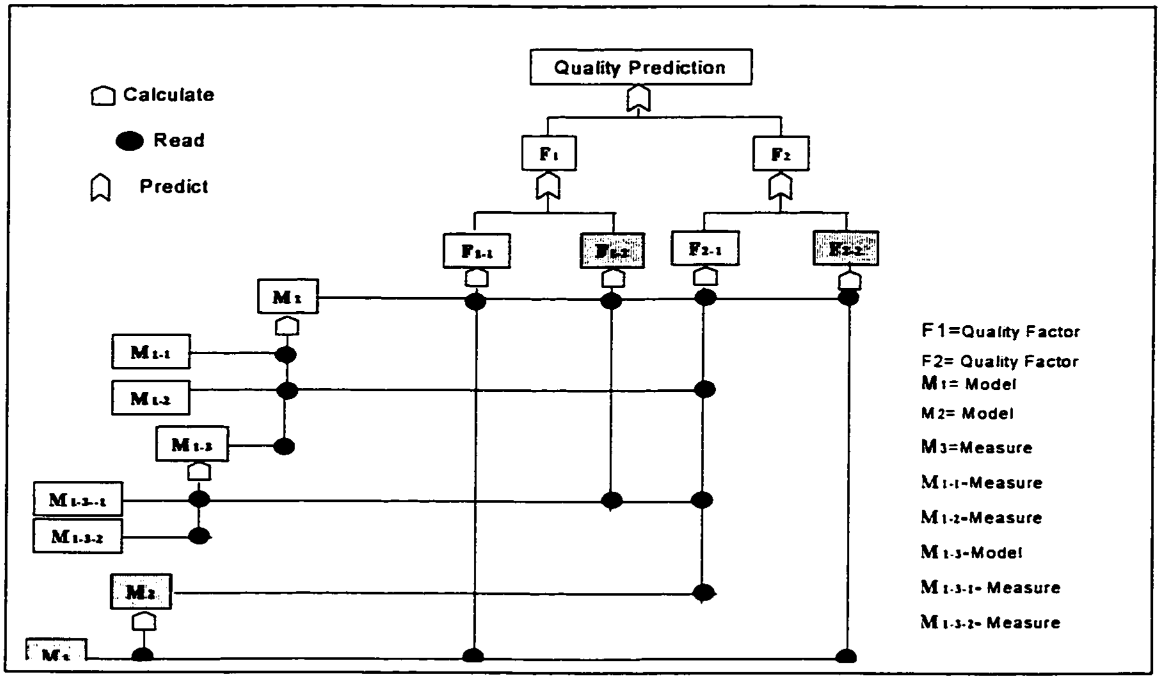

Figure 8 - GDQA method

Constructing a function with GDQA is relatively easy for a software developer. The logic-based graphical method provides a conceptual framework for analyzing and understanding the relationships between the components of any complex system. It can be used as an advanced technique for visualization and the QUIM editor can be extended to support this approach [15].

2.3 Original Applications of the QUIM Editor

Most of the usability software tools are designed to support collections and measurement of usability metrics based on a specific model. However, when it comes to metrics specifications and their relationships within different models and standards, users are referred to browse through thousands of website, book and publication sources. With the QUIM editor, users can easily explore usability attributes that belong to different models by an easy to use software tool. In addition, QUIM can be used:

- As an educational tool for software developers and end users to learn and understand the existing software engineering and human computer interaction quality models and their components, in particular those that address usability;
- To visualize internal relationships and understand their dependencies within the Quality in Use models, which is not easy to see in current standards;
- As a complementary tool for existing measurement and testing tools, such as MUSiC and SUMI which are mainly used by software and usability testers during the usability evaluation step;
- To compare different models, as models are described in a uniform manner. A quality manager may easily compare different models and select the most appropriate one for his or her type of applications.

During the design of a user interface, many usability aspects of the QUIM editor application had to be taken in consideration. The following features enable efficient and effective browsing through QUIM database:

- The QUIM editor application window is adaptable to user screen and resolution. It can be resized to make other applications visible simultaneously. The details frame can be hidden from the menu "Tools > Hide Details", which makes more space for attributes listing. In addition, the frame size can be changed by dragging splitter line between the frames;

- Cleaner and uncrowded interface is achieved by using tabs as a way of presenting layers of factor, metric, criteria and data in the quality in use map frame and specification and relationship in the details frame;

- When an attribute is selected and relationship boxes display all related attributes, a user is able to jump to that attribute by double clicking on any related attribute. In this way, a user may walk through different layers very quickly by the relationship path;

In addition to emphasizing the strength of the QUIM editor, these features establish a foundation for the knowledge map management (please see Chapter 3).

3 QUIM Editor as a Tool for Management of a Measurement Knowledge Map

"Building usability into a product requires an explicit engineering process. That engineering process is not logically different than any other engineering process. It involves empirical definition, specification of levels to be achieved, appropriate methods, early delivery of a functional system, and the willingness to change that system."

Dennis Wixon and John Whiteside (1985)

3.1 The Roles of Knowledge in Measurement

The primary requirement for design for usability is a knowledge and awareness of measurement, and methods for harnessing that knowledge in the production of useful and usable artifacts. Having described the constitutes of usability in different standards and models, this chapter presents the ways of gathering the knowledge about measurement and metrics, and conveying this knowledge to developers and quality managers.

3.1.1 Measurement Process

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules. Formally, measurement is defined as a mapping from the empirical world to the formal, relational world. Consequently, a measure or metric is the number or symbol assigned to an entity by this mapping in order to characterize an attribute [17]. Thus, the real world is the domain of the mapping and the mathematical world is the range. When an attribute is mapped to a mathematical system, there are many choices for the mapping and the range. For example, real numbers, integers, or even a set of non-numeric symbols may be used, in a subjective questionnaire filed by software users during the testing process.

Measurement can be seen as a necessary detour, since humans are usually unable to make clear and objective decisions, interpretations and judgments. In this direction, Kriz [18] summarized benefits of measurements and suggested the measurement process presented in Figure 9.

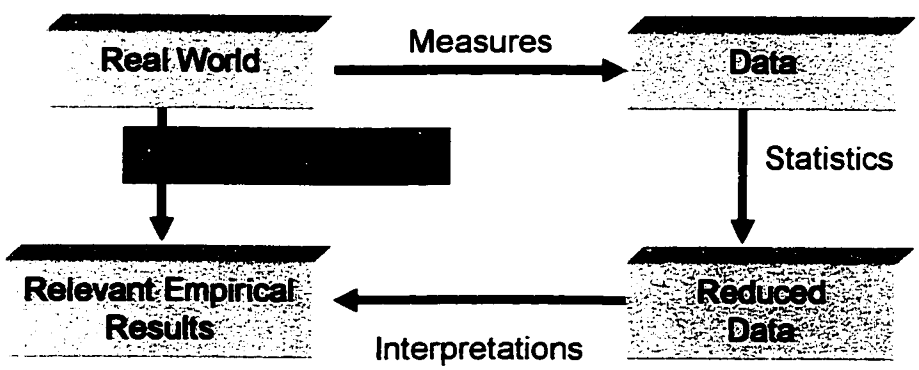

Figure 9 - Measurement diagram

This framework shows that measurement is more than producing quantitative and qualitative numbers. It highlights that measurement requires knowledge about how to combine empirical entities with numerical values. The process starts with the "real world" where developers or usability experts want to have "relevant empirical" results on usability problems. For example, developers want to have relevant empirical statements about the learnability of software. The real world contains the objects that should be measured. However, human brain is neither able to produce directly empirical results nor to reduce information without any help. Kriz [18] called this problem the knowledge intelligence barrier. One possibility to by-pass this problem is to use mathematical and statistical models. These models are used to reduce the data (reduced numbers), which can be statistical results such as means, variances, correlation coefficient, etc. The last and important step is to give the reduced numbers an empirical interpretation.

3.1.2 Knowledge Classification about Metrics and Measurement

IEEE metrics standard [19] defines software metric as:

> *"a function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which the software possesses a given attribute that affects its quality "*

Measures of quality of use provide the usability metrics, which quantify whether a design is successful in achieving usability. The main advantage of the metrics-based approach is that it provides a definition of usability against which the system external attributes can be measured.

There are two kinds of measurement: direct and indirect measurement. Direct measurement refers to the measurement of an attribute entity that involves no other attributes or entities. For example, measuring the number of buttons on the screen, the length of a table, etc. requires no knowledge about other attributes. Calculation or indirect measurement refers to the measurement of an attribute of an entity that involves other attributes or entities. Indirect measurement is useful in measuring visible interaction between direct measurements. For example, the metric "Task Effectiveness" involves "Percent of task completed" and "Percent of goals achieved".

When an entity is measured, it is the attributes of an entity that are actually measured. When inputs are quantitative values, the measurement is straight forward, but in the case of empirical results, the usability metrics can be difficult to interpret and translate into specific design recommendations. The richer the empirical relation system is, the more restrictive the set of representations and the more sophisticated the scale of measurement will be. There is always a representation problem when trying to represent the measurement of a system with a scale. In addition, software engineers skilled in usability are required.

Therefore, a knowledge base about measurement is crucial for successful usability evaluation. The QUIM framework provides a collection of measurement logistics defined during knowledge map management. The difference between QUIM and other static measurement models is in that the measurement knowledge within QUIM is manageable and dynamic.

3.2 QUIM Knowledge Map Organization

3.2.1 Knowledge about Factors, Criteria, Metrics and Data

The usability attributes, which contribute to quality in use, will include the style and properties of the user interface, the dialogue structure, and the nature of the functionality. The characteristics of a factor, a criteria or a metric usability attributes specification are:

- **Name** should be unique and self explanatory, so it can be used to distinguish usability attributes in an existing model. Different names are used to express the same meaning. For example Shneiderman [14] defines a "Time to learn" to refer to the "Learnability" defined by Nielsen [10]. Such inconsistency may produce confusion from the user perspective and should be avoided by adding references within an attribute definition;

- **Definition** must be used to meet customer expectations of quality. Including a list of references, conferences, sources, books, papers or other documents will support the validity of such a definition. Any aliases or other names of a definition of usability attribute that are the same or similar should be included in the definition. Many attributes that are already a part of established usability models or standards found in HCI community should be quoted and indexed. The definition should also include discrepancies affected by a different context of use, environment and product stage, especially when methods of collection are described. For example, if a product is in the early stage, methods used for measurement based on paper prototype may be different from suggested measurement methods based on the final product. A list of parties needed for conducting the measurement process, needed time and its cost are not mandatory, but can be useful as these information may vary between different projects.

- **Interpretation**: Relevant empirical statements cannot be made without interpretation. A good measurement theory should state conditions and rules for the translation of numerical statements back to empirical statements. Such translation is done using different measurement scales that are described in this chapter. Some elements of the interpretation are: 1) description how to recognize a usability attribute; 2) questions used during testing with user; 3) suggested answers of choices that reflect empirical results; 4) mapped numerical values defined by an appropriate scale. For example, the metric "Task help frequency" defined as "Frequency of using help and documentation for a task" may have a scale of three answers: "Never" "Sometimes" and "Often", and these choices can be mapped to percentage value of "10%", "50%" and "90%" (taking into consideration a possible variation in the received answers). This information is essential for conducting a precise measurement result.

- **Calculation**: If the usability attributes can be measured using a countable method, or by an interpretation, calculation may not be applicable. However, in case the measurement is done using a calculative method, like in the case of metric attributes using data attributes, calculation is used to describe the mathematical formula used to calculate attribute value. For example, the metric "Essential

Efficiency" (EE) [20] is calculated using the formula:

$$EE=100 \times S\_essential / S\_enacted$$

S_essential = The number of user steps in the essential use case narrative;

S_enacted = The number of steps needed to perform the use case with the user interface design;

3.2.2 Knowledge about the Relationships between Factors, Criteria and Metrics

Relationships among attributes are essential for understanding how usability is measured and how it can be improved. As mentioned above, QUIM is not exactly a hierarchical model. A specific metric could affect more than one criteria and one data can contribute to many metric attributes.

Figure 10 presents an example of how relationships among the QUIM components can be very complex. The Data "Number of visual components" is an input to two different metrics - "Visual Coherence" (*description: how well a user interface keeps related things together and unrelated things apart*) and "Layout Uniformity" (*description: uniformity or regularity of the user interface layout*) [20]. Those metrics measure the criteria "Minimal Memory Load", which affects the factors "Efficiency" and "Satisfaction". In addition to the criteria "Minimal Memory Load", the metric "Layout Uniformity" affects the criteria "Attractiveness". In this relation, the metric "Layout Uniformity" plays an important role because it is the only attribute that affects the criteria "Attractiveness". In another relationship, the metric "Task Concordance" (*description: index of how well the distribution of task difficulty using a particular interface design fits with the expected frequency of various tasks*) affects not only the criteria "Completeness", but also the factor "Effectiveness", since the factor "Effectiveness" depends only on the criteria "Completeness" (descriptions of factors and criteria attributes is listed in appendix).

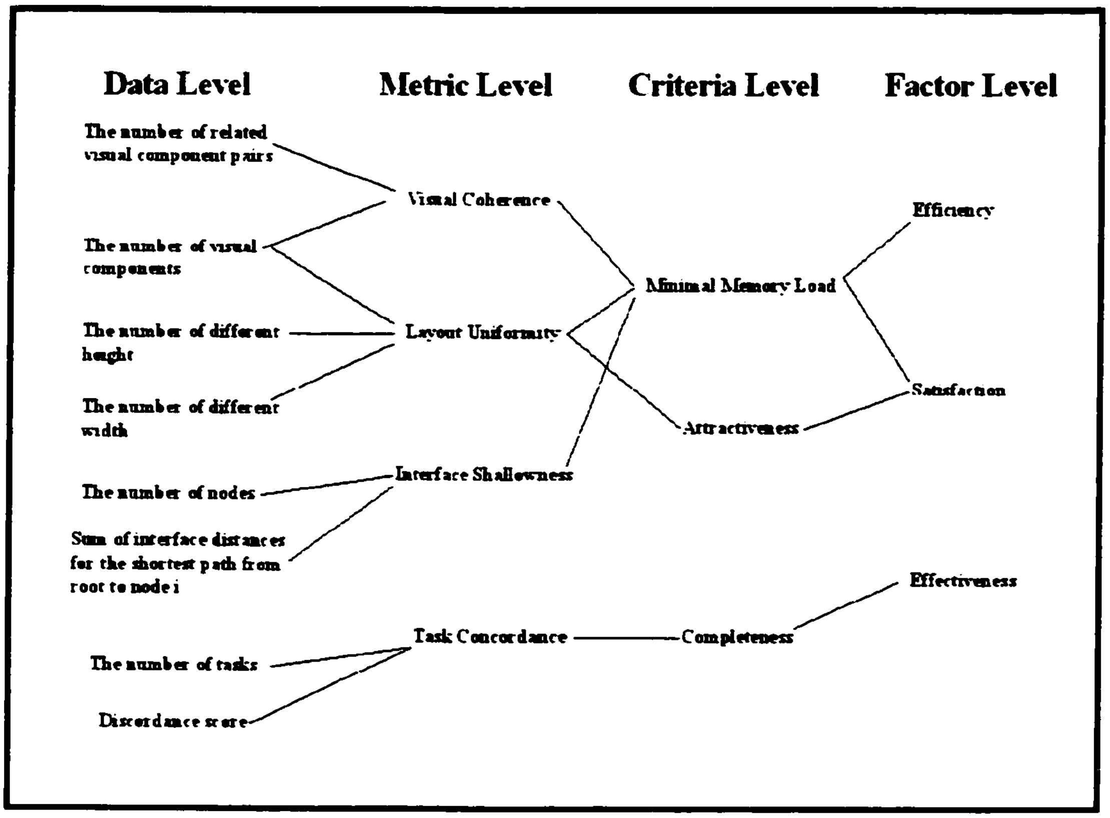

Figure 10 - Relationship tree among QUIM attributes

Figure 10 indicates the importance of the criteria "Minimal Memory Load", since it will contribute to two factors. The "Minimal Memory Load" has more importance for the suggested usability model compared to the criteria "Completeness". However, such prioritizing is not necessary based on a number of related attributes within the model. For example, the metric "Layout Uniformity" may be less critical for usability of the system than the metric "Task Concordance", even if it is related to more attribute(s) within the model.

