Chapter – 4

Concurrent Usability Evaluation and Design of a Software Component

4.1 Overview

In this chapter an attempt is made to address the *usability* issues of software component that can be extended to component based software system. The developed approach provides a framework, utilizing digraph and matrix approach, with which component and component based software system can be assessed, evaluated and designed as per *usability* point of view. The chapter is organized as follows. Section 4.2 provides a description on *usability* issues in relation to software component. In section 4.3, based upon concurrent approach, digraph modeling of sub-characteristics and associated attributes including interactive complexity are discussed. Section 4.4 focuses on the development of one-to-one representation of digraph model for computer processing. In section 4.5, evaluation of component based upon characteristic expression of *usability* is discussed. Section 4.6 describes procedure for developing *usability* index for component. In section 4.7, validation of the developed approach utilizing case study is demonstrated. Section 4.8 describes the usefulness of the developed methodology. Finally section 4.9 provides concluding remarks of the chapter.

4.2 Introduction

The *usability* characteristic represents the significant difference in meaning between traditional quality models and component quality model. The difference lies in the fact that in the component quality model the users are considered primarily system developers, who handle the integration of the components to their systems. Measuring and evaluating software component is an important step to avoid any delays in developing CBSS. This will also help in making flexible and maintainable CBSS. Most of the available literature considers *usability* sub-characteristics as independent entities affecting the software component's *usability*. But the extent to which one sub-characteristic is present may affect the other sub-characteristic. The estimate of the net effect of these interacting sub-characteristics is responsible for the success of the component *usability* philosophy. Quantification of component sub-characteristics and their interactions concurrently will lead to estimating the net effect of the component's *usability* in terms of a single numerical index. Thus, an attempt is made to consider component's *usability* sub-characteristics and interaction amongst them concurrently in a unified manner. The approach mentioned in chapter 2 is extended to compute *usability* index of a software component.

ISO 9126 quality model is chosen to identify and develop *usability* sub-characteristics of a software component. This model is well accepted worldwide but is general thus to be tailored or customized for component oriented domain as per *usability* point of view. Although most of the sub-characteristics remain the same in terms of naming (only *understandability* is renamed to *help* mechanisms). *Help mechanisms* need to target different people. Some people look for "*how to*" administer component while some are interested in understanding "*interfaces"* support. In component domain end users are the system developers and not the ones who interact with complete system thus sub-characteristics *attractiveness* has been removed from the sub-characteristics list of *usability*. Due to lack of standards, *compliance* is not possible and also has been removed. Additionally, *approachability* as sub-characteristic of *usability* is introduced to represent the need for efficiently identifying and fetching the desired component (Hansen, 2006). The description of *usability* sub-characteristics based on state of the art literature (Raje et al., 2001; Kallio and Niemela, 2001; Preiss, 2001; Kalaimagal and Srinivasan, 2008b, Dehlin et al., 2000; Goulao and Brito, 2002a; Goulao and Brito, 2002b; ISO/IEC 9126, 2001; ISO 9241-11, 1998; IEEE Std. 610.12, 1990; Bertoa et al., 2003, Hansen, 2010a; Bertoa, and Vallecillo, 2002a; Botella et al., 2002; Ali et al., 2001; Simao and Belchior, 2003; Washizaki et al., 2003; Simao, 2002; Kallio and Niemela, 2001) are mentioned below.

4.2.1 Help Mechanisms

This sub-characteristic denotes availability and effectiveness of help facility for the usage of the component. It is supported with manuals, demos, tutorials and support tools and services. The extent and depth of the coverage of help facility is dependent upon the type of component. For example, the coverage of help facility in white box component is extensive and comprehensive which includes complete source code and programming philosophy of the component. In case of black box component more emphasis is given to extensive description of interfaces. Supported artifacts as source code and programming philosophy are absent.

Following attributes are used to compute the level of *help mechanisms* supported by the component:

- *Help System:* It indicates availability and completeness of the help files for the system.
- *Manuals:* It provides the presence of *user manual, administration manual* and *installation manual*.
- *Tutorials and Demos:* It indicates presence of *tutorial*s and *demos* to support the usage of component.
- *Support Tools and services:* It denotes the presence of other supporting tools and services to enhance the usage of component such as online help using chat services or telephonic help, expert training modules or sessions etc.

Each of the above attribute can be measured on a scale of 1-5 for level of satisfaction, Table 4.1, where value 5 means very highly satisfied and value 1 means weakly satisfied (very low satisfaction level). It is to be noted that separate measure can also be taken depending upon the depth and the extensibility of the analysis. For example, *user manual* satisfaction level can be identified by further breaking it into – *manual coverage, manual consistency and manual legibility*. Here, *manual coverage* can be measured on the level of satisfaction in terms of percentage **(%)** of functional elements described in manual.

Description	**Level of Satisfaction Scale (1 – 5)**
Very Low (VL)	1
Low (L)	2
Moderate (M)	3
High (H)	4
Very High (VH)	5

Table 4.1 Attribute level of satisfaction (LOS)

4.2.2 Learnability

This sub-characteristic indicates the time needed by a user to learn *how to use, configure and administer* a component. Here, the users are considered to have an average experience in component related projects. This sub-characteristic is very important to original software component developers as the acceptance of component in project/architecture is heavily dependent upon its level of *learnabilty* factor.

The contributing attributes for *learnabilty* are as follows:

- *Time to use:* It indicates the time needed by an average user to learn how a component works and how it can be used in a software system.
- *Time to configure*: It denotes the time required for an average user to learn how to configure the component for its usage.
- *Time to administer:* It indicates the time required for an average user to administer the component.

Similar to *help mechanisms, e*ach of the above attributes can be measured on a scale of 1-5 for a level of satisfaction where value 5 means highly satisfied (very less number of hours an average user has to spend) and 1 means weakly satisfied (very large number of hours an average user has to spend).

4.2.3 Operability

It indicates the level of effort required to *operate, administer and customize* the component.

Following attributes contribute to *operability*:

- *Operation effort:* It indicates the effort required to operate the component. It is heavily dependent upon the suitability of the component on the assigned tasks. For example, whether a component requires some manual tasks to be performed for its operation or not.
- *Administration effort:* It denotes the effort required to administer a component. The functioning is similar to *operation effort* but it mainly focuses on administration related tasks.
- *Customizability effort:* It indicates the effort required to customize a component for its pre-defined interfaces.

Similar to *help mechanisms,* each of the above attribute can be measured on a scale of 1-5 for a level of satisfaction where value 5 means highly satisfied (very less effort required) and 1 means weakly satisfied (very high effort required).

4.2.4 Approachability

It indicates the capability of a component to be searched by its users for its usage through search mechanisms. It is directly linked to the quality of documentation and marketing information provided by the vendor. It is to be noted that though it is heavily dependent upon vendor, it affects the quality of a component and in particular, *usability* in the broader context.

