Chapter - 2

Modeling and Analysis of Component Based Software Systems

2.1 Overview

The present chapter provides a methodological framework based on systems engineering approach and graph theory to model and analyze component based software system. The methodology developed in this chapter not only takes into account the structure/architecture of the CBSS, but also effectively incorporates interactions among sub-systems using concurrent engineering approach. The main focus of the developed methodology is to:

- formulate system equation, which is a characteristic of the quality of the CBSS
- compare, evaluate and select alternate CBSS designs, and
- build effective CBSS considering all the aspects together in a unified manner

The developed methodology for modeling and analysis is a useful exercise for a designer as it enables him/her to select the best possible *component, sub-system* (*composition*) and structure of the CBSS at the initial stage of design, from the quality point of view. The rest of the chapter is organized as follows: In section 2.2 concepts necessary to build up the methodology is provided. Section 2.3 gives a description of (sub-) systems identification process. Section 2.4 discusses modeling of component based software system using graph theoretic systems approach. Section 2.5 provides an insight into a mathematical model, capable for computer processing by utilizing matrix algebraic approach. In section 2.6, evaluation of sub-systems is presented based on the developed approach. Section 2.7, discusses the generalized way of evaluating *N sub-systems* (sub-sub- systems down up to the component level and their interactions) of CBSS. In section 2.8, a structural comparative approach is shown by which two CBSS or set/family of CBSS can be compared on the basis of the aspects of structural complexity and interactions. In section 2.9, a case study of typical *component based web application* is used to demonstrate and validate the applicability of the developed methodology (same example is used to build up the methodology). In section 2.10, step-by-step procedure is documented to perform the methodology. Section 2.11, highlights the usefulness of the methodology to the stakeholders of CBSS project. Finally, Section 2.12 provides concluding remarks of the chapter.

2.2 Introduction

In the current scenario, a rapid demand for customizable, cost effective, just-in-time and reusable large-scale and complex software systems has invoked a new challenge before the software community. To overcome the underpinning challenge, the new trend is to develop component based software systems. Customers demand high quality and best features at low cost in every software system (product). Complexity has increased while the product life has reduced considerably. For the rapid software development, software designers are encouraged to integrate commercial-off-the-shelf (*COTS*) components in their software systems. Component-based software engineering as a novel thrust area, in particular, (Cai et al., 2000; Kozaczynski and Booch, 1998) has shown remarkable success in developing cost-effective and reliable applications to meet short time-to market requirements. Performance, reliability, dependability and other characteristics of software architecture is mostly analyzed and measured only at the time of implementing artifacts. It has been identified by industry and academia that analysis of architecture for risk and quality factors is very important to a project's success (Bosch, 2001; Clements et al., 2002; Kruchten, 1995; Shaw and Garlan, 1996). Systems engineering focuses on study of inter/intra communication, interactions and dependencies of systems/sub-systems and it has been evolved as a novel approach to model software architectures (Saradhi, 1992). It has been shown by researchers that the overall quality of a system depends upon the interaction/interdependencies of its systems and sub-systems (Maes and Guttman, 1998; Gray, 1997; Papaionnou and Edwards, 1998). One of the critical factors in pertaining overall software quality is the quality of a system's software architecture. The analysis of risks inherent at the software architecture level has shown a cut down in overall development cost (Juan et al., 2007). Therefore, a good architecture is important in order to achieve a high-quality software system, both in terms of development and long-term *maintainability*.

2.3 System Identification Process

The purpose of this section is to identify the potential of systems approach as a first step in building up of CBSS. A case study of typical component based web application (Hong, 2005) is used throughout the chapter to demonstrate and validate the developed methodology. One of the critical steps in building up CBSS is the identification of elements and their participation to achieve a goal. Using the systems approach a top-level CBSS system is viewed as a combination of various systems and sub-systems. In order to decompose system(s) into sub-systems down up to component level a complete understanding of systems approach and decomposition criterion is required. A decomposition criterion can be used to decompose the system into subjects or concerns by which specific sub-goals can be achieved. Saradhi (1992) has defined the concept of systems modeling by considering different views- *world view, environment view* and *component view*. These views are general and applicable to anything which can be considered as a system. Pressman (2005) has refined the earlier concept to fit into the software domain by including *world view*- to establish business or technology context as per the domain of interest, *domain view*- (using specific elements) to accomplish objectives and goals of domain and *element view*- to specify technical components that achieve the necessary function for the element. According to him, component may be visualized as the non-decomposable or minimal part of the system. Hatley and Pirbhai (1987) in their research work have discussed the usage of system model template to concentrate on five process regions- *user interface*, *input*, *system function* and *control*, *output* and *maintenance* and *self-test*. The systems model template creates the boundary by defining the context using system context diagram (*SCD*). The most frequently used decomposition criteria for system are defined below (Tagoug, 2002):