Choosing relationships and prioritizing them within a usability model is an extremely responsible task since it defines coherence of the usability model. Detailed knowledge of usability in whole, existing models and their attributes specification is necessary prior to defining new relationships.

3.2.3 Empirical Interpretations

In order to assess whether quantitative objectives have been met, the usability expert needs relevant empirical results. These results may be gathered from an interpretation of collected data from end-users using some of the methods described below (see page 37). Interpretation process that maps states-in-the-world to numbering systems is often integrated in measurement techniques, such as a number of reliable questionnaires for measuring user satisfaction [14] [21] [22] [30], or a part of the ESPRIT MUSiC project [23] attempted to lay down guidelines about how to measure aspects of human performance.

Several scales are used to express interpretations. Measurement scale is the combination of measurement mapping with empirical and numerical relation systems. The measurement scales enable managers and developers to obtain meaningful results and statements to be further applied within a measurement process, and to perform appropriate operations upon the measurement results. The following scales are most frequently used during usability measurement process [24]:

- **Nominal scale**: Nominal measurement consists of classifying items to groups or categories. Any distinct numbering or symbolic representation of the class is an acceptable measure. Classification is discontinuous in nature, which means that a variable has only certain discrete values, but no intermediate values between the variables. No quantitative information is conveyed and no ordering of the items is implied. Nominal scales are therefore qualitative rather than quantitative. An example of the nominal scale is the "User experience level" with the options "Beginner", "Intermediate" or "Expert";

- **Ordinal scale**: Measurements with ordinal scales are ordered: higher numbers represent higher values. The empirical relation system on an ordinal scale consists of classes that are ordered with respect to the attribute. However, the intervals between the numbers are not necessarily equal. There is no "true" zero point for ordinal scales since the zero point is chosen arbitrarily. Most (but not all) ordinal scales are discontinuous in nature, which means that they contain only discrete values, but no intermediates. An example of the ordinal scale is "The number of user steps in the essential use case narrative";

- **Interval scale**: On interval measurement scales, one unit on the scale represents the same magnitude on the trait or characteristic measured across the whole range of the scale. Examples of interval scales are the continuous scale of a thermometer in the "Critical Safety System" or the discontinuous scale of the attribute "Years of experience". Both of them have zero points, but these zero points are arbitrary and they could be placed anywhere. If the temperature today is 30 degrees and yesterday it was 10, we could not say that it is now three times warmer.

- **Ratio scale**: Ratio scale preserves the order, the size of intervals between entities and the ratios between entities. In the ratio scale, there is a zero element, which represents the complete lack of attributes. The measurement mapping for the ratio scale must start at zero and increase at equal intervals. An example of the ratio scale is the "Time interval" or the "Distance between visual components";

- **Absolute scale**: Absolute scale is obtained by counting the number of elements in an entity set. The attribute always takes the form "number of occurrences of x in the entity". There is only one possible measurement mapping, namely the action count. Absolute scale is the most restrictive among all scales. An example of the absolute scale is the "Number of elements on the screen".

There are other approaches to the usability evaluation, which focus on identifying usability defects. The defects can be identified by an expert assessment, aided by guidelines and checklists (e.g. heuristic evaluation) or by a user-based testing [10]. When these two approaches are optimized they can be equally cost effective [25], although the user-based testing is generally better at identifying more serious defects [26].

3.2.4 Knowledge about the Data Collection Methods and Processes

There are two broad classes of methods in expert-based evaluation. The first method requires direct participation of the end user, and the second method is only applied to attributes of the graphical interface and does not need the direct involvement of users. The first category of methods may be used to evaluate the product as a whole, but the second is more suitable for predictive evaluation of usability. The second category includes an expert evaluation, analytical, and usability inspection techniques. All of these methods typically involve the usability experts concerned with evaluating the conformance of the interface to a set of guidelines, principles, or heuristics and standards about usability [25]. The most frequently used methods and tools are:

- **Contextual Inquiry** - one of the most appropriate methods to use to understand the user's work context. It is a structured field interviewing method based on several core principles that differentiate this method from plain, journalistic interviewing. Contextual inquiry is more a discovery process than an evaluative process; more like learning than testing. This technique is best used in the early stages of development, to gain an understanding of how people feel about their jobs, how they carry out their work, how information flows through the organization, etc. [27];

- **Interviews and Focus Groups** - a focus group brings together a cross-section of stakeholders in a discussion group format. A facilitator elicits views on relevant topics. The meetings may be taped for later analysis. Focus groups are useful early in the requirements specification but can also serve as means of collecting feedback once a system or product has been in use or has been placed on field trials for some time. Focus groups help to provide a multi-faceted perspective on requirements and identify issues that may need to be dealt with [28];

- **Observation** - a method that involves an investigator that observes users as they work and takes notes on the activity that takes place. Observation may be either direct, where the investigator is actually present during the task, or indirect, where the task is viewed by some other means such as through use of a video recorder or other recording tools. For example, the data attribute "Total number of functions" or the metric "Task Time" may be collected by observing recorded tapes during a test with a user;

- **Surveys** - A survey involves administering a set of written questions to a large sample population of users. Surveys can help determine information on customers, work practice and attitudes. There are two types: "closed", where a respondent is asked to select from available responses and "open", where a respondent is free to answer as they wish. [29]. During the measurement process, many metric and data attributes can be collected using interpretation scales that relate the answer received from a user to its numerical value;

- **SUMI questionnaire** - SUMI (Software Usability Measurement Inventory) questionnaire is designed to collect a subjective feedback from users about a software product with which they have some experience. Users are asked to complete a standardized 50-statement psychometric questionnaire. Their answers are analyzed with an aid of a computer program - SUMISCO. SUMI data provides a usability profile according to five scales: perceived efficiency, affect (likeability), control, learnability and helpfulness. It also provides a global assessment of usability. The international database with which the software is being compared contains some 300 software products. SUMI may be used to assess user satisfaction with high fidelity prototypes or with operational systems. In addition, it may be used in conjunction with other usability tools and methods [30];

- **Cognitive Walkthrough** - a process of going step by step through a product or system design to get reactions from relevant staff and typical users. Normally one or two members of the design team will guide the walkthrough, while one or more users will comment as the walkthrough proceeds. This method is applicable in early stages of product development and can be used for data attributes like the "Number of functions identified by the user" or the "Total number of complex operations" [31];

The information above indicates there are differences among evaluation and data collection methods related to a stage of development process. Some methods are applicable during the whole development cycle, but precision of their measurement values may depend on the stage of development process. For example, the Cognitive Walkthrough based on a paper prototype will have a different impact on users compared to the Cognitive Walkthrough based on a finished product. Some methods are only applicable in specific product development stage, like for example surveys and some questionnaires.

3.2.5 Towards a Measurement Ontology

Ontology describes formal specification of how to represent the object, concept or other entities that are assumed to exist in some area of interest, and the relationships among them. QUIM may be formalized using ontology. QUIM ontology is presented as a model for representing quality, since quality needs to be measured prior to evaluation and management. An assessment system for measuring attributes of an entity, activities for measurement, and quality as conformance to requirements are the core concepts represented in the ontology. The formal representation of measurement is based on a detailed context of its components. The context is expressive and precise, minimizing ambiguity in interpretation. Data model of the QUIM domain, presented in Figure 11, consist of usability attributes decomposed as factors, criteria, metric and data and their inter-relationships.

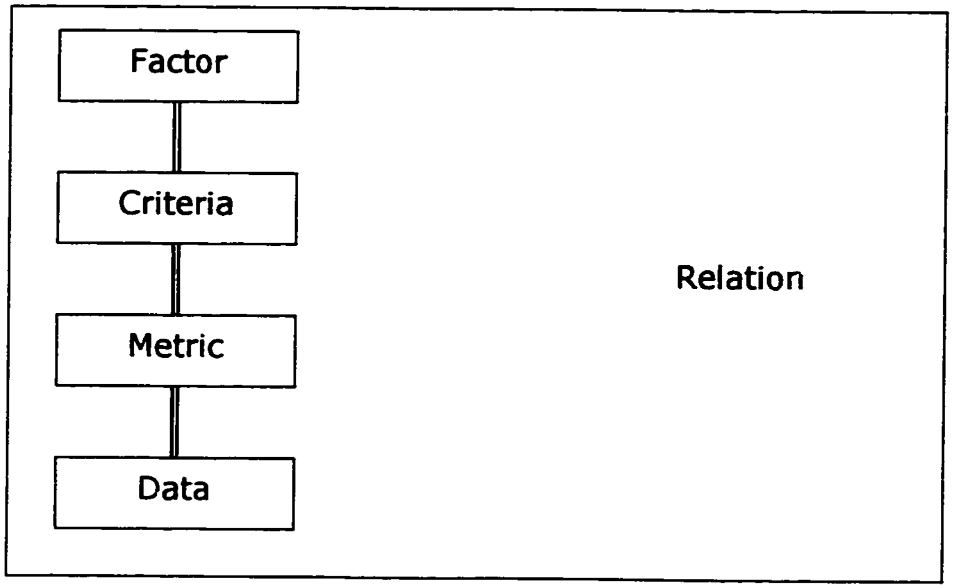

Figure 11 - QUIM ontology

Factors: A quality in use factor is a user-oriented attribute or characteristic, which is used to estimate one contribution to usability. A factor is easy to understand by a usability expert. However, it is difficult to measure and specify. Examples of factor attributes are the "Effectiveness", "Efficiency" and "Satisfaction". A factor could be refined into the sub factors or criteria.

Criteria: Criteria are sub factors or sub characteristics that are directly measurable via a set of metrics. Some examples of criteria attributes are the "Understandability", "Operability", "Attractiveness", "Consistency", "User Guidance" and "Compliance". Criteria may be determined by a metric.

Metric: Metric is a numeric value that summarizes the status of a specific user interface attribute. The advantage of using metrics is that they are faster, cheaper and less ambiguous than other usability evaluation attributes [32]. Different types of users, tasks or environments may influence context of specific metrics. Some examples of metric attributes are the "Layout Uniformity", "Task Concordance", "Visual Coherence", "Task Effectiveness" and "Interface Shallowness". In most cases a metric does not require test users and additional usability experts since evaluation is done using direct and indirect measurement. When metric attributes are measured using indirect measurement (calculation), data attributes may be involved in measurement.

Data: The lowest layer of QUIM is the list of data that are used to calculate metrics. Data values can be acquired from prototypes, use case, task analysis, system specification, user documentation, final system or users, using countable or calculable methods to acquire such data. Examples of data attributes are "The number of steps needed to perform the use case with the user interface design", "Distance between visual components", "Number of components", "Percent of task completed", "Time to complete a specific task" and "Total number of message boxes" [32].

Relationships: Relationships are rules of mutual contributions among factors, criteria, metric and data. They establish dependencies for each attribute, which form a model used for quality in use measurement. Also they establish a base of model analysis and prediction.

3.3 Scenarios of Creating and Maintaining a Map

The role of the QUIM knowledge base is to provide a practical means of achieving the quantitative and qualitative data required to support usability engineering. Such knowledge is expressed in a form of usability standards and/or models, as a collection of interrelated usability attributes, which form the QUIM knowledge Map. Each map can represent a replica of any usability model established using the QUIM editor Map Management tools.

The QUIM editor Map Management is collection of tools primarily aimed at usability experts, for defining new and modifying existing usability attributes within the QUIM Knowledge Map (Figure 12).

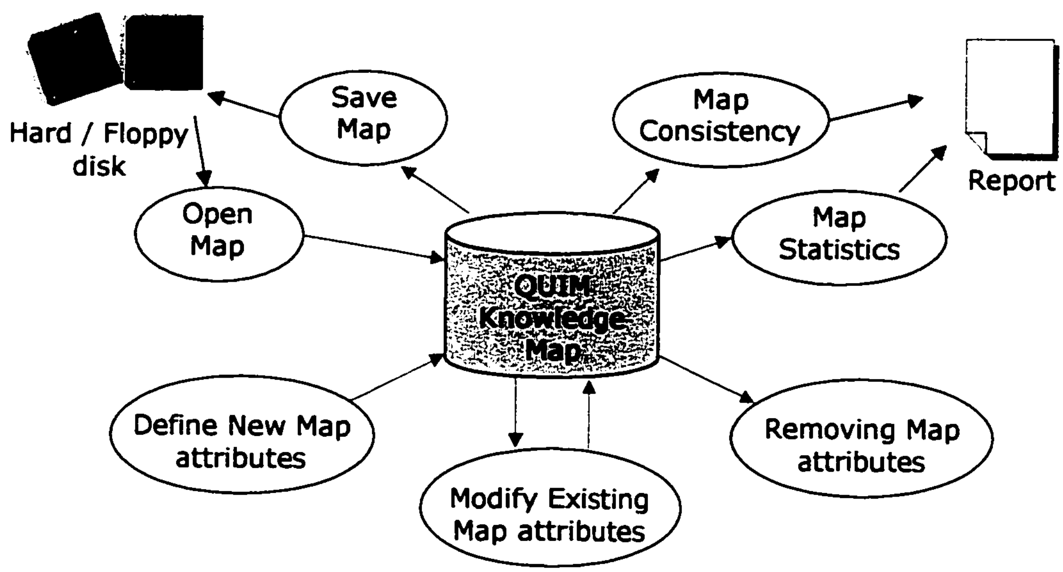

Figure 12 – QUIM Knowledge Map Management

When a user starts the QUIM editor, a default knowledge map provided in installation package is loaded and ready for browsing. This map is created by our usability research team, using the same tool, QUIM editor. The QUIM editor has the menu with four sections: Map, Model, Tools and Help. Map section refers to Map Management and contains functions used for the following knowledge base management operations:

- Creating and Opening new versions of Map database;
- Protecting Map Administration section;
- Adding new usability attributes in knowledge base database and entering their specification and relationships with other attributes;
- Modifying existing usability attributes and their specification and relations;
- Deleting attributes from knowledge base;
- Map consistency and attributes statistics.