The contributing attributes for *approachability* are as follows:

- *Directory listings*: It indicates the simplicity of finding a component. Dedicated sites on the World Wide Web or special software magazines are the source of components listings and can be used for such findings. A ratio of *popular directory listings that the component is marketed to the total number of popular directory listings* can be mapped to a scale of 1-5 for measuring level of satisfaction. Value 5 means highly satisfied (Very high ratio of presence) and 1 means weakly satisfied (Very low ratio of presence).
- *Search & Fetch:* It denotes the simplicity in searching and fetching a component. For example, what pre-requisites are required for approving download of component (trial version) such as registration, software or hardware, training etc., and whether they are available or not. It can be measured on a scale of 1-5 for level of satisfaction where value 5 means highly satisfied (presence of very effective search & fetch techniques) and 1 means weakly satisfied (presence of very less effective search & fetch techniques).
- *Classification:* It denotes the *supportability* of the classification scheme by a component. For example, the component can be classified according to packaging or platform specific views. It can be measured on a scale of 1-5 for level of satisfaction where value 5 means highly satisfied (presence of very effective classification) and 1 means weakly satisfied (presence of very less effective classification).
- *Marketing information:* It indicates the ability to understand the component capabilities without actually installing it. It deals with the information that the vendor provides as part of marketing strategies. It can be measured on a scale of 1-5 for level of satisfaction where value 5 means highly satisfied (presence of very effective information) and 1 means weakly satisfied (presence of very less effective information).

4.3 Concurrent Usability Digraph modeling

The purpose of concurrent approach is to consider several different activities together in a unified manner in order to attain a goal. This helps in reducing the overall time in reaching to the goal effectively. In this approach several different teams work together and based upon their experiences and expertise they attain the goal effectively and quickly. Evaluation and analysis of *usability* sub-characteristics and associated attributes should not be performed in isolation. It is meaningless to compute the measures of the sub-characteristics independently and later add to obtain the overall *usability* measure. Because in this kind of computation interactions/interdependencies among sub-characteristics are not considered which may have impact on the overall *usability* measure. Thus, in order to formulate realistic *usability* measure for software component, it is necessary to consider measures of sub-characteristics along with their inter and intra dependencies/interactions. Here, several teams can be formed depending upon the requirement of the sub-characteristics and their effects among themselves. For example, let us consider sub-characteristic *help mechanisms*. It consists of four attributes – *help system, manuals, tutorials and demos and support tools and services*. It can be seen that these attributes also affect other sub-characteristics of usability - *learnability, operability and approachability.* Thus, to perform intact analysis or evaluation of a component as per *learning mechanism* point of view other sub-characteristics should not be ignored. To attain proper *help mechanisms* in a component it is utmost important to use concurrent approach which involves considering inter and intra interdependencies/interactions (within attributes and across sub-characteristics respectively) in a unified manner. This can be made possible by involving concurrent teams in analysis and evaluation. It is clear from the above discussion that the sub-characteristics interactions (inter/intra) affect the *usability* of a component. Thus consideration of interactive complexity will enhance the evaluation. This can be accomplished using concurrent approach. The state of the art literature does not provide models which explicitly consider interactions of sub-characteristics along with their measures. This chapter models such approach using digraph model and matrix approach as developed in chapter 2.

There are number of sub-characteristics responsible for creating a component's usability (*help mechanisms, learnability, operability and approachability*). It is to be noted that the effectiveness of the component's *usability* depends upon the degree of inheritance of these sub-characteristics and the amount of interactions (interdependencies, strength of interactions) present amongst them, which conventional representations/techniques are unable to analyze. These interactions may be direction dependent or independent. The network showing these sub-characteristics and interactions is attempted to model the component's *usability* using *graph representation*. If interactions are not direction dependent, the component's *usability* is represented by an undirected graph. The interactions that are direction dependent are represented by digraph. The graph theoretic representation is suitable for visual analysis, it can be computer processed and can be expressed as a mathematical entity, whereas the conventional representations, like block diagrams, cause and effect diagrams and flow charts, although providing visual analysis, do not depict interactions among factors (sub-characteristics) and are not suitable for further analysis and cannot be processed or expressed in mathematical form. The sub-characteristics, their attributes and their measures identified in the previous section are used to evaluate the extent of the component's *usability* for an index known as the component's *usability* index. Thus

Component's usability index = f (elements, interactions) (4.1)

where, elements can be sub-characteristics or attributes. The major point here is to co-relate these elements, their quantification and interactions among them. Based on the above quantification it is proposed to find the concept of *usability* index (I_u), hypothetical best *usability* index (I_{bu}) and hypothetical worst *usability* index (I_{wu}) and benchmarking of component's quality as per *usability* point of view is defined. As already stated, this is achieved through a systematic approach called digraph and matrix approach. It consists of the digraph representation, the matrix representation and the permanent function representation. The digraph is the visual representation of the sub-characteristics and their interactions. The matrix converts the digraph into mathematical form. The permanent function is a mathematical model that helps to improve *usability* design and determine *usability* index. Once the index is determined then the user can rank alternative components and select best as per requirements. Similarly, component designer and developer can perform sensitivity analysis on critical elements (sub-characteristics, attributes etc.,) of *usability* and attain index closer to hypothetical best *usability* index.

A digraph (weighted), here onwards a digraph is referred as a weighted digraph, is used to represent the sub-characteristics and their interactions in terms of nodes and edges. In an undirected (weighted) graph, no direction is assigned to the edges also weights assigned to the edges and nodes/vertices in the graph, whereas directed graphs (weighted) or digraphs (weighted) have directional edges, with respective weights. A *component usability* digraph (weighted) $G_U = (V_U, E_U)$ is defined to represent the interactive complexity (inter and intra) of sub-characteristics of component in terms of nodes and edges. The set V_U consists of finite nodes, *U_i's*, that represents the sub-characteristics of component (e.g. *help mechanisms, learnability, operability and approachability*) and this quantitatively represented by the weight of the respective characteristic/sub-characteristics. The edge set E_U consists of edges, *a_{ij}'s* that represent the strength of interactions (weights) of sub-characteristics of component (see Table 4.2, e.g. strong, medium, weak and nil). The digraph permits visualization of the component *usability* composition and provides interactions among the sub-characteristics. The number of nodes in a digraph is equal to the number of sub-characteristics of a component. Sub-node u_{im} corresponds to the m^{th} attribute of i^{th} *sub-characteristic*, and is placed in the node U_i representing the attribute of a component.

Description	**Strength of Interaction Scale (1 – 5)**
Nil	0
Weak	1
Medium	3
Strong	5

Table 4.2 Strength of interaction (SOI)

Based on Table 4.3, interactions of *usability* sub-characteristics are shown in Figure 4.1. As discussed, directed edges in the digraph represent the strength of interaction of one sub-characteristic on another. A dashed directed edge shows *inter* interactions amongst sub-characteristics while dark bold directed edge shows *intra* interactions of sub-characteristics. The *learnability mechanisms* (U_1) is shown affecting all the other sub-characteristics, i.e. a directed edge from U_1 to U_2, U_3, and U_4. These edges are present due to the interactions of U_1 sub-characteristics i.e. time to use, time to configure and time to administer with the other sub-characteristics of U_2, U_3 and U_4 respectively. It is to be noted that digraph representation is helpful in visual analysis but to the limited extent. As it is not possible to identify the contributed factors such as sub-characteristics or attributes that are responsible for high level or low level of *usability*. Also, it is difficult to identify potential design parameters to attain high level of *usability* design. Thus digraph is the first step which can be used during brainstorming session in order to identify interactions, in particular, potential interactions among sub-characteristics. Later for detailed analysis it is to be converted into some mathematical model where further discussion on various objectives such as- potential parameters affecting *usability*, critical levels of interactions etc, among sub-characteristics can be considered.