- *data criterion*: *component* sharing same data
- *business function criterion: component* contributing to achieve same business function
- *time criterion*: *component* instantiated in the same slice time
- *organizational structure criterion*: *component* belonging to the same organizational unit
- *behavior criterion*: *component* affected by change of state in some *component*

It is clear from the above discussion that, to build up complete CBSS knowledge of the following are required: system's confined boundary (considering all possible constraints), system's constituents i.e. elements and their participation and system's decomposition criteria. In order to perform complete designing and analysis of CBSS the contributing factors other than the main physical sub-systems and their interconnections have to be considered. It is to be noted that a sub-system is a system in itself. To define a CBSS engineering process, an outline of the necessary tools and procedures to support it are required. This is followed by identifying system requirements which can be broken down for further analysis, generating their own set of requirements. The whole process is repeated containing more detailed view of the system and sub-systems, until the component level is reached. By utilizing the system approach in building up CBSS, evaluation and proper accommodation of new concepts and technology is possible. Once a system model is created then it can be transformed into some mathematical entity which can be further used for detailed analysis. Based upon systems approach a typical component based web application with following sub-systems Figure 2.1 is considered:

- Client Side User Interface (Client Side UI)
- Web Application Server
- Database Server
- Page Generator

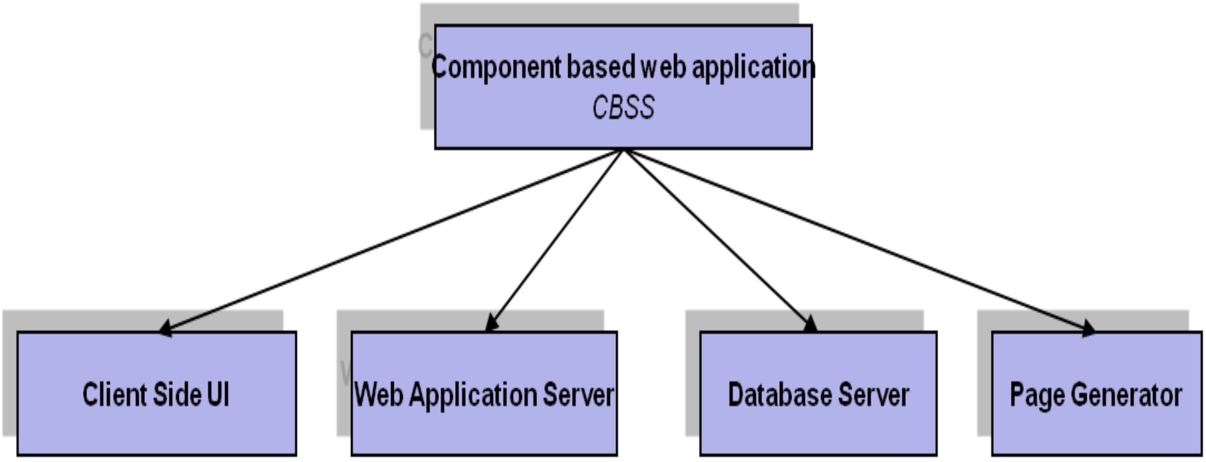

Figure 2.1 Sub-systems of typical component based web application (system structure tree)

The purpose of the above application is just to demonstrate the utility of systems approach and other approaches to build up complete CBSS by analyzing, evaluating and understanding its overall characteristics in order to achieve the goal. The software industry is free to identify more or altogether different sub-systems depending upon requirements, aim, scope, objectives and domain. Figure 2.1 does not show interactions between sub-systems. It just represents a hierarchical decomposition of concerns. In the real application, interactions are present among these sub-systems. The component based typical web application in Figure 2.2, includes the interactions. It consists of four components (Hong, 2005) – User interface (*UI*), *web (application) server, database server* and *page generator*. The *UI* executes on the user's computer and is connected to the *web (application) server* through internet. The *web (application) server* processes sequence of user's requests, which is in the form of messages, and passes the request to a *database server* for processing database queries.