3.3.1 Creating, Archiving and Protecting a Map

The QUIM editor is designed to update the knowledge map database file on a user hard drive, after every database manipulation. If a user exits from the QUIM editor or restarts computer, changes and updates are saved. Database file that contains all knowledge base information is located in the Application directory, which is per default installation "C:/Program Files/QUIM editor/QUIM.mdb".

For user information, the name of active database is written in brackets in capitation of the main application window, after the title "QUIM v.2.0". Another way to find out the name and location of an active database file is trough the "Preferences" window.

In an environment where the QUIM editor is accessible by users other than usability experts, the Map Administration can be protected by a password, as presented earlier. When user selects the "Set Map Password" under the menu "Map", a Map

Administration Password window appears (Figure 13).

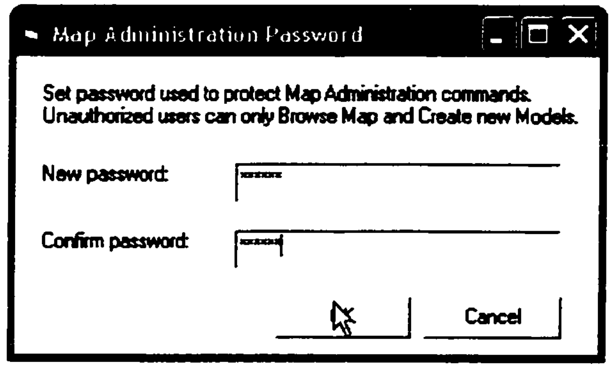

Figure 13 - Set Map Administration Password

If the password is entered and confirmed, it will be saved in the QUIM configuration file. Leaving blank fields in the "Set Map Password" prompt can disable password protection. Next time, when an attempt to access the Map Administration occurs, a prompt to enter this password will appear. If the password is correct, the menu "Map" expands with Administration commands.

3.3.2 Defining New Factors, Criteria and Metrics

Adding new usability attributes is technically easy, but it is a challenging decision because the success of measurement model depends on its attributes. Usability experts must first decide if an attribute is needed, keeping in mind whether the model is used for measurement of specific domain, product, particular user factor or existing usability standard. It is important to distinguish the type of attribute (factor, criteria, metric or data), whether the attribute is user-, engineer- or metric-oriented, what are its dependencies and which other attributes it contributes to. If an attribute is complex and ambiguous, it is wise to split it into two or more attributes if possible. In addition, any similarity in the name or definition with existing attributes must be avoided.

There are four menu selections for creation of usability attributes, based on the usability attributes layers within the QUIM framework. Each selection starts multi step wizards:

- Add New Factor creates a new factor in the QUIM database. It includes only the "Factor Specification" step and the "Related Criteria" step, since it is the highest layer within the QUIM framework;
- Add New Criteria creates new criteria in the QUIM database. It includes the "Criteria Specification" step, the "Related Factors" step and the "Related Metrics" step;
- Add New Metric creates a new metric in the QUIM database. It includes the "Metric Specification" step, the "Related Criteria" step and the "Related Data" step;
- Add New Data creates new data in the QUIM database. It includes only the "Data Specification" step and the "Related Metric" step, since it is the lowest layer within the QUIM framework.

In the example presented in Figure 14, after selecting the "Add New Metric", a user can start writing the new metric "Specification". The "Name" should be simple and self-explanatory. The "Definition" should provide description of a usability attribute in more details. The "Interpretation" and "Calculation" should provide critical information for testing and measurement.

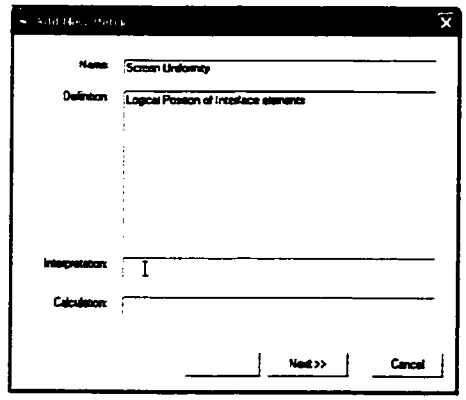

Figure 14 - Add New Metric Specification

There are several important characteristics associated with useful usability attribute. Usability attribute must be [33]:

- Simple to understand and precisely defined, in order to facilitate consistency both in the calculation and the analysis of metric values;
- As objective as possible, in order to decrease the influence of personal judgment to the calculation and analysis of metric values;
- Cost effective, in order to have a positive return of investment (the value of the information obtained must exceed the cost of collecting the data, calculating the metric, and analyzing its value);

- Informative, in order to ensure that changes to metric values have meaningful interpretation.

The next two steps (or one step in the case of factor or data) are used to define the related attribute from upper or lower layer. In the example presented in Figure 15, the step that follows the completion metric specification is the defining criteria that will be affected by this metric. All the existing criteria from the QUIM database are listed in scroll window. Each criteria attribute has a checkbox in front of the "Criteria Name", used to select all related criteria. If new metric attribute should influence more than one criteria attribute, a user can select the checkbox "Select All" located under the list, instead of selecting one by one. Then, a user can deselect unrelated criteria attributes. As a help to user, attributes definition is presented as the "Description" column in the criteria list box.

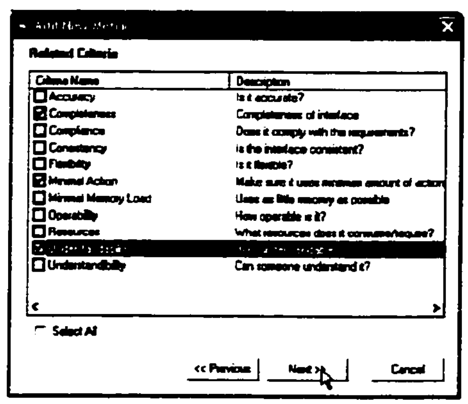

Figure 15 - Add New Metric related Criteria

As a final step, the "Finish" button adds the new attribute in the database and closes the "Add New" wizard. Once the attribute is added in the database, it may be reviewed using the Browsing function.

3.3.3 Modifying an Existing Map Attributes

All attributes in a model affect the quality in use, even if they may appear unrelated. Modification of any attribute or its properties inevitably influences other attributes, particularly those that are closely related. Prior to proceeding with a modification, a user should take into account the consequences. If any attribute contribution to a measurement model is removed, a replacement should be provided. For example, if a decision is made to delete the relationship among the criteria "Attractiveness" and the factor "Satisfaction" (Figure 10), a user will be fully satisfied with a faster system response (the criteria "Minimal memory load), even if the interface is not visually attractive (the criteria "Attractiveness"). In this case, an existence of the criteria "Attractiveness" is questionable, as it will be unrelated to any factor of the model. Thus, the "Attractiveness" may be removed from the model.

Selecting an attribute in the "Quality in Use Map" window followed by selecting the "Edit a Selected Map item" from the menu "Map" can modify any factor, criteria, metric or data. Similarly, pressing a right mouse button over an attribute, and then selecting the "Edit" from a small floating menu can achieve the same effect. Either way will start the wizard similar to the "Add New" wizard described earlier.

Novice QUIM editor users may neglect the fact that relationship among two attributes will appear in the "Add New" or the "Edit Selected" wizard of both attributes. For example, if the metric "Screen Uniformity" is edited, the "Related Criteria" step will show that it is related to the criteria "Completeness". If the "Edit attribute" window is closed and the criteria "Completeness" is edited, the "Related Metric" step will display that it is related to the metric "Screen Uniformity". If this relationship is "unchecked", followed by saving the change by pressing the "Finish" button and then attempt to edit the metric "Screen Uniformity" is made, the relationship to the criteria "Completeness" will become unchecked.

3.3.4 Removing Attributes from the Knowledge Base

Removing attributes from the knowledge base is a function often used for extracting a specific model from an existing model. Deletion of an attribute carries responsibilities to other attributes, because once the attribute is deleted, its relationships to other attributes will be deleted, too. Such action may leave other attributes unrelated. For example, if the metric "Task Concordance" is deleted, the criteria "Completeness" and the factor "Effectiveness" will stay related only to each other (Figure 10), leading to an unacceptable measurement model.

Similarly to editing, any factor, criteria, metric or data may be deleted by selecting an attribute in the "Quality in Use Map" window, followed by selection of the "Delete a Selected Map item" from the menu "Map". Pressing a right mouse button over an attribute, followed by selecting the "Delete" from small floating menu (Figure 16) is another way of achieving this.

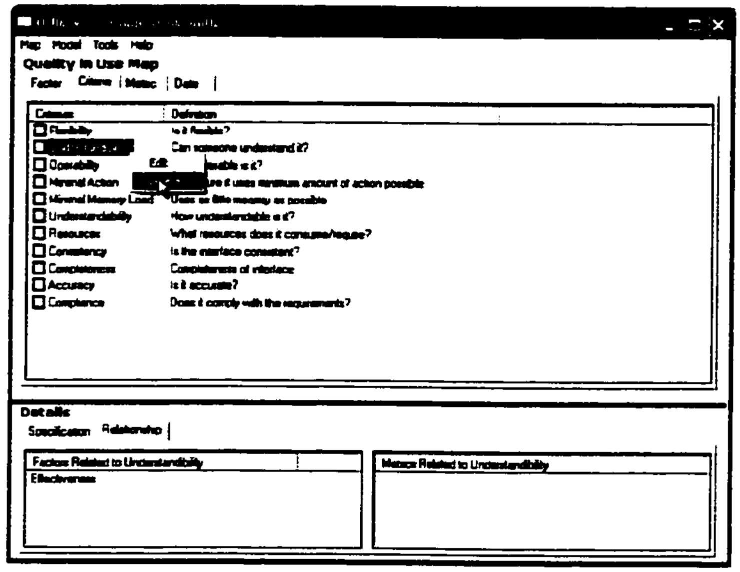

Figure 16 - Deleting Selected attribute

In the case some attributes in lower layers of the Map are related only to the attribute that user wants to delete, the window "Delete Confirmation" will appear on the screen. Figure 17 presents the scenario where user tries to delete the criteria "Flexibility". Since the metrics "Essential Efficiency", "Weighted Essential Efficiency" and "Layout Uniformity" are related only to the criteria "Flexibility", the QUIM editor will automatically delete these attributes as well. This feature is designed to prevent creation of orphan attributes in the knowledge base (attributes that are not related with any attribute in upper layer).

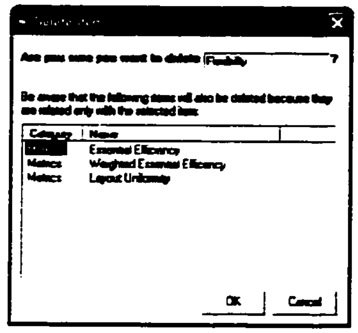

Figure 17 - Delete confirmation

3.4 Advanced Features

By providing a knowledge map and all of the tools required to access, share, discuss and work with metrics, the QUIM editor enables developers to better understand the measurement theory. In addition, it enables managers to integrate usability metrics into the software development.

Theory of the QUIM map may establish relationship among existing usability models and models that emerged from practices. Models within the QUIM framework may be modified and updated according to insights from industry. The QUIM editor already supports tools for model comparison, extraction and adaptation to specific domain or context of use (please, see Chapter 4).

Usability experts must perform such iterations manually, but integration of statistical analysis may be considered as an advance potential feature of the QUIM editor that should be implemented in the next version of QUIM editor. Developments in the field of statistical data analysis often lead to better decision making under uncertainties via a data-oriented approach. Usability experts react based on feedback information, which may be explained in structured form. The sequence from data to knowledge is: from data to information, from information to facts, and finally, from facts to knowledge. Data becomes information when it becomes relevant to a decision problem. Information becomes fact when the data can support it. Fact becomes knowledge when it is used in the successful completion of a decision process. If we apply the basic knowledge, concepts and methods of statistics using widely available commercial statistical computer packages such as SAS [34] and SPSS (Statistical Package for the Social Sciences) [35], we will inevitably find ourselves asking whether it is possible to develop computer models of the perceptual and other cognitive processes that can read, learn, generalize, and adapt (neural networks). Such machine learning would inevitably automate process of decision-making that can lead to the fastest integration of usability in development cycle.

4 QUIM Editor as a Tool for Creating a Customized Usability Models from a Map

As presented, users and developers may benefit from a knowledge base, and usability experts and researchers may define their own knowledge base. As described in Chapter 3, knowledge base represents one or many usability standards and models. The QUIM knowledge map customization that enables usability experts and managers to build new usability models is presented in this chapter (Figure 18). The usability models may represent usability ISO standard [3], a model focused on particular user factor (User satisfaction with Web), general area or context of use (Critical safety system), or models of explicit domain (QUIM Web model). From another point of view, model customization (modeling) may be used for specific product or system, when usability or development managers need a tool that will help them integrate usability during development cycle and not after a product is developed. The QUIM editor contains all these features.

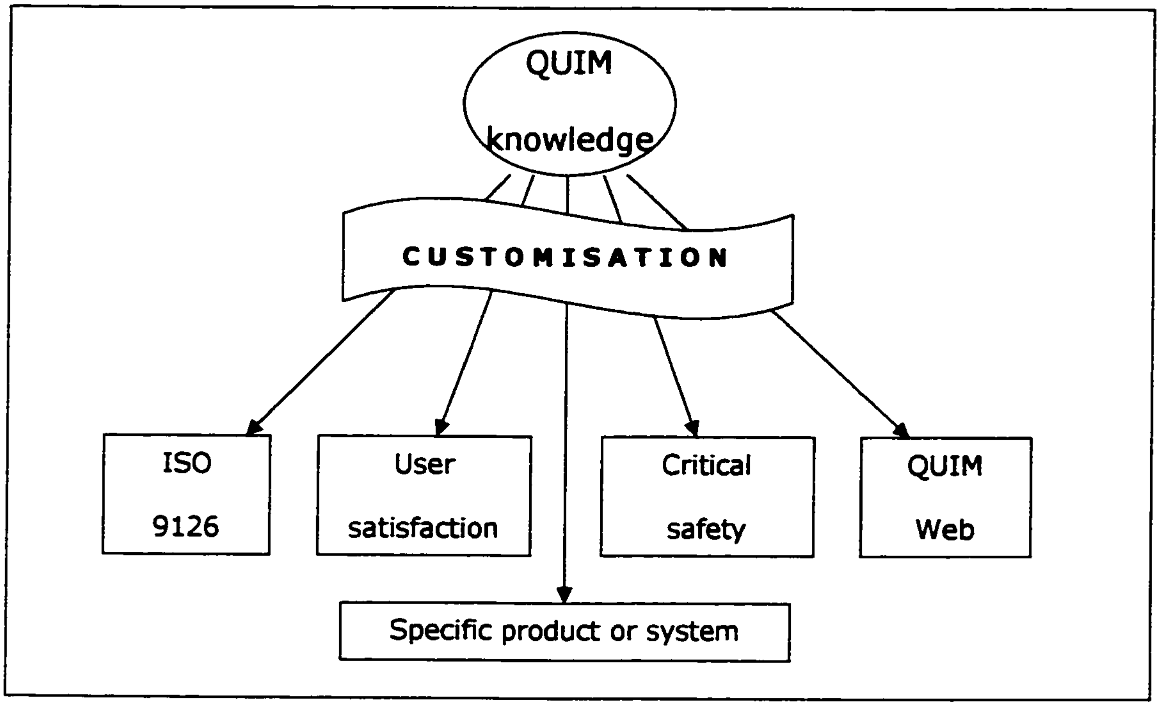

Figure 18 – Map Customized Models

The mentioned issues are addressed in the QUIM editor modeling features. A user may build a specific model by extracting relevant factors, criteria, metrics, and data from the QUIM Knowledge Base (QUIM Map), followed by generating reports and forms to be used for design cycle and usability testing.