<table>
<tr><th rowspan="2">Sub-characteristics</th><th rowspan="2">Attributes</th><th colspan="2">Interactions</th></tr>
<tr><th>Inter</th><th>Intra</th></tr>
<tr><td rowspan="3">U_1 Learnability</td><td>(U_{11}) Time to use</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}; U_{31} ; U_{41} ; U_{42}; U_{43}</td><td>U_{12} ; U_{13}</td></tr>
<tr><td>(U_{12}) Time to configure</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}; U_{33} ; U_{41} ; U_{42}; U_{43}</td><td>U_{11} ; U_{13}</td></tr>
<tr><td>(U_{13}) Time to administer</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}; U_{32} ; U_{41} ; U_{42}; U_{43}</td><td>U_{11} ; U_{12}</td></tr>
<tr><td rowspan="4">U_2 Help mechanisms</td><td>(U_{21}) Help system</td><td>U_{11} ; U_{12} ; U_{13} ; U_{31}; U_{32} ; U_{33} ; U_{43}</td><td>U_{22} ; U_{23} ; U_{24}</td></tr>
<tr><td>(U_{22}) Manuals</td><td>U_{11} ; U_{12} ; U_{13} ; U_{31}; U_{32} ; U_{33} ; U_{43}</td><td>U_{21} ; U_{23} ; U_{24}</td></tr>
<tr><td>(U_{23}) Tutorials and Demos</td><td>U_{11} ; U_{12} ; U_{13} ; U_{31}; U_{32} ; U_{33} ; U_{43}</td><td>U_{21} ; U_{22} ; U_{24}</td></tr>
<tr><td>(U_{24}) Support tools and Services</td><td>U_{11} ; U_{12} ; U_{13} ; U_{31}; U_{32} ; U_{33} ; U_{43}</td><td>U_{21} ; U_{22} ; U_{23}</td></tr>
<tr><td rowspan="3">U_3 Operability</td><td>(U_{31}) Operation Support</td><td>U_{11} ; U_{21} ; U_{22} ; U_{23} ; U_{24} ; U_{43}</td><td>U_{33}</td></tr>
<tr><td>(U_{32}) Administration Support</td><td>U_{13} ; U_{21} ; U_{22} ; U_{23} ; U_{24} ; U_{43}</td><td>U_{33}</td></tr>
<tr><td>(U_{33}) Customizability Support</td><td>U_{12} ; U_{21} ; U_{22} ; U_{23} ; U_{24} ; U_{43}</td><td>U_{31} ; U_{32}</td></tr>
<tr><td rowspan="4">U_4 Approachability</td><td>(U_{41}) Directory Listing</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}</td><td>U_{42} ; U_{43} ; U_{44}</td></tr>
<tr><td>(U_{42}) Search & Fetch</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}</td><td>U_{41} ; U_{43} ; U_{44}</td></tr>
<tr><td>(U_{43}) Classification</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}</td><td>U_{41} ; U_{42} ; U_{44}</td></tr>
<tr><td>(U_{44}) Marketing Information</td><td>U_{21} ; U_{22} ; U_{23} ; U_{24}</td><td>U_{41} ; U_{42} ; U_{43}</td></tr>
</table>

Table 4.3 Usability sub-characteristics interaction (inter and intra level)

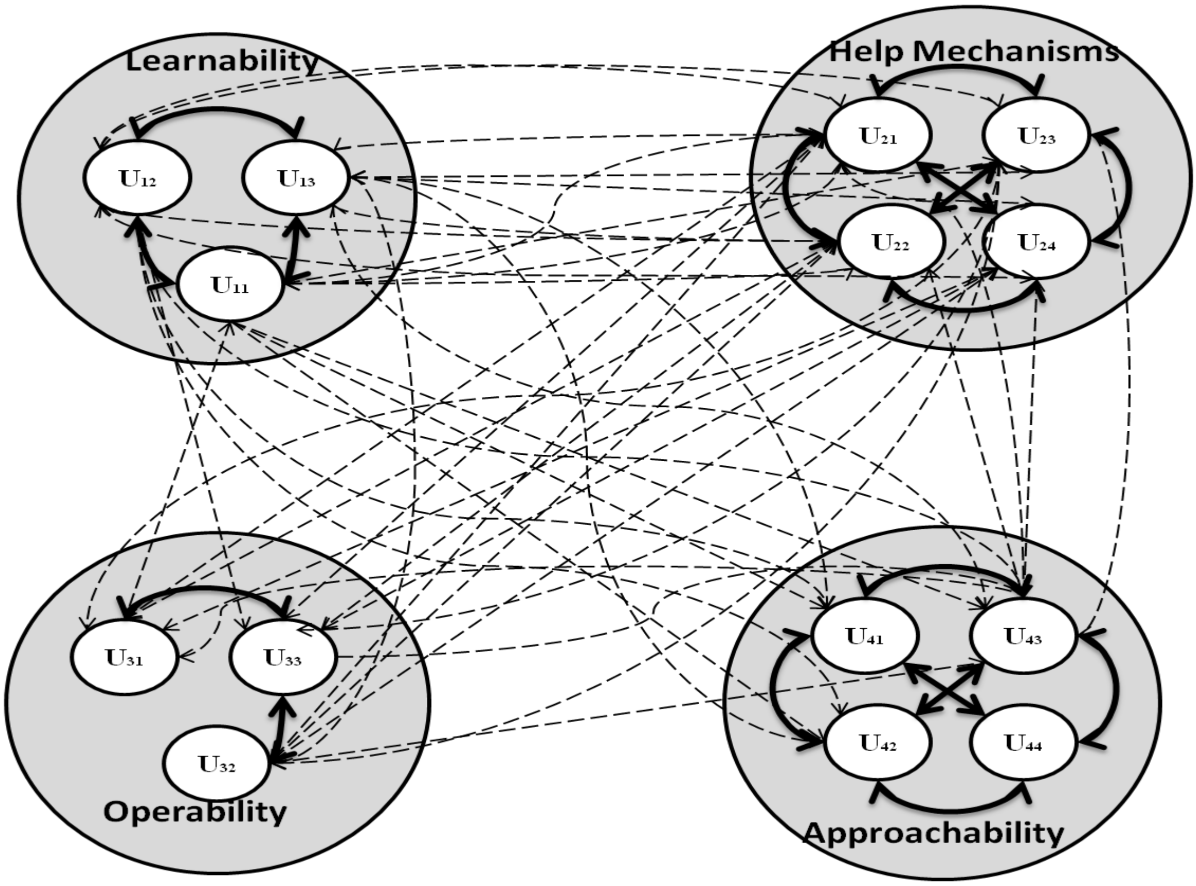

Figure 4.1 Usability sub-characteristics interaction digraph

Dashed line represents sub-characteristics inter interactions

Dark bold line represents sub-characteristic intra interactions

Considering this viewpoint, a one-to-one representation of the considered *usability* digraph, a (permanent) matrix model is developed. This will help in establishing expression characteristics of *usability* of a component and is convenient for computer processing. As digraph, Figure 4.1, contains 4 nodes (sub-characteristics), its matrix representation is of size 4*4. In the matrix, the *diagonal* elements represent *usability* sub-characteristics of a component while *off-diagonal* elements correspond to interaction of one sub-characteristic on another sub-characteristic of a component (direct or indirect interactions of sub-characteristics). *Permanent* of a matrix is called *variable permanent usability function* (*VPF – u*). This expression is the component characteristic of *usability* by considering all interactive complexity. As the *usability* digraph is crucial for study, this may be prepared by the experts in the area/domain. The digraph representation thus permits component's *usability* experts to conceptualize and visualize the developed/designed component's *usability* environment. The designer of the environment can easily modify the environment network to achieve the desired results.