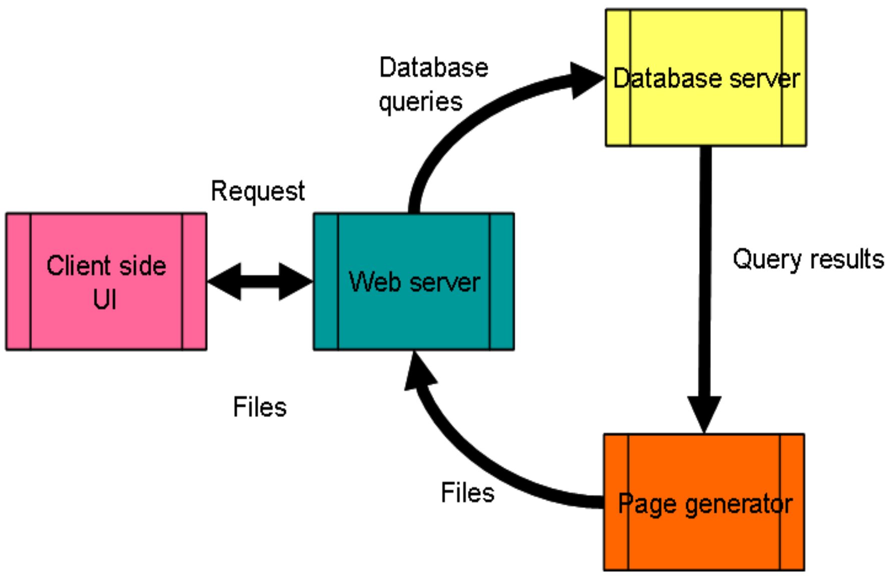

Figure 2.2 Typical component based web application structure along with interactions (adopted from Hong, 2005)

The result of the database queries are processed by the page generator to produce required file, which is sent back to the UI of user's computer screen to display through web (application) server.

A hierarchical decomposition utilizes "top-down" approach to perform overall designing and analysis of CBSS system. In this, system, sub-systems, sub-sub-systems etc are identified up to the component level. The tree structure as generated allows all the parts to be designed from components level to the system level in the hierarchical order by using "bottom-up" approach. This helps to ensure design and geometric compatibility in the system. In general, the hierarchical tree structure may have (N+ 1) level where N represents the number of distinct concerned sub-systems. The hierarchical trees of CBSS may differ depending upon the choices of the distinct sub-systems up to the component level. Identification of tree structure helps in understanding of CBSS system engineering process and acts as an asset in improving quality of the system.

2.4 Graph Theoretic System Modeling of *CBSS*

From the previous section it can be seen that a hierarchical decomposition of a system in the form of system hierarchical tree will help in understanding the basic elements and outline of the CBSS. In reality such sub-systems in a system hierarchical tree collaborate and participate together in order to achieve some goal. To understand such kind of collaboration or participation, a mere hierarchical tree representation would not work. Thus a model is needed which should be capable enough of representing and analyzing such kind of elements and their participations considering all kind of constraints in a unified manner. To accomplish this, a graph theoretic systems approach which comprises of *graph model*, *permanent matrix* and *permanent function* is developed. A graph model is capable of representing a complete system considering all interaction of sub-systems together. A *permanent matrix* is a one-to-one representation which converts the graph model in a mathematical form and is capable for computer processing. A *permanent function* is a mathematical model which characterizes complete CBSS in a single system equation. Utilizing such system equation various stakeholders can get benefit. For example, designers can take decisions on whether to improve designs or not by identifying critical parameters subjected to it. Similarly, an integrator can rank and select best CBSS design, or sub-system or component by putting proper, precise and accurate values in the system equation.

A system graph $G_s = (V_S, E_S)$ is used to model CBSS. Let each of the sub-systems of node set V_S, of CBSS be represented by S_i ($i=1,\ldots, n$), as nodes and the interactions among these sub-systems as an edge set E_S by edges e_{ij} ($i, j = 1,\ldots,n$) connecting the two nodes S_i and S_j. The graph theoretic representation (V_S, E_S) of node and edge sets of the N sub-systems of CBSS is called the CBSS system structure graph. Various types of edges and weights can differentiate the type of interactions.

The system structure graph (*SSG*) of CBSS of Figure 2.2 is shown in Figure 2.3. The four nodes represent respective sub-systems – *client side UI* (S_1), *web (application) server* (S_2), *database server* (S_3) and *page generator* (S_4) of CBSS and edges corresponds to the connections/interactions among the sub-systems.