4.1 The Process of Customizing a Map

Before we explore the process of creation custom usability models, we should review the process flow and interaction among the functionalities that exist in the QUIM editor (Figure 19). Work with the QUIM editor can be separated into three parts. The first part is the browsing knowledge and getting familiar with usability attributes, their specification, interpretations, methods of collection and relationships of existing QUIM knowledge maps. The second part is the modification of existing and building of new quality in use maps. These knowledge maps may be used for browsing or in the third part for the creation of new Custom Models. New extracted Models may be converted into a new quality in use map that may be used to create new models or as a usability measurement model for specific context of use. Based on a measurement feedback from the empirical world, a custom model may be modified as the development cycle continues. The QUIM editor also allows a user to update the quality in use map based on evaluation practices. Such iteration is not possible with other usability frameworks. In the QUIM editor all knowledge and models information relies on the QUIM knowledge database.

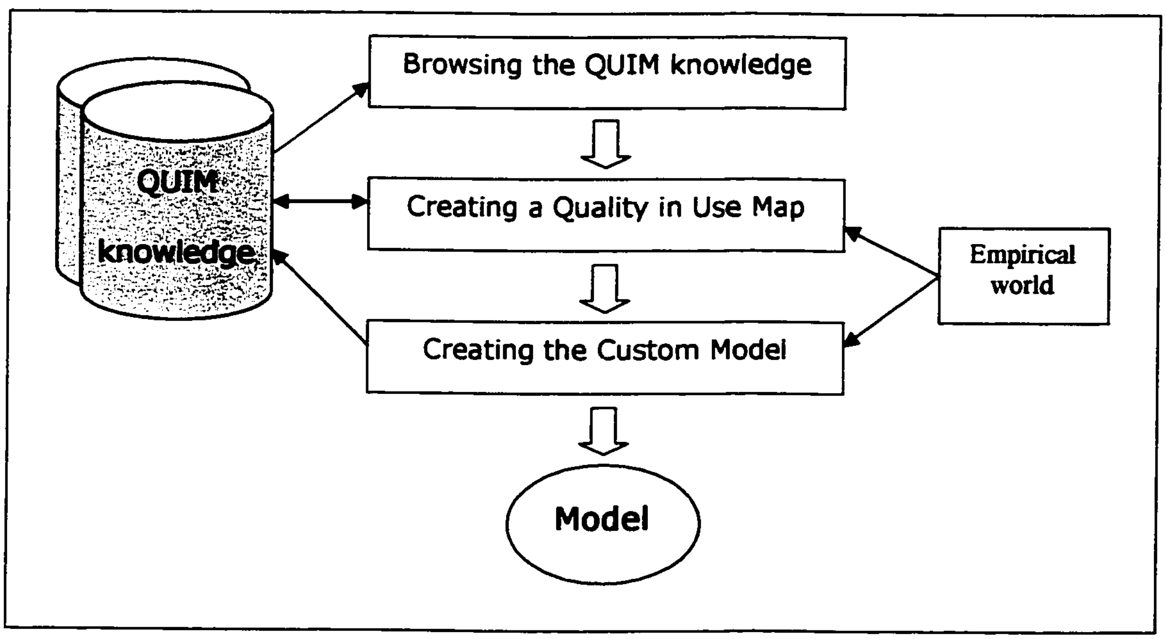

Figure 19 - QUIM cycle

Depending on user experience in the QUIM framework, a model may be created using the "New Model" or the "Model Wizard" function. The **New Model** function creates an empty model and allows a user to add attributes of choice, one by one. Such a function is suitable for usability experts that require full control over the model created for explicit domain, test environment, context of use or new measurement model research. The **Model wizard** function is developed to assist inexperienced users in a model establishment, through a step-by-step process. This approach is also effective for usability managers who need a measurement model for specific product or system.

4.2 A Case Study: MUSiC Model

The QUIM framework integrates existing usability models and standards in a form of knowledge map that may be used to create new usability measurement models. This does not necessary mean that QUIM framework will act as replacement for other measurement frameworks. A case study of the QUIM editor use to support MUSiC Performance Measurement Method [36] is presented here, as an example of the QUIM editor use as a support tool for other measurement frameworks.

The principal aim in developing MUSiC has been to provide a practical means of achieving the quantitative and qualitative data required to support usability engineering. The MUSiC Performance Measurement Method has evolved and has become refined to meet the demands of commercial application. It follows the basic principles of usability engineering and testing, which are well established in theory but too often absent in practice. It may be defined operationally as effectiveness, efficiency and satisfaction with which specified users can perform specified work tasks in given environments [2].

The MUSiC basic outputs, derived in all versions of the method, are measures of:

- **Effectiveness** - how correctly and completely goals are achieved in context;
- **Efficiency** - effectiveness related to cost of performance (calculated as effectiveness per unit of time);
- **Satisfaction** - the user's perception of comfort and acceptability.

Optional MUSiC outputs include further measures and diagnostic data:

- **Relative User Efficiency** - an indicator of learnability (relating the efficiency of specific users to that of experts);
- **Productive Period** - the proportion of time spent without having problems;
- **Snag, Search and Help times** - time spent overcoming problems, searching unproductively through a system, and seeking help. These problem-related measures are valuable sources of diagnostic data about specific areas where designs fail to support adequate performance. In use, the method provides pointers to problem causes [36].

According to the QUIM framework, basic and optional outputs refer to the QUIM factor attributes, as they are end-user perspective of product quality. Effectiveness and efficiency are measures of performance in the given context of use. These basic and optional factors are supported by the following tools and collection methods, which may be chosen and used according to specific development needs, budget and timescales:

- Usability Context Analysis Guide: used to specify usability requirements and to select the tasks and contexts for which usability should be evaluated.

- Performance Measurement Handbook: provided with training, reference source details, procedures to be followed to design and run evaluations to provide reliable performance measures of usability.

- DRUM (Diagnostic Recorder for Usability Measurement) [2] is a software tool that provides a broad range of support for video-assisted observational studies following the MUSiC Performance Measurement Method for usability evaluation. DRUM delivers evaluation data in a format compatible with spreadsheets and statistical packages, for further analysis and graphical display. DRUM assists the generation and delivery of diagnostic feedback to product's designers concerning usability defects.

The effectiveness with which a user uses a product to carry out a task is comprised of two criteria attributes: the "Quantity" of the task the user completes, and the "Quality" of the goals the user achieves [37]. Quantity is a measure of the metric "Amount of a task completed by a user". It is defined as a proportion of the task goals represented in the output of the task. Quality is a measure of the degree to which the output achieves the task goals.

In engineering, the term "Efficiency" is understood as the ratio of useful energy output to energy input. For example, Timoshenko and Young [38] define the efficiency of a machine as the ratio of the useful work performed to the total energy expended. For a work system in which a human is interacting with a computer, the efficiency is interpreted as the amount of effectiveness with which the user and the computer work together. From a user's viewpoint, the amount of effort input may be quantified as the time spent carrying out the task, or mental/physical effort required to complete the task. However, from the viewpoint of the organization employing the user, the input to the work system is the cost to the organization of the user carrying out the task, namely the labor costs of the user's time, cost of the resources and the used equipment and cost of any user training. These types of input produce three different definitions of efficiency, which may be stated with its sub-factors (criteria attributes) "User Efficiency", "Human Efficiency" and "Corporate Efficiency", metrics needed to determine the "Efficiency" are "Task Time", "Task Effectiveness", "Effort" and "Total Cost"

For the satisfaction, SUMI [30] questionnaire is used to evaluate measures of perceived usability across all the tasks performed. Satisfaction is composed of the comfort and acceptability of use. Comfort refers to overall physiological or emotional responses to the use of the system (whether the user feels good, warm, and pleased, or tense and uncomfortable). Acceptability of use may measure the overall attitude towards the system, or the user's perception of specific aspects such as whether the user feels that the system supports the way the task is carried out, whether the user feels in command of the system, is the system helpful and easy to learn, etc. If the satisfaction is low when the efficiency is high, it is likely that the user's goals do not match the goals selected for the measurement of efficiency. Satisfaction can be specified and measured by attitude rating scales such as SUMI.

To measure the user satisfaction, and thus assess the user perceived software quality, the University College Cork has developed the Software Usability Measurement Inventory (SUMI) as part of the MUSiC project [36]. SUMI is an internationally standardized 50-item questionnaire, available in seven languages and takes approximately 10 minutes to complete. The results provided by SUMI are based on an extensive standardization database built from data on a full range of software products such as word processors, spreadsheets, CAD packages, communications programs etc. The whole package contains tools and methods for data gathering and interpretation. SUMI provides an Overall Assessment and a Usability Profile. The Overall Assessment contains 5 sub-scales that can refer to the QUIM criteria: "Affect", "User Efficiency", "Helpfulness", "Control", and "Learnability". These 5 criteria are determined by 50 metrics in the form of questions such as "Using this software is frustrating" and "Learning how to use new functions is difficult".

The optional factor attributes Snag, Search and Help Times may be obtained from a closer inspection of the DRUM diagnostic data results. The Productive Period is the Productive Time expressed as a percentage of the Task Time. The Relative User Efficiency indicates the level of staff performance after a short training, relative to experts. It is an optional factor that can be determined using the criteria "User Efficiency" and "Expert User Efficiency".

The presented case study is an example of how attributes used in MUSiC method may be interpreted within the QUIM framework. For a user that will use MUSiC for measurement of usability, the QUIM editor may provide attributes specification and model relationships graph. In addition, a user may compare MUSiC model with others that exist in the QUIM framework.

4.3 Approach 1: Using Model Wizard

The QUIM editor custom model can be created using the "Model Wizard" or "New Model" function. Success of the Model Wizard directly depends on attribute relationships defined in the knowledge map. Automated top-to-bottom modeling approach starts from end-user oriented attributes (factors) and finish with measurement metrics. This step-by-step process is guided by relationships of attributes that are defined in the knowledge map. If the Model Wizard aims to be used as a tool for product usability measurement of an explicit domain (e-commerce, web, critical safety system), it is wise to choose an appropriate knowledge map (if one exists) prior to proceeding with the Model Wizard.

- In Step 1 (Figure 20), basic information of a new model should be provided. Some fields, like the "Author Name" and the "Author's Email", will be included in model reports. The field "Notes" may be used to describe the product and development stage or status, useful during testing and measurement.

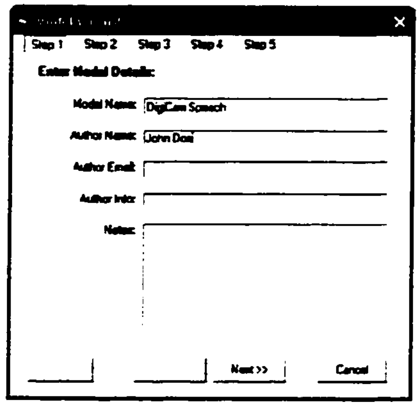

Figure 20 - Model Wizard - Step 1.

- Step 2 (Figure 21) will display a list of all factors that exist in the knowledge base. For example if a developing software should offer the "Effectiveness" and "Satisfaction" to end-users, these attributes will be included in the new model by selecting a checkbox in front of the attribute name.

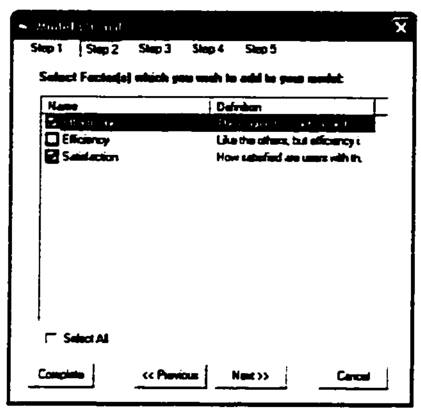

Figure 21 - Model Wizard - Step 2.

- In Step 3, only the criteria attributes that are related to selected factors, ("Effectiveness" and "Satisfaction" in the example provided in Figure 22), will be listed. With this narrowed list, a user cannot make mistake by including criteria attributes for the model that are not related to the selected factors. If a user wants to add a new factor or exclude the selected ones, the button "<< Previous" will bring back the Step 2 – the list of factors.

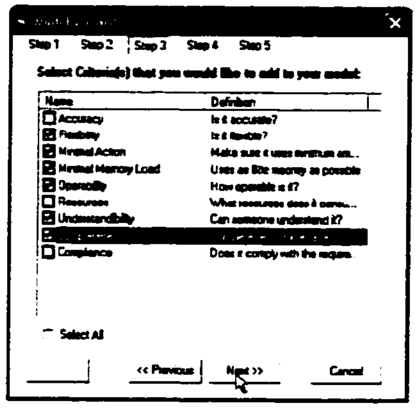

Figure 22 - Model Wizard - Step 3.

- Step 4 (Figure 23) will display a list of metric attributes. As in Step 3, the list will depend on the selection in the previous step. The "Select All" Model Wizard checkbox that automatically selects all listed attributes may be used to add all metrics that contribute to the selected criteria.

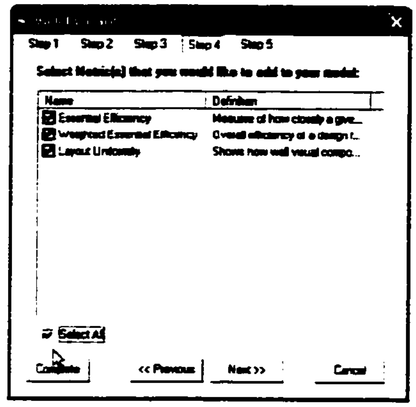

Figure 23 - Model Wizard - Step 4.

- Step 5 will display a list of all included data related to the selected metrics without a possibility to exclude them from the model, since data attributes related to metrics are used as variables for metric calculation and cannot be excluded from the model.

In each step window, there is a button "Complete" that makes the fast model building process even faster. By pressing this button in any step, all attributes from lower layers that are related with the selected attributes will be automatically included in the new model. For example, if a user presses the "Complete" button after selecting two factors (Step 2), the QUIM editor will add all criteria related to the two selected factors, then all metrics related to the added criteria and all data needed to measure the added metrics. Since all steps list only the attributes that exist in the knowledge map and such a list is narrowed down (based on the selected related attributes in the previous step), clarify importance of knowledge map for Model Wizard function.