4.4 Matrix Representation: Variable Permanent Usability Matrix (VPM – u)

In this section, *usability* of component is represented using permanent matrix, which provides one-to-one correspondence of the digraph. Permanent matrix keeps information of system characteristics (Upadhyay et al., 2009) thus suitable for *usability* analysis, evaluation and design. The significance and utility of such type of matrix was shown in chapter 2.

The permanent matrix (Jurkat and Ryser, 1996) generates a standard function known as *permanent* which is used in combinatorial mathematics. The *permanent* of a matrix is the matrix multinomial and is called *variable permanent usability function* (*VPF – u*), each term of it has a physical significance related to the component's *usability*. The calculation of a permanent (function) of a permanent matrix is similar to the calculation of a simple matrix determinant. The only difference in permanent matrix multinomial (determinant), *VPF – u*, is that all negative signs that appear in the determinant expression is substituted by positive signs. This multinomial representation includes all the information regarding *usability* sub-characteristics and interactions amongst them. Quantitative component's *usability* evaluation is obtained from *VPF – u*, by substituting numerical values of the permanent expression's (function) elements which are obtained analytically, experimentally or through comparison with the ideal cases. This single numerical index is the representation of a typical component's *usability* in quantitative terms. The *variable permanent usability function*, being the characteristic of the component's *usability* is a powerful tool for its analysis. This helps in representing interactive complexity as viewed from combinatorial viewpoint. The *variable permanent matrix* (*VPM – u*) is capable of representing one-to-one mapping of *usability* digraph. For further analysis matrix can be computer processed. The *VPM – u* generates *VPF – u* from which component's *usability* index is calculated. The benefit of using *VPF – u* is that no information about *usability* (sub-characteristics and inter-intra interactions complexity) of *component* is lost as expression (*VPF – u*) does not contain any negative sign. Application of this concept will lead to better understanding of *usability* analysis of a component.

Let us consider the digraph shown in Figure 4.1 for defining variable permanent usability matrix for *component*. Let a *diagonal matrix*, U_D, with diagonal elements U_i, i = 1, 2,…, 4 be considered. Here, U_i represents the sub-characteristics of *usability*, whose value can be obtained by considering permanent model for associated attributes. Let us also define another matrix, U_O with *off-diagonal* elements, a_{ij}*'s* representing the (strength of) interaction between sub-characteristic *i* and *j*. It is to be noted that U_i*'s* and a_{ij}*'s* represent nodes and edges respectively in *usability* digraph of the component. The matrix for Figure 4.1 is written as:

$VPM - u = [U_D + U_O]$

$$\begin{array}{c} \quad\quad\quad\quad\quad 1 \quad\; 2 \quad\; 3 \quad\; 4 \quad \textit{Sub-characteristics} \\ VPM - u = \begin{bmatrix} U_1 & a_{12} & a_{13} & a_{14} \\ a_{21} & U_2 & a_{23} & a_{24} \\ a_{31} & a_{32} & U_3 & a_{34} \\ 0 & a_{42} & 0 & U_4 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \tag{4.2}$$

The *permanent* of the matrix is written as:

$$VPF - u =$$
$$\mathbf{[}U_1*U_2*U_3*U_4\mathbf{]+[}a_{23}*a_{32}*U_1*U_4 + a_{24}*a_{42}*U_1*U_3 + a_{12}*a_{21}*U_3*U_4 + a_{13}*a_{31}*U_2*U_4\mathbf{]}$$
$$\mathbf{+[}a_{23}*a_{34}*a_{42}*U_1 + a_{21}*a_{14}*a_{42}*U_3 + a_{21}*a_{13}*a_{32}*U_4 + a_{12}*a_{23}*a_{31}*U_4\mathbf{]}$$
$$\mathbf{+[}a_{13}* a_{34}* a_{42}* a_{21} + a_{13}*a_{31}*a_{24}*a_{42} + a_{31}*a_{14}*a_{42}*a_{23}\mathbf{]} \tag{4.3}$$

On a critical analysis of permanent function, expression 4.3, it is inferred that this multinomial contains only distinct sub-characteristics – U_i and loops – $a_{ij}\ a_{jk}\ \ldots\ldots\ a_{ni}$. Here, sub-characteristics are the one which are considered (permanent matrix) for the evaluation of *usability* and they are *help mechanisms, learnability, operability and approachability.* It can be seen that for the directed edges the strength of interaction between two sub-characteristics is perceived as 2-node loop. For example, sub-characteristic U_1 has interaction with U_2 and vice versa which can be represented by 2-node loop ($a_{12}a_{21}$). Similarly, *m*-node loop represents *m*-sub-characteristics strength of interaction based on permanent matrix. For example, in expression 4.3, $a_{13}a_{34}a_{42}a_{21}$ represent strength of interaction of l*earnability on operability, operability on approachability, approachability on help mechanisms and help mechanisms on learnability* in totality. It can be noticed that expression 4.3 does not include a_{41} and a_{43} interactions or its combination with any term as these are not present in the permanent matrix. Thus permanent function, *VPF-u* maintains all the information regarding sub-characteristics and strength of interactions among them concurrently. In short *VPF-u* can be represented as:

$$
\left.\begin{aligned}
VPF\text{-}u &= f(U_{i,},\ a_{ij}^2,\ a_{ij}\,a_{jk}\,a_{ki}\ \text{etc}) \quad \{\text{ if } a_{ij} = a_{ji}\} \\
&= f\,(\text{Nodes, dyads, loops}) \\
&= f\,(\text{Usability structural components}) \\
VPF\text{-}u &= f'\ (U_{i,},\ a_{ij}a_{ji}\ ,\ a_{ij}\ a_{jk}\,a_{kl}\ a_{li}\ ,\ a_{ij}\ a_{jk}\,a_{kl}\ a_{lm}\ a_{mi}\ ) \quad \{\text{ if } a_{ij} \neq a_{ji}\} \\
&= f'\ (\text{Nodes, 2-vertex loops, loops}) \\
&= f'\ (\text{Usability structural components})
\end{aligned}\right\} \qquad (4.4)
$$

The inclusion of *permanent matrix* and *permanent function* to analyze *usability* in component is completely new and is a significant contribution in *usability* analysis which leads to *usability* design and evaluation of component and as a whole to *CBSS*. This is because concurrent teams can simultaneously perform evaluation and analysis. By following the procedure of *permanent matrix* and *permanent function*, end user will be able to take the decision on the improvement, benchmarking, selection and ranking of alternatives components/designs. If an end user is a designer of a component then varying values of *VPF-u* based on sensitivity analysis on critical parameters can select best design for a component. Similarly, if an end user is an integrator then alternative component can be ranked on the basis of *VPF-u* value (index). Both designer and integrator can set up the benchmark - hypothetical best *usability* index and hypothetical worst *usability* index in order to attain improvement in design and ranking alternate components/designs. Section 4.7 and section 4.8 explore the usefulness of the developed methodology in detail with an example.

4.5 Component Usability Evaluation

The *VPF – u* is an invariant thus is a useful expression for the *usability* evaluation of component. Expression 4.3 consists of various terms/entities as point of combinatorial mathematics. Each term of the expression (function) yields a test/heuristic. Expression 4.3 in symbolic form is a useful tool for analysis. Terms in expression 4.3 can be arranged in number of groupings. It is to be noted that mathematically each term is a product of four different matrix elements (N = 4, in the example). Interpreting same expression it can be noticed that different terms are the set of distinct diagonal elements (U_i, U_1,…,U_4) and loops (strength of interactions) of *off-diagonal* elements of different sizes (a_{ij} a_{ji}, a_{ij} a_{jk} a_{ki}). This generates new meaning to multinomial from component *usability* sub-characteristics interactive point of view considering their respective set of attributes and inter-intra interactions among them. By arranging terms of the same structure (set representing sub-characteristics and elements involved in formation of loops) in the same grouping and of different structure in different groupings, *VPF – u* may easily be written in (N + 1) groupings, where N refers to the number of sub-characteristics. The terms of (N+ 1) groups are arranged in the decreasing order of number of vertices/sub-characteristics U_i. The first group contains terms with N unconnected U_i*'s***.** Each successive group has one less sub-characteristic than the previous group and rest of the elements are the combination of dyads and loops. The last group does not contain any U_i in its terms. It contains only terms such as a_{ij}^2, a_{ij} a_{jk} a_{ki} , etc. The evaluation is carried out term by term. If there are M distinct terms then there will M distinct ways for analyzing *usability* of component. The overall analysis is done in the following manner:

1. The first term of expression 4.3 represents *usability* in the combination of *learnability, help mechanisms, operability and approachability* sub-characteristics and is written as

   $$/U_1/U_2/\ U_3/U_4/$$

   Or

   /*learnability*/*help mechanisms*/*operability*/*approachability*/

   A slash represents separation mark between two entities here they are sub-characteristics i.e. *learnability, help mechanisms, operability and approachability*. A designer or practiced expert in the *usability* domain needs to consider the sub-characteristics and associated attributes and suggest ways and means to improve its contribution in increasing *usability* of component (and system).

   For the analysis purpose, the first term is written as:

   /LOS **(***learnability***)** / LOS **(***help mechanisms*)/ LOS **(***operability*)/
   LOS **(***approachability*)/

   Designers and developers of the component have to identify contributed factors and methods which can improve the level of satisfaction (LOS**,** see Table 4.1) of sub-characteristics. Similarly, if an integrator has some component alternatives for particular application domain then by putting respective sub-characteristics LOS values, best out of alternatives can be identified.

If entity two i.e. *help mechanisms* is more critical to a component in some application domain then designers and developers of a component have to identify the factors attributed to high level of *help mechanisms*. It is to be noted that even if a component is mature and provide required functionality to the application domain it is not useful if it does not contain required level of *help mechanisms*. A manual should be designed and created in such a manner that it should have minimum number/percentage of functional elements incorrectly described. It must cover detail description of functional elements, interfaces, methods and configurable parameters, difference in component version if any etc. It is to be noted that acceptable and understandable ratio of words per functional element, ratio of words per interface, ratio of words per methods and ratio of words per configuration parameters also play important role from end user point of view to learn usage of component functionality quickly and effectively. Similarly other contributed factors have to be identified to increase the level of *usability* sub-characteristics. It is to be noted that the concurrent team can be prepared which will focus on first term entities and which will be responsible to find out the method and ways to increase the level of satisfaction of respective entities. For example, one team can look for sub-characteristic *help mechanisms* while at the same time other team can focus up on *operability* of a component. Thus using aforementioned technique concurrently many team can work together to understand and identify ways to improve the first term of expression 4.3. Using similar approach, the attributed factors of other sub-characteristics can be identified.

2. The second term is absent as there is no self loop present in the digraph. That means no entity in the terms of expression 4.3 has interaction with itself.
3. The third term consists of two – sub-characteristics loop (i.e. $a_{ij}a_{ji}$), which represents strength of interaction between two entities, and N – 2 sub-characteristics. The term is represented as

$$/a_{ij}a_{ji}\ /U_k/U_m/$$

or

$$/A_{ij}/U_k/U_m/$$

For convenience $a_{ij}a_{ji}$ is represented as A_{ij}. In the above set the entity to be analyzed first is $a_{ij}a_{ji}$. This is a two – sub-characteristics loop and it represents interactive quality between sub-characteristic *i* and *j*. If the resultant value is towards lower side as per the analysis, then in-depth study is needed to increase its value. For the present *usability* digraph, the third term contains four entities:

$$/a_{23}a_{32}U_1U_4/a_{24}\,a_{42}U1U_3/a_{12}a_{21}U_3U_4/\,a_{13}\,a_{31}U_2\,U_4/$$

The first entity of first term of this group can be written as:

/SOI (between *help mechanisms* and *operability*)

LOS (*learnability*) LOS (*approachability*)/

The first entity to be analyzed is the two – sub-characteristic loop i.e. $a_{23}a_{32}$**,** which represents interaction between tutorials and demos and administration support. It is to be noted that the use of good quality tutorials and demos to handle administration related tasks affects the administration skills. Thus if the interaction is critical then further analysis will reveal the elements (just-in-time availability of administration training, online/offline trouble shooting etc.,) that has to be enhanced in order to increase the interactive quality and overall impact.

4. The fourth set consists of a three – sub-characteristics loop i.e. A_{ijk} or its pair A_{ikj} and N - 3 sub-characteristics. The set is represented as:

$$/A_{ijk} + A_{ikj}\,/U_m/$$

The first entity to be analyzed is the three – sub-characteristics loop A_{ijk} or its pair A_{ikj}. The entities values for the fourth set are:

$$/a_{23}\,a_{34}\,a_{42}\,U_1\,/\,a_{21}\,a_{14}\,a_{42}\,U_3\,/a_{21}\,a_{13}\,a_{32}\,U_4\,/a_{12}\,a_{23}\,a_{31}\,U_4/$$

The first entity to be analyzed in the set is A_{234}. This is the resultant interaction relation between sub-characteristics *help mechanisms*, *operability* and *approachability*. That is the interactive quality between *help mechanisms* and *operability* and between *operability* and *approachability* and between *approachability* and *help mechanisms* are to be studied. Along with this analysis, other entities of the term have to be considered and studied for the resultant interactive quality.