After the model wizard is completed, the QUIM editor will generate a model tree in the

"Model" frame, based on the selected attributes.

4.4 Approach 2: Using the Model Editor

The "New Model" function should be used if a user wants to create a model that is based on an existing measurement framework, especially in case of MUSiC, which emerged from several usability models (ISO, SUMI). This function defines in the QUIM editor an empty instance of a model with its basic information (Figure 24).

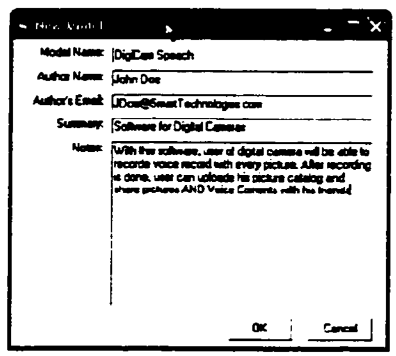

Figure 24 - New Model form

After the new model form completion, a new frame will appear in the upper part of a main application window (Figure 25). In addition to the familiar "Quality in Use Map" frame used for navigating through the QUIM knowledge base (left side), the frame "Model" will appear as the "explorer tree" representation of the new custom model (right side) in the same application window.

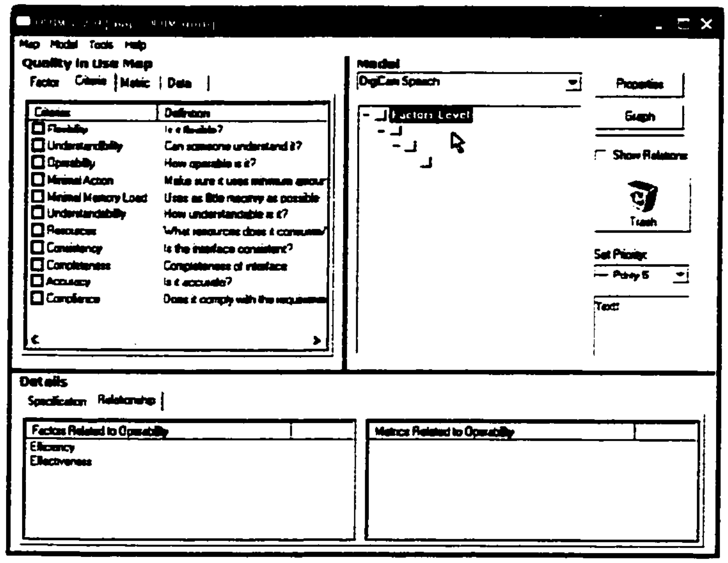

Figure 25 - Empty Model

The model frame contains a dropdown box with the name of a currently active model. The QUIM editor may contain up to 10 concurrent custom models and switching between them is performed by selecting model name from the dropdown box. There is an empty model tree under the model selection box that represents a hierarchy with the QUIM framework usability layers: factor, criteria, metric and data.

After the new model is created, any modification, like including or excluding usability attributes and defining their priorities, may be performed from the main window. The frame "Quality in Use Map" that is used for browsing the knowledge map is also used for adding or excluding attributes to the model using a checkbox located in front of the attribute name. When the attribute check box is selected, the attribute will be included to the custom model, and that attribute will appear in the custom model tree under the appropriate layer (factor, criteria, metric or data). By unselecting the check box, the attribute will be removed from the model. Another way to include an attribute to the model tree is to drag-and-drop the attribute to the model tree frame (press and hold the left mouse button at the attribute name, move the mouse cursor over the model frame and release the mouse button). Besides unselecting the attribute checkbox from the "Quality in Use Map", an attribute may be removed from the model by drag-and-drop the attribute name from the model tree to the "Trash" button, or by selecting the attribute name on the model tree and selecting the "Trash" button.

The QUIM editor uses black, blue and red color for an attribute name in the "Quality in Use Map" to express the status of that attribute according to an active model (Figure 26).

- The attribute name in black suggests that an attribute cannot be included in the model because it is not related with any selected (included) attribute from upper layer. An exception is with the factor attributes because the factor is the highest layer in the QUIM hierarchy.

- The attribute name in blue suggests an attribute is already added to the model. Consequently, the checkbox of that attribute should be selected. By unselecting its checkbox, that attribute will be removed from the model.

- The attribute name in red is a warning that, by default relationships in the QUIM knowledge base, it should be added to the model, because at least one of the related attributes from upper layer is already added in the model. The checkbox of that attribute can be selected, and by doing so, that attribute will be included in the model.

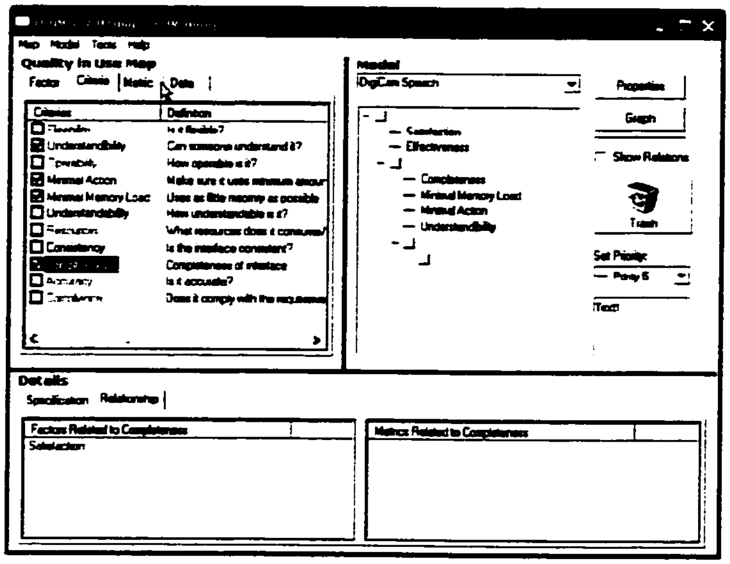

Figure 26 - Attributes colors

The best way to understand these simple rules is by analyzing the scenario presented in Figure 26. There are two factors and four criteria included in the model in the "Model" frame. All criteria attributes from knowledge base are listed in the "Quality in Use Map" frame. The criteria attributes "Completeness", "Minimal Memory Load", "Minimal Action" and "Understandability" are displayed in blue, as they are already in the model tree. The attributes "Flexibility" and "Operability" are displayed in red, as they are not included in the model, but their related factor attribute "Satisfaction" is. Similarly, the criteria attributes "Resources", "Accuracy" and "Compliance" are not included in the model, but their related factor "Effectives" is. The remaining criteria attributes that are displayed in black may also be included in the model. However, this is not advisable, because it does not correlate to relationship map defined in the QUIM editor.

4.5 Prioritizing the Attributes

Most educational material concerning usability and its applications is written for a specific context of use, like web sites. Nowadays, web sites are not just a few pages of information. They offer dynamic content, searches, database connectivity, and also e-commerce, and they can vary in usability attributes relevant to their model. Even the same function may have different usability expectation from the end-user. For example, the "Search" function should be fast and get first results as soon is possible in one case, but in another case the priority of the "Search" function may be to get more precise results independent of the required time.

Attributes included in the model may not be equally important for a future product. For example, a product should offer the "Satisfaction" and "Effectiveness" to user, but the "Satisfaction" is more important than the "Effectiveness". The QUIM editor is able to assign priorities to attributes. There is a scale of 9 levels of priority, used to precisely define a level of importance - 1 for the highest priority, 5 for no priority and 9 for the lowest priority. When an attribute is included into the model, its priority is 5.

To assign priority to an attribute, a user needs to select the attribute in the model tree and than from the "Set priority" dropdown box, select the desired priority level. Each attribute in the model tree contains a small icon in front of its name that represents the priority level. When the priority of an attribute is changed, its icon is updated according to the priority level, as shown (Figure 27). All model attributes in the model tree may be sorted by the attribute priority or by the attribute name from the menu "Model".

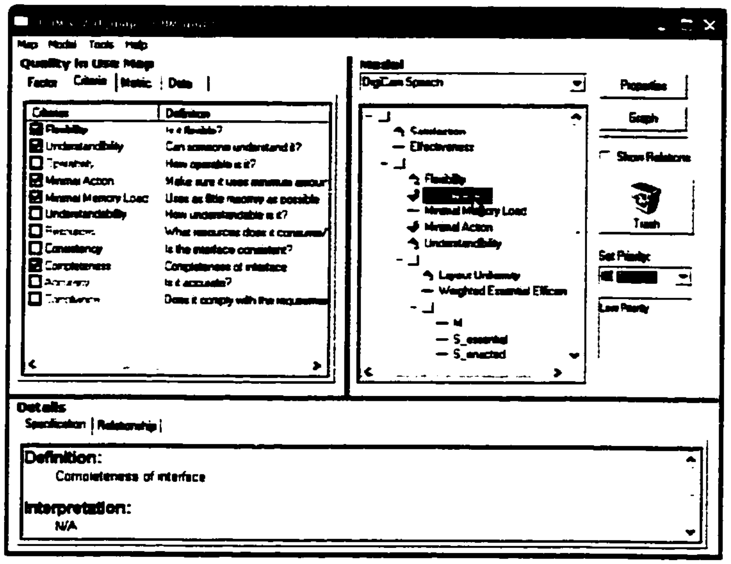

Figure 27 - Attributes Priority

The QUIM editor enables usability administrators to customize priorities and to provide users with description of each priority. This description is displayed in a small text box under the "Set Priority" dropdown box. Priority description may be customized from the preferences window (Figure 28) accessible from the menu "Model > Preferences".

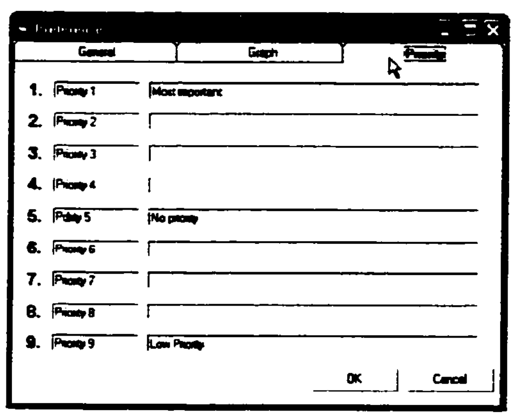

Figure 28 - Preferences Priority

These changes will be active immediately and present in every created or opened model by users' instance of the QUIM editor. Other tabs of the "Preferences" window are presented in the next chapters.

4.6 Relationships within a Model

Usability of the QUIM editor is achieved by integration of browsing and modeling function under one user interface. During modeling, users may refer to the attribute specification and relationships from the details frame by selecting any attribute from either the "Quality in Use Map" or the "Model" frame. For example, (Figure 29), to determine whether the blue colored attribute "Flexibility" in the "Quality in Use Map" is related to the factor "Satisfaction" or to the factor "Effectiveness", the attribute "Flexibility" should be selected. Subsequently, under the relationship tab, all related factors from the "Related Factors" list, will be displayed, and among them the factor "Effectiveness".

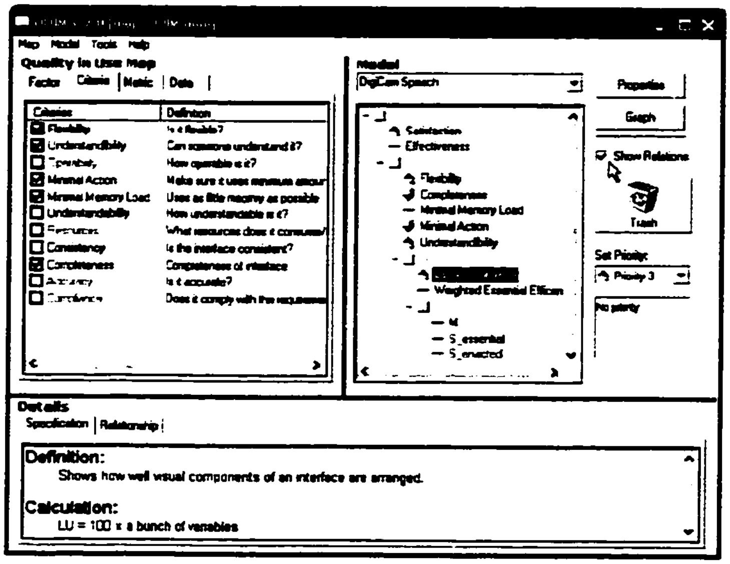

Figure 29 - relationships within Model

Identical approach may be used for the model tree if a user wants to find out, for example, to which of the included criteria is the metric "Layout Uniformity" related. However, a more efficient way to get this information is by selecting the checkbox "Show Relations". This will show in blue all attributes in all layers that are in any way related to the previously selected attribute. This will facilitate finding out how the model is affected by each attribute. This feature may be useful during the product development cycle, since it, for example, enables determination, whether it is more important to reduce screen complexity, database connectivity or servers efficiency.

4.7 Generating a Graph Model

For better understanding of relations, the QUIM editor provides a model visualization tool referred to as the "Model Graph". The graph generates the attributes map divided in four layers that contain all attributes and their relations, (Figure 30)

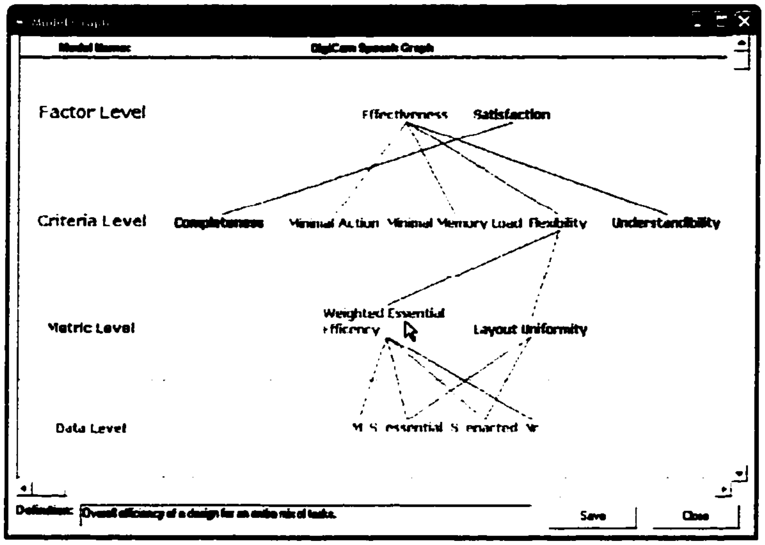

Figure 30 - Model Graph

Model graph window is interactive. When a user moves a mouse over any attribute in the text box, a definition of covered attributes will be displayed at the bottom of the window. Model graph can be saved as an image file in "gif", "jpeg" or "bmp" format, by pressing the button "Save".

Text of an attribute name may be displayed in three different colors, depending on the priority of the attribute. Colors, type, font size and other model graph customizations may be performed from the menu "Model > Preferences" under the tab "Graph" (Figure 31).