5. The entities of the fifth grouping are arranged in two sub-groupings. The entities of the first sub-grouping are two – sub-characteristics loop (i.e., $a_{ij}a_{ji}$). The second sub-

grouping consists of a four – sub-characteristics loop (i.e., $a_{ij}\ a_{jk}\ a_{kl}\ a_{lm}\ a_{mi}$) or its pair ($a_{im}\ a_{ml}\ a_{lk}\ a_{kj}\ a_{ji}$). The fifth set contains three entities.

$$/a_{13}\ a_{34}\ a_{42}\ a_{21} /\ a_{13}\ a_{31}\ a_{24}\ a_{42} /\ a_{31}\ a_{14}\ a_{42}\ a_{23} /$$

Since in the expression 4.3 for fifth sub-group there is no two – sub-characteristics loop thus terms of first sub-group are absent. The second group has $a_{13}a_{34}a_{42}a_{21}$ as four – sub-characteristics loop. This represents interaction between *learnability* and *operability* and between *operability* and *approachability* and between *approachability* and *help mechanisms* and between *help mechanisms* and *learnability*. Standard procedures and techniques must be used to improve the overall interactive quality of the loop. Along with this other entities sub-groups have to be studied and analyzed.

It is to be noted that the required procedures and techniques have to be used to improve the overall quality of the entities considering interactive complexity as per the *usability* point of view. Finally, this leads to the improvement in the *usability* of a component. In order to get complete overview of *usability* of a component an in depth analysis on the basis of aforementioned procedure has to be employed and performed.

4.6 Component Usability Index (I_u)

A quantitative or qualitative index/measure is needed to evaluate the *usability* characteristic. The lexical meaning of the index is "*a number or formula expressing some property, ratio, etc. of something indicated*" (Lexico Publishing Group, 2006) and "*something that leads one to a particular fact or conclusion*" (Merriam-Webster, 2006). Thus, a *usability* index must be able to represent the extent of *usability* of a component. The *usability* index (I_u) is a quantitative measure of the *component's usability*. As *VPF – u* considers structural and interactive complexity of sub-characteristics and associated attributes it can be used to generate the measure. Based on the I_u the selection and evaluation of the component can be carried out as per *usability* point of view. A computer program in 'C++' language is developed (*Appendix F*) to compute the value of permanent of *usability* characteristic. To evaluate *VPF – u*, numerical value of U_i and a_{ij} is required.

It is to be noted that using *usability* index **(I_u)**, one can carry out the comparison of two or more than two alternative components available in the market as per *usability* point of view. The designers and developers of the component will also get to know which factor has to be improved so as to increase the overall *usability* of a component. The higher the value of the index, the more is the *usability* of the component. This is due to the presence of high values of U_i and a_{ij}*'s*. The lower value of index implies lower *usability* of the component and is a result of low values of U_i and a_{ij}*'s*. Based on this, given component and/or component designs are compared and may be ranked from increasing or decreasing value of index. This aids in the selection of an optimum component (from market/ off-the-shelf) and component designs (as per designers point of view considering *usability* aspects) based on the *usability*. Permanent, expression 4.3 is a function of component *usability* of a number of distinct structural elements i.e. U_i, $(a_{ij}a_{ji}/a_{ij}^2)$, $(a_{ij}a_{jk}...a_{mi})$ etc. Decision maker may compare potential candidates on the basis of these structural elements. This comparison provides complete insight for making right selection of acquiring component in a project.

4.7 Case Study

The applicability of the developed approach is validated with the *usability* evaluation of database component (C_x, ORACLE) selected from the component classified list (Upadhyay and Deshpande, 2010), see Table 3.4, for a typical component based web application project (Hong , 2005; Upadhyay et al., 2010). Table 4.4 lists down the level of satisfaction, utilizing Table 4.1, and strength of interaction, utilizing Table 4.2, of *sub-characteristics and associated attributes considering interactions*.

The interaction of i^{th} sub-characteristic on j^{th} sub-characteristic, a_{ij}, can be computed in following ways:

- if there exists only one interaction between i^{th} sub-characteristic on j^{th} sub-characteristic then direct value from Table 4.2 is used in place of a_{ij}.
- if i^{th} sub-characteristics have multiple interactions with many attributes of j^{th} sub-characteristics then value of a_{ij} is computed by taking root mean square value of all interactions.

To obtain the realistic values of U_i*'s* and a_{ij}*'s*, Table 4.4 may be prepared by experts. For calculation of the I_u the permanent of the matrix, expression 4.3 is utilized and is given as:

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \quad \textit{sub-characteristics} \\
VPM-u = & \\
I_u = per(u) = 9.883993\text{e}{+}09 & \begin{bmatrix} 158 & 1.19 & 2.88 & 0.33 \\ 1.04 & 486 & 1.04 & 2 \\ 2.88 & 1.04 & 57 & 0.33 \\ 0 & 0.90 & 0 & 2256 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
$$

The index is the estimated index of the *usability* of a database component (C_x) in typical component based web application project. This may be compared with the other alternate candidate indices.

Elements	**LOS** $^{\epsilon\epsilon}U_{ij}$	Elements	**SOI** $^{*}U_{ij}$ **and** $^{**}U_{ijkm}$	Elements	**SOI** $^{*}U_{ij}$ **and** $^{**}U_{ijkm}$	Elements	**SOI** $^{*}U_{ij}$ **and** $^{**}U_{ijkm}$	Elements	**SOI** $^{*}U_{ij}$ **and** $^{**}U_{ijkm}$	Elements	**SOI** $^{*}U_{ij}$ **and** $^{**}U_{ijkm}$
U_1	158	U_{12}	1.9	U_{1323}	3	U_{2311}	3	U_{2343}	1	U_{4121}	3
U_2	486	U_{13}	2.88	U_{1324}	5	U_{2312}	3	U_{2443}	1	U_{4122}	3
U_3	57	U_{14}	0.33	U_{1131}	5	U_{2313}	3	U_{3111}	3	U_{4123}	3
U_4	2256	U_{21}	1.04	U_{1132}	5	U_{2411}	3	U_{3112}	5	U_{4124}	5
U_{11}	4	U_{23}	1.04	U_{1133}	5	U_{2412}	3	U_{3113}	5	U_{4221}	3
U_{12}	3	U_{24}	2	U_{1141}	1	U_{2413}	3	U_{3121}	3	U_{4222}	3
U_{13}	3	U_{31}	2.88	U_{1142}	1	U_{2131}	3	U_{3122}	5	U_{4223}	3
U_{21}	4	U_{32}	1.04	U_{1143}	1	U_{2132}	3	U_{3123}	3	U_{4224}	5
U_{22}	3	U_{34}	0.33	U_{1241}	1	U_{2133}	3	U_{3124}	3	U_{4321}	3
U_{23}	2	U_{42}	0.90	U_{1242}	1	U_{2231}	5	U_{3221}	3	U_{4322}	3
U_{24}	2	U_{1121}	3	U_{1243}	1	U_{2232}	5	U_{3222}	5	U_{4323}	3
U_{31}	4	U_{1122}	5	U_{1341}	1	U_{2233}	5	U_{3223}	3	U_{4324}	3
U_{32}	3	U_{1123}	3	U_{1342}	1	U_{2331}	3	U_{3224}	3	U_{4421}	5
U_{33}	3	U_{1124}	5	U_{1343}	1	U_{2332}	3	U_{3321}	3	U_{4422}	3
U_{41}	2	U_{1221}	3	U_{2111}	3	U_{2333}	3	U_{3322}	5	U_{4423}	3
U_{42}	2	U_{1222}	5	U_{2112}	3	U_{2431}	5	U_{3323}	3	U_{4424}	5
U_{43}	3	U_{1223}	3	U_{2113}	3	U_{2432}	5	U_{3324}	3		
U_{44}	3	U_{1224}	5	U_{2211}	5	U_{2433}	5	U_{3143}	3	---	---
---	---	U_{1321}	3	U_{2212}	5	U_{2143}	1	U_{3243}	3	---	---
---	---	U_{1322}	5	U_{2213}	5	U_{2243}	1	U_{3343}	1	---	---

* SOI: from U_i to U_j ; ** SOI: from U_{ij} to U_{km} ; $^{\epsilon\epsilon}$ LOS: of U_{ij}

Green color cells represent permanent values of respective sub-characteristics

Blue color cells represent root mean square values of SOI between respective sub-characterics.

Table 4.4 C_x elements and respective values (sub-characteristics associated values considering strength of interactions)

4.8 Usefulness of the Developed Methodology

The developed methodology will benefit both component designers and developers, and software architect/software designers and decision makers (responsible to assess, rank, select, procure and acquire component) as per *usability* point of view. Computer software is developed (*Appendix F*) to automate the process of permanent computation.

- *Benefit to software* architect/software designers and decision makers

  The *usability* of a complete *CBSS* depends upon *usability* of all components considering interactions among them. It is very important and critical for architect, designer and decision makers to select best component from a pool of alternatives. A spider web diagram, see Figure 4.2, can be utilized for ranking purpose. Each sub-characteristic of *usability* is represented with a *spoke*, where each *spoke* represents unique value/measure from set of values in the acceptable range minimum to maximum of respective sub-characteristic. For example in the current example a spider web diagram has 4 *spokes* i.e. *help mechanisms, learnability, operability and approachability*. The hypothetical maximum (best) *usability* and hypothetical minimum (worst) *usability* value (index) of sub-characteristic (as per *usability* factors point of view) of a component has to be placed in the respective *spoke*. Later, each alternative component (in Figure 4.2, alternative 1 and alternative 2 are shown for visualization purpose) has to be analyzed for each *spoke* and a web (by joining sub-characteristic value in respective *spokes*) can be created for visual analysis. It is to be noted that component which is closer to the hypothetical best value will be selected as a prime candidate. It can be seen that the level of satisfaction for component sub-characteristics and associated attributes are in the range of 1-5 and strength of interaction from 1-5. It is expected that the strength of interactions may vary from one project to another. Once the strength of interactions is fixed then the permanent index value will be contributed by the variations in the values of sub-characteristics or associated attributes level of satisfaction.

  To get hypothetical minimum *usability* index the level of satisfaction for sub-characteristics and/or associated attributes has to be set to 1. The resultant value after performing permanent computation is the hypothetical minimum *usability* index of a component. To get hypothetical maximum *usability* index the level of satisfaction for sub-characteristics and/or associated attributes has to be set to 5. The resultant value after performing permanent computation is the hypothetical maximum *usability* index of a component. It is to be noted that decision makers are free to use different scale for different sub-characteristics and/or associated attributes which will result in different hypothetical maximum and minimum *usability* indices respectively. For the simplicity, the same scale is considered.

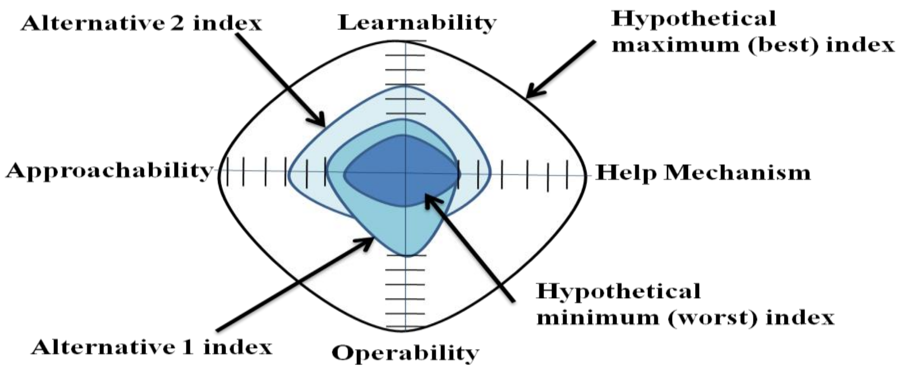

Figure 4.2 Spider diagram for usability characteristic

According to customer potential requirements two candidates C_x and C_y were short listed for evaluation and ranking from the classification list which is shown in Table 3.4.

The index of sub-characteristic *help mechanisms* for C_x can be calculated as:

$$\begin{array}{cccccc} & 1 & 2 & 3 & 4 & \textit{attributes} \end{array}$$

$$VPM - u = \begin{bmatrix} 4 & 2 & 1 & 1 \\ 3 & 3 & 1 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 3 & 3 & 2 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

I_u(*help mechanisms*) = *VPF-u*(*help mechanisms*) = 486

The hypothetical best index of sub-characteristic *help mechanisms* for C_x can be calculated as:

$$\begin{array}{cccccc} & 1 & 2 & 3 & 4 & \textit{attributes} \end{array}$$

$$VPM - u = \begin{bmatrix} 5 & 2 & 1 & 1 \\ 3 & 5 & 1 & 1 \\ 1 & 1 & 5 & 3 \\ 3 & 3 & 3 & 5 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

I_{bu}(*help mechanisms*) = *VPF-u* (*help mechanisms*) = 1546

The hypothetical worst index of sub-characteristic *help mechanisms* for C_x can be calculated as:

$$
\begin{array}{cccccc}
 & 1 & 2 & 3 & 4 & \textit{attributes} \\
\end{array}
$$

$$
VPM-u=\begin{bmatrix} 1 & 2 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 3 \\ 3 & 3 & 3 & 1 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}
$$

I_{wu}(*help mechanisms*) = *VPF-u* (*help mechanisms*) = 194

Based on the same concepts different *usability* sub-characteristics indices (permanent values) for C_y (IBM DB2) component is computed which is shown in Table 4.5. The values of SOI remain constant for the overall comparison and have been utilized from Table 4.4.