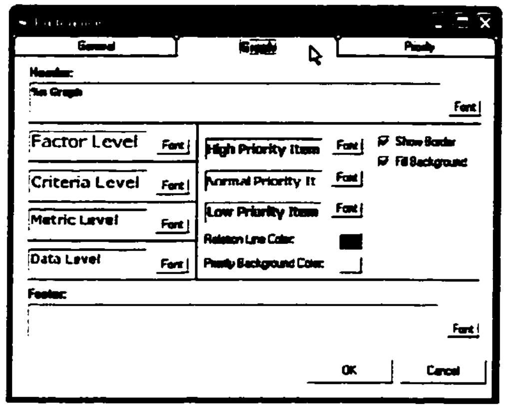

Figure 31 - Preferences Graph

Such a detailed customization is primarily motivated by a possible use of the model map graph outside of the QUIM editor. Since the model graph may be saved as an image file, it may be imported in any document, e-mailed to team members or even posted on the web site. The customization enables the model map graph to be adapted for environment where it will be used.

4.8 Generating a Usability Specification Portfolio

It is generally believed that it is hard to implement usability during a development cycle, since in most cases developers are not aware of usability models and methods, and usability experts often have difficulties communicating with developers. In the QUIM editor, this issue is addressed with the "Generate Model Report" function accessible from the menu "Model". This function generates an HTML based report containing 3 parts (Figure 32), based on a custom model. The first part provides details about the model and description notes, and identifies its author. The second part is a model map graph included as an image file. The third part is a list of all factors, criteria, metric and data included in the model. Priority, definition, interpretation, calculation (if applicable) and list of related attributes are provided for each attribute.

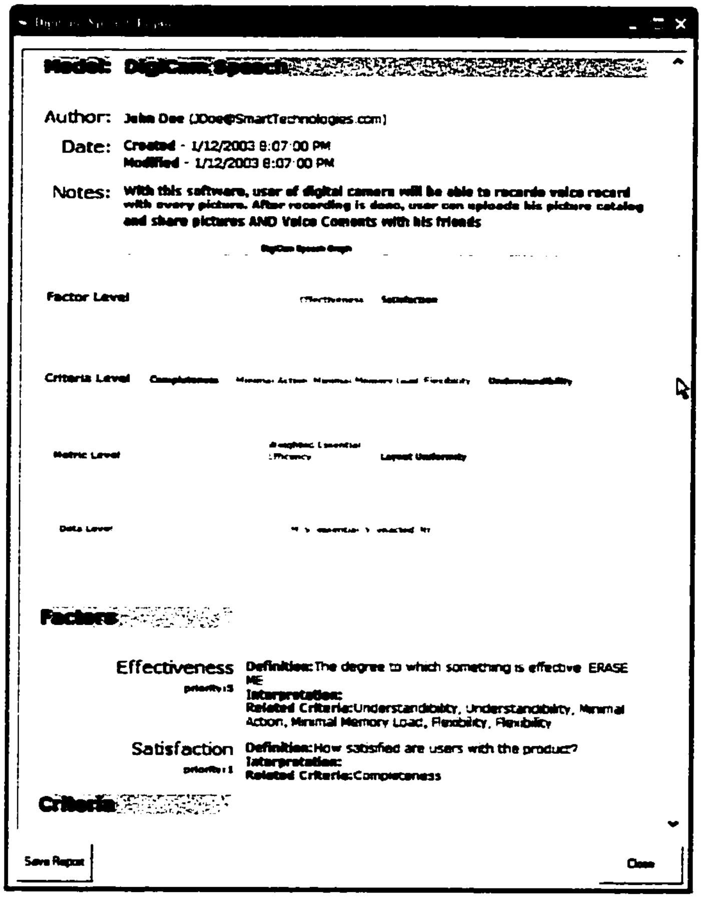

Figure 32 - Model Report

Model may be saved by pressing the button "Save Report". A user may enter the report name and location in the standard "file save" window. When saving is confirmed, the QUIM editor will generate two files in the user directory. One file is a report in "html" format and the other file is a graph image that has the same name as the report, but with "jpg" extension. The saved report may be loaded by any Internet browser. If a user wants to move, copy, or e-mail report to another user, or post it to web server, it is important to include both files. The model report enables easier collaboration with team members that are focused only on usability attributes related to a current project.

4.9 Generating Data Collection Forms

Jacob Nielsen believes that to improve a design, an insight is better than numbers [10]. He says usability metrics let one track progress between releases and applying metrics can cost four times as much as conducting qualitative studies. However, we believe that measuring usability by involving a real user is an expensive method and alternatives for predicting usability during design process are needed. It is generally accepted that the design of a web site should take into account the established guidelines for web writing style, navigation, site structure and page design. The problem with guidelines is that they provide subjective information to the developer, for example they suggest to "minimize the need for scrolling" [39]. Only an experienced web site designer may be able to envision such a guidelines recommendation, but objective and measurable attributes may be used by any developer.

There are many tools and methods that may be used during the usability evaluation process in order to achieve good product usability. Many of these methods and tools are presented within the attributes specification, with a purpose to help a user in inquiry, collection and measurement process. As a support for this process, a function that generates the HTML data collection form based on custom model attributes and relationships within a model is developed in the QUIM editor. When a model is defined and ready for usability testing, system will generate a new data collection form by selecting the "Generate Test form" from the menu "Model". Figure 33 summarizes the role of QUIM in collecting and using the measurement knowledge as well as the bridging of measurement theory and practices.

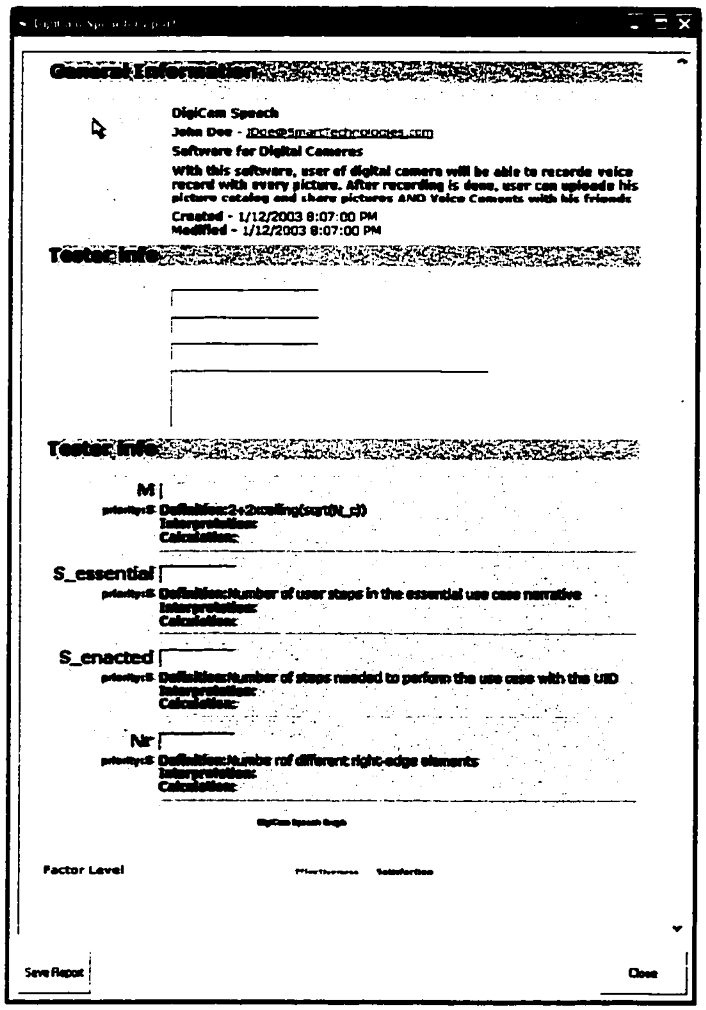

Figure 33 - Model Testing

The data collection form consists of four parts:

- "General Information", which provides details about model defined by the user when model is created;
- "Tester Info" section, which contains fields related to the person who will provide values for tested attributes;
- "Test attributes" section, which lists all data attributes included in the Model. For each attribute, the priority within the model and the Definition, Interpretation and Calculation defined in the QUIM knowledge base are presented. There is a textbox field near the attribute name, where a user may type numeric value collected by the usability tools or methods;
- Graphical representation of the Model map included as an image file, also used for the "Model Report".

Similar to the "Model Report", the content of a new window may be saved on a hard drive as an HTML file. When the form is saved, it may be moved, copied, e-mailed or posted to any remote location where usability measurement will take place.

5 QUIM Editor: Design Challenges and Rationale

The QUIM editor that includes all features presented in chapters 2, 3 and 4 is a result of one and a half year of research and iterative development. During that period, we developed several versions (mainly 3) and we conducted several tests with potential users of the tool, including graduate students from COMP 675 course at Concordia University and researchers from the human-centered software engineering group as well as from elsewhere. The tool has been presented at many conferences and industry forums (Donyaee, 2001; Seffah, 2002). Other research groups have also tried to use it to support their research investigations on usability measurement or as an educational tool.

In this chapter, we summarize the development methodology and milestones that lead to the QUIM editor version 2.0 presented in this thesis. We also highlight some of the lessons learned that can be useful for researchers and practitioners in the field of usability measurement.

5.1 QUIM Editor Compared to Existing Tools

It is difficult to compare the QUIM editor to other tools since there are no similar tools in the area of usability measurement. The QUIM editor offers support in the area of education, modeling and measuring.

5.1.1 ZD-MIS

As a tool for understanding and mastering usability metrics and measurement, the QUIM editor can be compared to a database called ZD-MIS (Zuse / Drabe Measure-Information-System). This database has influenced the development of the QUIM architecture and its conceptual design, although it is not dedicated to the usability attributes measurement.

ZD-MIS is used to quantify software quality attributes. The major goal of the ZD-MIS is to provide a comprehensive framework for mastering the software measurement.

ZD-MIS contains a large database with more than 1500 software measures of all kinds in a uniform description. Every measure is described by more than a dozen criteria like: definition of the measure, captured quality attributes, type of measure, assumed empirical axioms, assumed scale types, applications in the software life-cycle and class of related measures (product, resource or process measure). The database contains also more than 180 object-oriented software measures. The measures can be selected by more than a dozen criteria. This database is designed with a purpose to help scientists, students and practitioners to select measures for their needs. It also contains more than 1600 references to literature and glossary of more than 600 terms helping a user to get a better understanding of the software measurement area [24]. The field of software measurement is much larger and mature compared to the usability measurement, which makes it impossible to compare the QUIM knowledge map with this database. But, all metrics included in the database are described in a uniform manner as we did in the QUIM database.

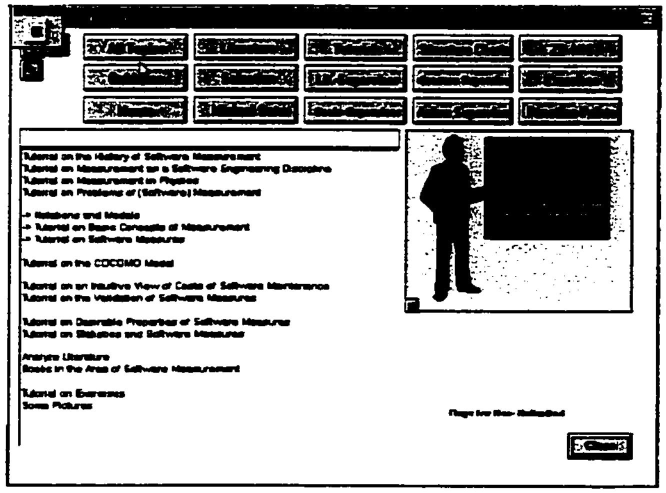

Figure 34 – ZD-MIS tool (adapted from [24])

ZD-MIS also provides many guidelines on how to make a quantitative assessment of software quality using the software measures. It provides detailed information about the correct use of software measures. This information is not tailored for specific software or model, as we did in the QUIM editor. ZD-MIS demonstrated its usefulness as a tool for educating students and scientists in the filed of software measurement. Our impression is that people sometimes think that measurement is a simple procedure. Many scientists point out that the current state of the software measures is not satisfying, and tools for education are needed.

ZD-MIS provides, among others, the following information that is extremely useful in an educational setting:

- Benefits of software measurement;
- Comparing measurement in physics and software engineering;

- Illustration of the definitions and properties of many measures;
- The Function-Point method;
- The foundations of wholeness, counting, etc.;
- Brief introduction in the measurement theory as a foundation for software measurement;
- Learning the meaning of axiom systems by a dialog;
- Learning what type of attributes is measurable;
- Learning the meaning of scales by a dialog;
- Demonstration of models of complexity, understandability, maintainability, etc. behind the software measures;
- Demonstration of the models behind the COCOMO model and the Function-Point Method;
- Understanding the meaning of scale types, meaningfulness and empirical conditions for the use in practice;
- Foundations of validation of software measures and prediction models;
- Specification of measures;
- How to setup a measurement program;

- Application of many measures to structure charts in a dialog [24].

5.1.2 BORE

As a tool for managing usability measurement knowledge, the QUIM editor is completely new in the usability community because all existing usability models are static. They include a predefined list of usability attributes, and the practice of using them is hard to capture and include. Therefore, creating the tailored models is not an easy task.

BORE (Building an Organizational Repository of Experiences) is a tool designed to further explore and refine the requirements for tools supporting practices or experience-based approaches. Its purpose is to be used as a proof-of-concept prototype to articulate organizational learning and experience-based software development techniques. Again here, part of QUIM editor can be compared with BORE. The BORE tool creates a framework for the experience factory by combining a work breakdown structure with repository tools for designing software process methodologies, and repository technology for capturing and applying knowledge artifacts. The BORE tool and methodology extend the experience factory concept through rule-based process tailoring, support for process modeling and enactment, and case-based organizational learning facilities [40].

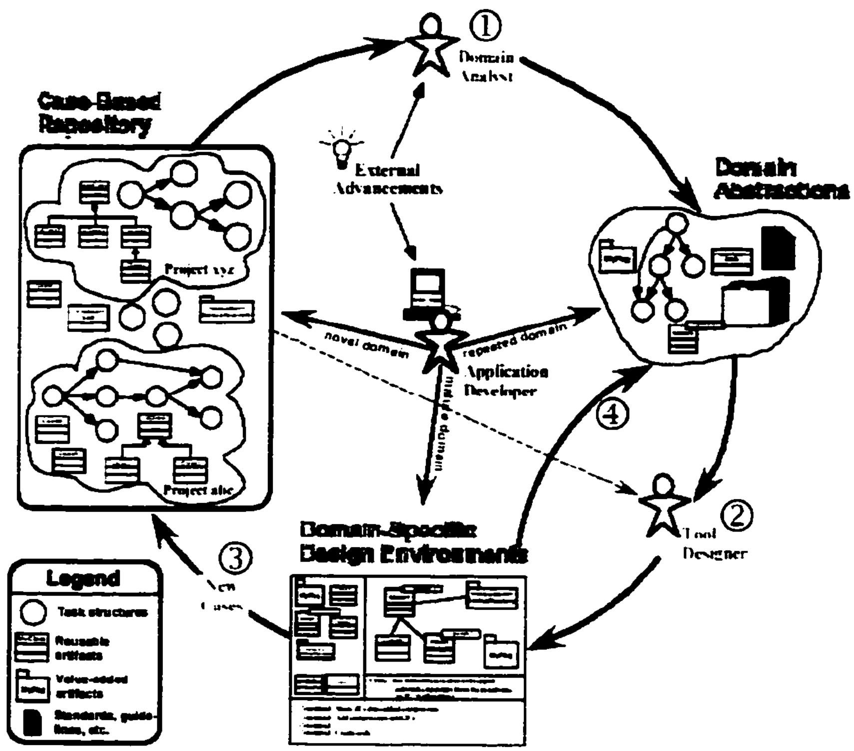

Figure 35 - BORE tool (adapted from [40])

The domain lifecycle (shown in Figure 35) is a representation of two simultaneous phenomena. The outside circle shows how knowledge evolves in an experience-based environment. The inside of the circle represents a model of the level of support software developers have at their disposal during the development process. It depicts domain analysis and the design of development tools as key experience factory concepts that contribute to the evolution of knowledge, while providing essential tools for the software developer.