Elements	**Permanent value**	Elements	**LOS €€*U*ij**	Elements	**LOS €€*U*ij**	Elements	**LOS €€*U*ij**	Elements	**LOS €€*U*ij**
U_1	87	U_{11}	2	U_{22}	2	U_{32}	2	U_{43}	3
U_2	332	U_{12}	2	U_{23}	2	U_{33}	2	U_{44}	4
U_3	20	U_{13}	2	U_{24}	2	U_{41}	3	---	---
U_4	2826	U_{21}	2	U_{31}	2	U_{42}	3	---	---

€€LOS: of ***U*ij**

Green color cells represent permanent values of respective sub-characteristics

Table 4.5 ***C*y** elements and respective values (sub-characteristics associated values)

Similarly, hypothetical maximum and hypothetical minimum index of other sub-characteristics for C_x and C_y are obtained, see Table 4.6. After getting values for each sub-characteristics hypothetical best and hypothetical worst *usability* index of C_x and C_y are obtained (utilizing sub-characteristics value) based on same methodology. It is to be noted that component C_x is ranked higher than component C_y due to its higher *usability* index value, Table 4.7. Decision maker can also set acceptable threshold values of each sub-characteristic and attributed factors for component and can compare the candidates' respective values against the set threshold values. The final selection of components will depend upon other factor such as- business policies, business vision, cost, environmental conditions, legal issues etc.

Characteristics	Permanent value (C_x)	Permanent value (C_y)	Hypothetical maximum Value	Hypothetical Minimum Value
$U_{1:}$ Learnability	158	87	290	46
$U_{2:}$ Help mechanism	486	332	1546	194
$U_{3:}$ Operability	57	20	155	7
$U_{4:}$ Approachability	2256	2826	4848	1480

Table 4.6 Sub-characteristic permanent value, hypothetical maximum and hypothetical minimum value

Component alternatives	Permanent value	Hypothetical maximum value	Hypothetical minimum value	Ranking
C_x	9.883993e+09	3.369653e+11	9.493108e+07	I
C_y	1.640664e+09	3.369653e+11	9.493108e+07	II

Table 4.7 Ranking for C_x and C_y

- *Component designer and developer*

It is very important for component designers and developers to identify potential *usability* factors affecting component before taking decision to float component in the global market. Considering *help mechanisms* as a factor under study for design and development of a component the issues such as preparing help systems, manuals, demos and tutorials, providing support tools and services, marketing information, component directory listing, facility of easily searched and fetched, meaningful names to functional elements – interfaces, methods, parameters etc., have a major impact on component adoption in CBSS project. It can be seen that the end-users do evaluation of components based upon *usability* factors/ sub-characteristics. Thus component designers and developers have to concentrate on *usability* factors and perform the sensitivity analysis in order to improve the overall *usability* of a component. For *help mechanisms,* following decisions are very important:

- percentage of functional elements, interfaces, methods and configurable parameters to be covered in manuals,

- percentage of functional elements covered in demos and tutorials and receptive threshold values,
- allowable complexity threshold value
- ratio of words per functional element,
- ratio of words per interface,
- ratio of words per methods
- ratio of words per configuration parameters,
- demos and tutorials version difference receptive threshold values etc,

Similarly, for *marketing information*, following decisions are very critical:

- choice for potential or all directory listing,
- number of functional elements covered,
- provision for providing information for help mechanisms etc,

Care has to be taken in incorporating all in a component in an efficient manner in order to consider *usability* perspectives from end-user point of view. Likewise other issues can also be identified and considered for detailed level of designing and development. For each sub-characteristic, spokes have to be identified based on the attributed factors. For example, *help mechanisms* has four attributed factors – *help system, manuals, tutorials and demos* and *support tools and services.* Benchmarking can be done by placing hypothetical maximum and hypothetical minimum value (index) of attributed factor for a sub-characteristic (as per usability factors point of view) of a component in the respective spoke. Later, standard procedures and techniques have to be used to increase the value of each contributed factor in order to improve the sub-characteristic value which will contribute to the *usability* of a component. Sensitivity analysis can also be applied by varying the parameters/attributed factors values. Special attention has to be given in improving attributes value so that the overall sub-characteristic value should reach closer to the hypothetical maximum value. It is to be noted component C_x is considered to understand the applicability of the proposed benefits. For getting hypothetical minimum index all sub-characteristics *LOS* and *SOI* values are set to minimum, see Table 4.1 and Table 4.2, respectively. Similarly, for getting hypothetical maximum index all sub-characteristics *LOS* and *SOI* values are set to maximum, see Table 4.1 and Table 4.2, respectively. It is expected that the strength of interactions may vary from one designer to another. Once the strength of interactions is fixed then the permanent index value will be contributed by the variations in the values of sub-characteristics or associated attributes level of satisfaction. Designers and developers are free to use different scale for different sub-characteristics and/or associated attributes which will result in different hypothetical maximum and hypothetical minimum index/value.

Since the permanent value of *help mechanisms* for C_x is value **1546** which is not equal to its corresponding hypothetical maximum value thus there exist a provision to improvise attributed factors. For example, component designers and component developers can think of improving attributes - *tutorials and demos* and *support tools and services* as attributes respective value in the matrix is 2. This can be done by developing interactive, graphical and animated tutorials, easy to understand screen shots, providing on demand services, provision for round the clock expert availability for trouble shooting etc. Thus when the value is improved from 2 to 3 then there will be major change (improvement) in permanent value of *help mechanism*, which will lead to higher value of component C_x *usability*. Using the developed methodology designers and developers can benchmark their component and devise strategies to enhance the *usability* index of components. Table 4.8 shows the sensitivity analysis performed on sub-characteristics with special emphasis on tutorials and demos and support tools and services (making other factors associated *LOS* and *SOI* to be constant).

Attributes for **(C_x)**	**Case I**	**Case II**	**Case III**	**Case 4**
U_{21} Help System	4	4	4	4
U_{22} Manuals	3	3	3	3
U_{23} Tutorials and Demos	2	3	2	3
U_{24} Support tools and Services	2	2	3	3
per(help mechanism)	486	558	534	624
per(u)	9.88393e+09	1.13482e+10	1.086013e+1	1.269039e+10

Orange color cells represent change of attribute value

Blue color cells represent change of permanent value

Table 4.8 Sensitivity analysis on tutorials and demos and support tools and services

It can be easily seen that a little change in a value will have a major impact on *usability* index value. From the analysis, it can be seen that identification of standard tools and techniques should be employed to increase the *LOS* of tutorials and demos as this is the candidate to high *usability* index of a component as compared to support tools and services.

4.9 Concluding Remarks

In this chapter, component specific *usability* characteristic along with sub-characteristics, associated attributes, measures and interactive complexity have been identified. Later a mathematical model comprising of digraph and matrix approach is developed to analyze concurrently *usability* characteristic of a software component based on attributed factors which leads to improvement of the component *usability* both at the designing and development of component. End user such as component designer, developer, acquirer and integrator will get benefit by utilizing the developed concept of hypothetical best *usability* index and hypothetical worst *usability* index. By benchmarking a component or component design for having best index it is possible for the designer and the developer of a component to identify critical parameters contributed to the high level of *usability*. By understanding and interpreting the terms of *permanent function* as discussed, the design and the process can be improved which thereby improve the *usability* of a software component. Similarly, acquirer and integrator can rank alternatives components based on *permanent function* value (*usability* index) and select the best whose value is closer to hypothetical best value.

In the next chapter, *maintainability* aspect of software component is dealt. Design for *maintainability* is an important step in creating flexible systems. In the chapter, firstly contributing factors for *maintainability* are discussed then an insight is given to concurrent evaluation of software component's *maintainability* using graph theoretic systems approach. The chapter also discusses development of *maintainability* index by which optimal selection of design and development of component as per *maintainability* point of view can be achieved.