The domain analyst evaluates systems (1) begins to create a core set of best practices, and catalogs the variations between the different development contexts and the requirements these systems need. This knowledge is formulated into reusable tools, frameworks, or other reusable artifacts that can be reused, extended, and instantiated by a tool designer (2) in domain-specific tools. In the QUIM editor, the domain oriented knowledge is expressed in the knowledge map.

Although the BORE domain-oriented tools can provide substantial coverage of a domain, it does cover entire project domains. The rest of the project artifacts needed to complete the project would be created by the project team, creating situation-specific cases (3) or by the Experience Factory, which would create new reusable artifacts (4) and design environment (tool) extensions (2) to meet the project's needs. For example, even with the creation of knowledge environments, knowledge creation is still an essential part of the software development, and each project will extend the case-based repository in some way to fill in what the environment lacks [40]. Similar process is used in the QUIM editor, where a model for specific domain or context of use can be extracted from the knowledge map. The map may be extended and adjusted according to empirical results obtained from best practices.

5.1.3 DRUM

In the area of usability measurement, most of the existing tools are designed to measure specific usability factors that have predefined meanings and interpretations. For example, SUMI questionnaire, presented in Chapter 3.2.4, can be used to measure the user satisfaction. There are also tools designed to support an explicit usability model. For example, DRUM is commercial software that provides support for extracting a list of behavioral patterns from video observational data, which can be analyzed using the MUSiC method, described in Chapter 4.2. The evaluation sessions are video-recorded and then analyzed how successfully people achieve the task goals while using a system. DRUM creates a log of the significant events it has observed on the video from the test. It also assists the generation of diagnostic feedback concerning the usability problems with the system. The DRUM tool provides automated calculation from the logs that gives the analysts the following metrics: Task Time; Snag, Search and Help Times; Efficiency; Relative User Efficiency and Productive Period [36].

The QUIM editor is not a substitute for tools like SUMI and DRUM. It can complement them by encapsulating the best practices on usability measurement. The Test forms generated by the QUIM editor can be used as guidelines for measurement and analysis using SUMI, DRUM, and MUSiC.

5.2 Milestones in QUIM Editor Development

During his master thesis work, Mohammad Donyaee introduced the idea of the QUIM editor. With 3 other graduate students, I developed the first prototype version of the QUIM editor. This first version was roughly a horizontal and low-fidelity prototype. It described the look and feel of the system without implementing any functionality. This prototype was useful for gathering more requirements as well as difficulties for understanding the usability measurement process (Figure 36).

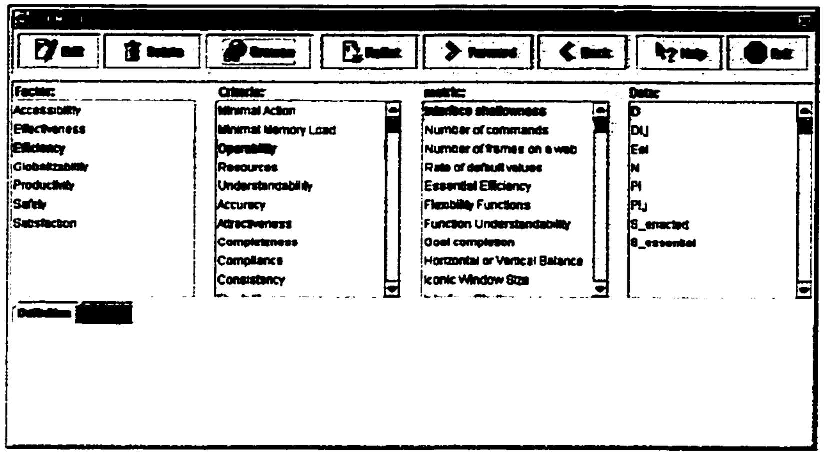

Figure 36 - Donyaee prototype

Several researchers used this first prototype. It highlights the need to automate or support the development of custom model using the knowledge maps, which include a large set of metrics and the related measurement practices. Furthermore, the proposed user interface model, which distinguishes four list boxes for the QUIM attributes, left no space for adding new functionality. Therefore, during the COM 675 graduate course, we conducted a participatory design workshop with 9 groups of students. We designed a new prototype of the QUIM editor user interface based on newly recognized functionalities. Figure 37 presents the result of this prototyping session.

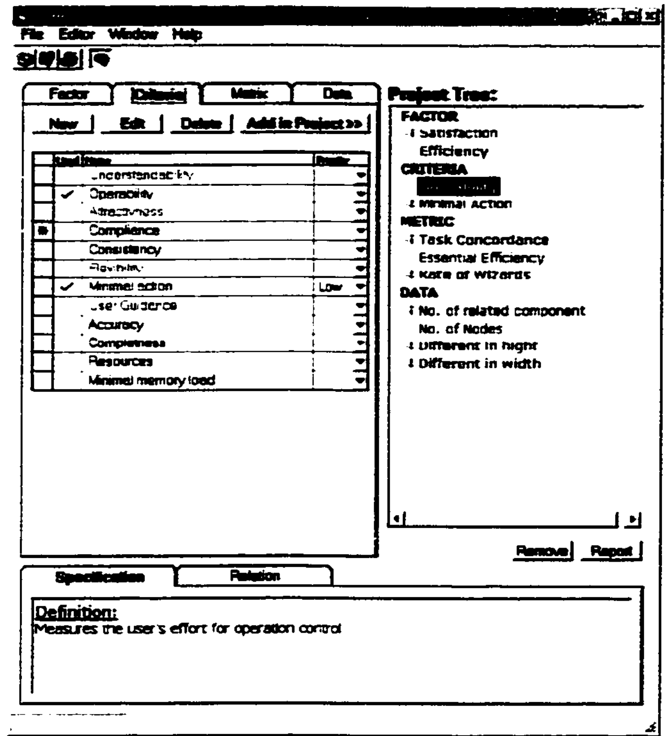

Figure 37 - QUIM editor version 1.0

A radical change of the interface was introduced in this prototype. The four usability layers of knowledge map were presented as "tabs" and on the right side the model (formally project) can be displayed as a tree. A new database structure that supports custom models development and priorities was also designed. This first fully functional version of the QUIM editor version 1.0 was introduced to our research team for tests.

Our research team used the QUIM editor for knowledge map management. We were able to conduct interviews that helped us to identify requirements for a new version of the QUIM editor v.1.4 (Figure 38). The following major improvements have been proposed by researchers:

- Interactive graphical model representation;
- Loading and switching between multiple models;
- HTML report based on custom model;
- Wizard for helping novice designers to quickly build a custom model.

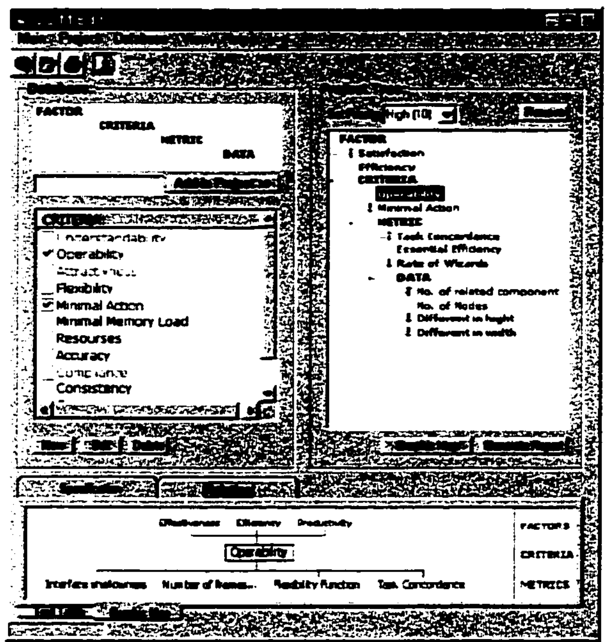

Figure 38 - QUIM editor version 1.4

The QUIM editor version 1.4 fulfilled all the requirement of a usability expert's community. However, this application seemed too complicated from the point of view of non-expert users. Users were confused when displaying the map and the model tree under one interface since they are mostly interested in the browsing and learning aspect of a tool. In the QUIM editor version 2.0 (the final version), presented in this thesis, the user interface with model related functionality is hidden, until the user starts creating a new model. Several new functionalities were also added, such as:

- HTML Test form based on custom model;
- Model image imbedded in HTML reports;
- Protection of Knowledge Map database;
- Support multiple knowledge maps;
- Custom definition of attributes priorities.

For facilitating the testing of the QUIM editor, a web site was created to provide background information about the QUIM project and an interactive demo version of the QUIM editor was also developed.

6 Conclusion and Future Investigation

6.1 QUIM Editor Advantages

The QUIM editor described in this thesis promotes greater integration of the crucial issue of usability in the overall software engineering lifecycle, as well as it helps in the establishment of the theory for usability/quality in use measurement. Since usability is still an evolving concept, the QUIM editor was designed to support rapid changes and updates in measurement models, and to make them immediately available to a wide audience. Usability measurement models are more accessible and usable for non-expert users. The multi-perspective browsing features facilitate the understanding of the complex usability attributes such as trustfulness and accessibility and how they can be quantified and measured. For usability evaluators, the QUIM editor provides knowledge of gathering and interpreting usability data and metrics and mapping their numerical values to empirical statements. QUIM helps also to identify predictive metrics that allow Human Computer Interaction (HCI) group and usability experts to extend the usability knowledge base by adding their own factors, criteria, and metrics to it. Existing usability models are fixed and immutable. Since definitions and approaches to usability can vary depending on the context of use, the QUIM editor makes it possible to adapt the QUIM knowledge map to a custom model. Customizable models may help developers who are not usability experts to create more usable systems. The QUIM editor includes only relevant factors, criteria, metrics and their internal relationships, priorities and dependencies, and related measurement methods. Unlike models that require particular expertise to apply them, the QUIM models are applicable for anyone involved in the development process.

The QUIM editor has been implemented as a standalone application for Microsoft Windows Platforms and uses Microsoft Access Database as the QUIM database file format. The QUIM editor is distributed as self extractable installation package. For more technical information on the QUIM editor, please refer to the QUIM editor technical document [41].

6.2 Future work

Further development of the QUIM editor includes a validation of the metrics. A metric is valid if it accurately characterizes the criterion it claims to measure. This requires the comparison of performance models against external data. It is, thus, necessary to conduct empirical studies about the potential relationships among the QUIM attributes. There are several methods for validating software metrics [30] [36]. These methods may be used to validate quality in use metrics.

When the QUIM knowledge map becomes more stable and mature, it will be much easier to identity future development. Future investigations should include the usability measurement based on specific models and post analysis of collected data, as collection, measurement, interpretation and calculation of usability attributes depends on applicable methods, development stage, participants, time, environment, etc. Thus, it is necessary to develop a centralized knowledge repository accessible via the Web and a Database server (QUIM server). The QUIM server should offer the following:

- Information portal;
- Metric and data collection;
- Measurement, data analysis and recommendations.

6.2.1 Information Portal

As an information portal, the QUIM server should contain information about organizations, companies or individuals that are able to contribute to the QUIM framework development. The collaboration is an important factor in the promotion and recognition of the QUIM framework. The QUIM server may keep worldwide community up to date, by posting news, articles, latest versions of measurement models, and updated versions of the QUIM knowledge base. In addition to this, registered users, in particular HCI experts, should be allowed to post their own versions of the QUIM knowledge base and make it available for other users. Users should constantly follow future framework development. Such collaboration is necessary among end users interested only in usability, managers using the QUIM editor in development process, and HCI experts doing research on usability measurement.

6.2.2 Metric and Data Collection

The QUIM server may also offer support to testers during usability testing. Beside HTML Reports that contain specifications of all attributes included in a custom QUIM model, the QUIM editor can also generate data collection forms, which can be distributed over the Internet. The data collection forms are in HTML language and can be opened by any Internet browser. Collection form contains the specification and values input text field for each data attributes, and input fields used to describe testing site, environment and tester personal information. Entered values could be submitted in a future version of data collection form, to remote QUIM server by adding "submit" button at the end of the form. Using this method, each data collection form, distributed to one or multiple Internet-connected remote testing sites, could be used to submit data values to the QUIM server. The QUIM server would receive two groups of information with submitted values. The first group of information would be the same for each instance of the generated data collection form and there would be the exact number of received values. The second group of information could represent the data (or metric) values. As every model may have different amount of data and metric attributes needed to measure usability, the number of data or metric values to be received is unknown. However, it is known that for each instance of the data collection form, independent of a number of times a form is submitted, the same number of all entered values will be received. After the testing, usability managers would be able to login to the QUIM server over the Internet and locate the exact version of the model for which they distribute the data collection form, based on the first group of received data collection form values. These tested model versions should represent different stages in the development process. Based on users' feedback and previous results, the model should be constantly adapted and tuned during the development cycle. For each test version of the model, managers would be able to review data collected from users and proceed with interpretation, evaluation and measurement.

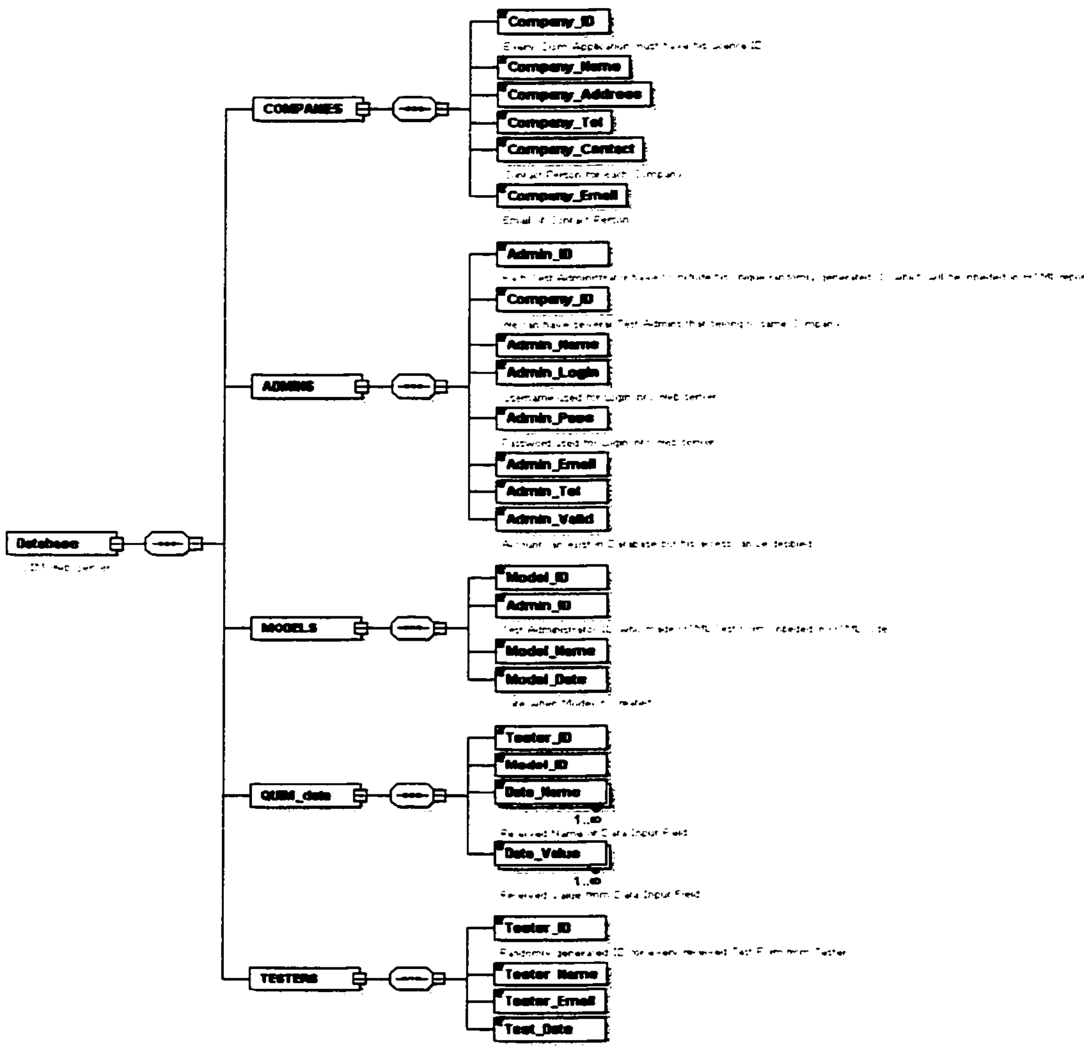

Figure 39 - QUIM server database diagram

Figure 39 suggests a database structure of the QUIM web server for collecting values from submitted test forms. The table "Companies" may hold information about the QUIM framework members and organizations, like universities and companies that use the QUIM server for testing. Each organization should contain usability experts, administrators or managers who are responsible for creating usability models and generating test forms from these models at different stages of the product/system development lifecycle. After submitting the test forms, a new instance of a model version should be created in the table "Models". The collected attributes values should be stored in the table "QUIM_data". Information about tester should be saved in the table "Testers". This suggested structure is a starting point for more detailed investigations.

6.2.3 Measurement, Data Analysis and Recommendation

The QUIM server could also benefit measurement prediction, analysis and recommendations based on collected values from the data collection forms. Development of such a platform will need extension of the QUIM framework by including calculation of metric attributes, followed by mapping of these values to criteria and factors using the relationships defined in the "Quality in Use Integrated Map". As a result, specific values should be received, leading to recommendation based on the context of use.

Another improvement of the QUIM editor would consist of adding measurement scales. During the knowledge map acquisition, usability experts should be able to enter empirical values (choices offered to users) that are mapped to numerical representations based on the measurement scales, so that further calculations can be done. In the Test form for such attributes, tester should be provided with a dropdown box with selectable options instead of a text box.

In the area of usability prediction, the QUIM server could offer further analysis of the collected data. Once calculation is automated, with "*what if*" scenarios, usability managers could discover attributes whose small improvements may reflect significant improvements of the product usability as a whole.

References

[1] Bevan N., Macleod M., "Usability measurement in context", National Physical Laboratory, Teddington, Middlesex, UK, 1994

[2] ISO/DIS 9241-11: "Guidance on usability. Ergonomic requirements for office work with visual display terminals (VDT)s", 1996.

[3] ISO 9126: "Software product evaluation - Quality characteristics and guidelines for their use", 1991.

[4] ISO 8402: Quality management and quality assurance – Vocabulary, 1994.

[5] Shackel, B., "Usability - Context, framework, definition, design and evaluation. In B. Shackel and S. Richardson (eds.) Human Factors for Informatics Usability". Cambridge University Press, UK, 1991.

[6] Mohammad K. Donyaee., "Towards an Integrated Model for Specifying and Measuring Quality in Use", Master thesis, Concordia university, Canada, 2000.

[7] Garvin, D., "What does quality Really mean?", Sloan Management Review, p. 25-45, Fall 1984.

[8] Bevan, N., "Measuring Usability as quality of use", Software Quality Journal, 4(2), p. 115-30, 1995.

[9] European Community: "Council Directive 90/270/EEC on the minimum safety and health requirements for work with display screen equipment", Official Journal, no. L 156, 1990.

[10] Nielsen J., "Usability Engineering", Academic Press, London, UK, 1993.

[11] Preece J., Rogers Y., Sharp H., Benyon D., Holland S., Carey T., "Human Computer Interaction", Addison-Wesley, 1994.

[12] Retalis S., Psaromiligkos Y., "The experience of hands-on evaluation of educational software", Proc. 8th Panhellenic Conference on Informatics, Nicosia, Cyprus, 2001.

[13] Martijn van Welie, Gerrit C. van der Veer, Anton Eliëns, "Breaking down Usability", Edinburgh, Scotland, 1999.

[14] Shneiderman B., "Designing the User Interface", Addison-Wesley Publishing Company, USA, 1998.

[15] Ahmed S., Keceli N., Mohammad K. Donyaee, "QUIM: A Framework for Quantifying Usability Metrics in Software Quality Models", Concordia University, Canada, 2000.

[16] Keceli N., Abran A., "Analyzing, Measuring and Assessing Software Quality in a Logic Based Graphical Model", 4th International Conference on Quality and Dependability – QUALITA, France, 2001.

[17] Norman E. Fenton, Shari L. Pfleeger, "Software Metrics A Rigorous and Practical Approach", International Thonson, London, UK, 1996.

[18] Lisch R. & Kriz J., "Foundations and models of content analysis", Hamburg, Germany, 1978.

[19] IEEE Standard for a Software Quality Metrics Methodology. IEEE std. 1061-1992, New York, USA, 1993.

[20] Larry C., Lucy L., "Software for Use", Addison-Wesley, New York, USA, 1999.

[21] Kirakowski J., M. Corbett, "Effective Methodology for the Study of HCI", Amseterdam, North-Holland, 1990.

[22] Lewis, J. R., "IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use", International Journal of Human-Computer Interaction, p. 7, 1, 57-78, 1995.

[23] Corbett M., Macleod M., Kelly M., "Quantitative usability evaluation - the ESPRIT MUSiC project", In Smith, M.J. and Salvendy, G. (eds.). Human computer interaction: applications and case studies. Advances in human factors/ergonomics, 19 A, Elsevier, Amsterdam, 313-318., 1993.

[24] Horst Zuse, "A Frameworks of Software Measurement", Berlin, Germany, 1997.

[25] Nielsen, J., Mack, R. L., Usability inspection methods, New York, USA, 1994.

[26] Karat C.-M., Campbell R. L., Fiegel T., "Comparison of empirical testing and walkthrough methods in user interface evaluation. Proceedings of CHI 92", p. 397-404. New York, USA, 1992.

[27] Beyer, Hugh, and Holtzblatt, Karen, "Contextual Design : A Customer-Centered Approach to Systems Designs", Morgan Kaufman Publishers, San Francisco, USA, 1997.

[28] Caplan, S., "Using focus groups methodology for ergonomic design", Ergonomics 33(5), p. 527-533, UK, 1990.

[29] Salant, Priscilla, Dillman, Don A., "How to Conduct Your Own Survey", John Wiley & Sons, New York, NY, 1994.

[30] Kirakowski J., Corbett M., "SUMI: The Software Usability Measurement Inventory", British Journal of Educational Technology, 24(3), p. 210-212, 1993.

[31] John B. E., Packer H., "Learning and using the cognitive walkthrough method: A case study approach", Proceedings of CHI 95, 429-436. New York, NY, 1995.

[32] A. Seffah and M. Donyaee, "Theorical Framework for Quality in Use", Department of Computer Science, Concordia University, Canada, 2001.

[33] Michael K. Daskalantonakis, "A practical view of software measurement and implementaion experiences within Motorola", The Institute of Electrical and Electronics Engineers (IEEE) Transaction on Software Engineering, Vol 18, 1992.

[34] SAS Institute, http://www.sas.com/

[35] SPSS for Windows (Version 10.0). [Computer Software], http://www.spss.com

[36] Macleod M., Bowden R., Bevan N., "The MUSiC Performance Measurement Method", Submitted for publication.

[37] Rengger R., Macleod M., Bowden R., Drynan A., Blayney, M., "MUSiC Performance Measurement Handbook", V2. NPL, Teddington, UK, 1993.

[38] Timoshenko and Young, "Engineering Mechanics – Statics", McGraw-Hill, 1937

[39] Bevan Nigel, "Usability issues in web site design, Proceeding of UPA'98", Washington, USA, 1998.

[40] Walz D.B., Elam J.J., and Curtis B., "Inside A Software Design Team: Knowledge Acquisition, Sharing, and Integration", Communications of the ACM, 36 (10). pp. 62-77., 1993

[41] Jovan S., "QUIM editor: Technical document", Concordia University, 2003

Appendix

Knowledge Map Factors attributes:

Efficiency: capability of the product to enable users to expend appropriate amounts of resources in relation to the effectiveness achieved in a specified context of use.

Effectiveness: The capability of the software product to enable users to achieve specified goals with accuracy and completeness in a specified context of use

Satisfaction: Satisfaction describes the subjective response of user while using a software product in a specified context of use.

Productivity: The level of effectiveness achieved in relation to the resources (i.e. time to complete tasks, user efforts, materials or financial cost of usage) consumed by the users and the system while using a software product in a specified context of use.

Learnability: In all, it is the ease with which a user can master the required features for achieving a certain goal in a certain context of use.

Safety: The degree to which a software product limits the risk of harm to people in a specified context of use.

Trustfulness: Trustfulness is the degree of faithfulness a software product offers to its users in a certain context of use.

Accessibility: The capability of a software product to be used by permanently or temporarily disabled persons (i.e. vision, hearing, motor, cognitive and language impairment) in a certain context of use

Universality: The degree to which a software product accommodates diversity of user cultural/social archetypes in various context of use

Usefulness: The degree to which a software product actually helps to solve user's real practical problems in a certain context of use

Knowledge Map Criteria attributes:

Time behavior: The capability of the software product to provide appropriate task time when performing its function, under stated conditions.

Resource Utilization: The capability of the software product to use appropriate amounts and types of resources when the software performs its function under stated conditions.

Attractiveness: The capability of the software product to be attractive to the user. This refers to attributes of the software intended to make the software more attractive to the user, such as the use of color and the nature of the graphical design.

Likeability: It refers to the user's perceptions, feelings, and opinions of the product.

Flexibility: In terms of user interface, flexibility involves providing the opportunity for users to tailor interactive elements to fit their personal preferences.

Minimal Action: It is the capability of a software product to help users achieve their tasks in minimal steps.

Minimal Memory Load: It is the extent to which user needs to keep minimal amount of information in mind to achieve a specified task.

Operability: It is a measure of users' effort for operation and operation control of a software product.

User Guidance: The extent to which the interface provide context-sensitive user help facilities and meaningful feedback when errors occur.

Understandability: It is the capability of a software product to enable the user to understand whether the software is suitable, and how it can be used, for particular tasks and conditions of use.

Consistency: It is the degree of uniformity or mutual agreement of the components of a user interface design of a software product and with interface designs of other software products familiar to the user or correspondence with the real world metaphors beyond computer domain.

Self-Descriptiveness: It is the ability of the software product to introduce itself and its purpose, as well as provide users clear and concise assists for its correct operation.

Feedback: It is the response of the software product to user actions.

Accuracy: The capability of the software product to provide the right or agreed results or effects.

Completeness: It is the extent to which the user can complete a specified task.

Fault Tolerance: The capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.

Resource Safety: The capability of software product to handle resources properly without any hazard or mishap to people in its context of use.

Readability: Readability is the ease with which visual content can be understood or it is the degree to which the meaning of text is accessible.

Controllability: It is the degree to which the users feel that they are in charge of the software product.

Navigability: It is the degree to which a user can move around in the application.

Simplicity: It means to eliminate the extraneous and enhance the user experience, while at the same time not sacrificing the quantity of information.

Privacy: It is the ability of the software product to handle user's personal information and disclosing it to the third party only with the user's consent.

Security: The capability of the software product to protect information and data so that unauthorized persons or systems cannot read or modify them and authorized persons or systems are not denied access to them.

Insurance: Insurance is the liability of the software product vendors in case of fraudulent use of the users personal information; or the users change their minds while shopping online in eCommerce sites.

Familiarity: In terms of user interface, it is the degree to which a user recognizes user interface components and views their interaction as natural; the similarity of the interface to concrete objects the user has interacted with in the past.

Loading Time: It is the time it takes for a web page to load i.e. how fast a web site responds to a user.

Presentation: It is a measure of the degree of professionalism of the user interface design of a software product.

Knowledge Map Factors – Criteria relations

FACTORS / CRITERIAS	Efficiency	Effectivenes	Satisfaction	Productivity	Learnability	Safety	Trustfulness	Accessibility	Universality	Usefulness	Relative Importance Of Criteria
Time Behavior	•			•							2/10
Resource Utilization	•			•						•	3/10
Attractiveness			•						•		2/10
Likeability			•								1/10
Flexibility		•	•					•	•	•	5/10
Minimal Action	•		•		•			•			4/10
Minimal Memory Load	•		•		•			•	•	•	6/10
Operability	•		•				•	•		•	5/10
User Guidance			•		•			•	•		4/10
Understandability	•		•		•			•	•		5/10
Consistency		•			•	•		•	•		5/10
Self-Descriptiveness					•		•	•	•		4/10
Feedback	•	•							•	•	4/10
Accuracy		•				•				•	3/10
Completeness		•				•					2/10
Fault- Tolerance						•	•			•	3/10
Resource Safety						•					1/10
Readability								•	•		2/10
Controllability							•	•	•	•	4/10
Navigability	•	•					•	•	•		5/10
Simplicity					•			•	•		3/10
Privacy							•		•	•	3/10
Security						•	•			•	3/10
Insurance						•	•				2/10
Familiarity					•		•				2/10
Loading Time	•			•					•	•	4/10
Presentation							•	•	•	•	4/10
Relative Importance Of Factors	9/27	6/27	8/27	3/27	8/27	7/27	10/27	13/27	15/27	12/27	

Table 1 - QUIM map Factor / Criteria relationships