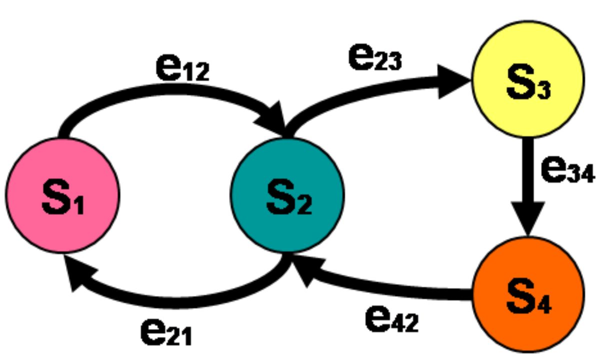

Figure 2.3 Typical component based web application (system structure graph)

These sub-systems of the CBSS are also connected physically or indirectly at the level of their sub-sub-systems/components. The CBSS *SSG* is capable of updating, modifying and deleting of sub-systems or sub-sub-systems based on different design aspects as per real life situation. The developed *SSG* representation, Figure 2.3, is suitable for understanding and visual analysis, but not appropriate for computer processing. If the number of sub-systems is large, then the overall system becomes more complex for understanding and visual analysis. Moreover, changing of labels of nodes results in a new *SSG*. In view of this, a better computer efficient representation is presented. Many matrix representations are available in the literature (Deo, 2004; Upadhyay, 2004)**,** for example, adjacency and incidence matrices.

2.5 Matrix Models

An adjacency matrix representation is used to show the equivalent representation of *SSG*. Variants of adjacency matrices of the *SSG* are defined to find out which matrix is more suitable to represent CBSS. The matrix should be flexible enough to incorporate the structural information of sub-systems and interactions among them. Following sub-sub-section discusses the different type of matrices and the rationale for choosing them to represent graph model (*CBSS SSG*) mathematically.

2.5.1 System Structure Matrix (*SSM*- CBSS)

The adjacency matrix **(**Deo, 2004; Jurkat and Ryser, 1996) of a graph G_S with '*n*' nodes is an '*n*' order symmetric binary (0, 1) square matrix where e_{ij} represents the connectivity/interaction between (sub-) systems *i* and *j* such that:

e_{ij} = 1, if the (sub-) system '*i*' is connected/interacted to the (sub-) system '*j*' and

= 0, otherwise.

However, e_{ii} = 0, as sub-system is not connected to itself. In a case where it is connected to itself e_{ii} = 1. This implies a self-loop at node '*i*' in the graph. It is to be noted that for a directed graph $e_{ij} \neq e_{ji}$.

In the (0, 1) adjacency matrix of the system structure matrix off-diagonal elements e_{ij} in the matrix represent connection between sub-systems S_i and S_j. The adjacency matrix for a graph as shown in Figure 2.3 is given below:

$$
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \quad \text{Sub-systems} \\
A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
\tag{2.1}
$$

The (0, 1) adjacency matrix does not contain properties/attributes characterizing different interactions/ connections between different sub-systems. It only represents the sub-system connectivity. In order to get information about the structural characteristics of the CBSS, the association of variables is done with the elements of adjacency matrix. In order to show connectivity/interconnection/interdependence between different systems '*i*' and '*j*' of the CBSS, let off-diagonal elements be represented by a symbol e_{ij} whose functional value will depend upon type of connection/interaction. Adjacency matrix A = $[a_{ij}]$ will be (0, e_{ij}) instead of (0, 1) matrix. Variable adjacency/system structure matrix (VSSM- *CBSS*), V_A, of the system shown in Figure 2.3 is developed below as:

$$
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \quad \text{Sub-systems} \\
V_A = \begin{bmatrix} 0 & e_{12} & 0 & 0 \\ e_{21} & 0 & e_{23} & 0 \\ 0 & 0 & 0 & e_{34} \\ 0 & e_{42} & 0 & 0 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
\tag{2.2}
$$

The e_{ij} of (VSSM- *CBSS*) apart from representing connectivity also represents influence of structural attribute characteristics of i^{th} sub-system on j^{th} sub-system. For example S_1 sends request to S_2 using HTTP and a change of communication protocol in S_2 from HTTP to HTTPS will have impact on S_1 etc. As this matrix also does not infer anything about the characteristic features of the sub-systems because diagonal elements are zero, a new matrix called '*characteristic system structure matrix*' is defined.

2.5.2 Characteristic System Structure Matrix *(CSSM- CBSS)*

By defining characteristic system structure matrix 'C', realization of the presence of different systems (based upon system structure) can be done. The CBSS characteristic system structure matrix (CSSM- *CBSS*), C = [SI – A], corresponding to the systems graph in Figure 2.3 is given below:

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \quad \textit{Sub-systems} \\
C = & \begin{bmatrix} S & -1 & 0 & 0 \\ -1 & S & -1 & 0 \\ 0 & 0 & S & -1 \\ 0 & -1 & 0 & S \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}
\tag{2.3}
$$

where *I* is the identity matrix and *S* is used as a variable to represent systems characteristic features of the basic structure. This matrix is similar to the characteristic matrix defined in graph theory (Deo, 2004). The characteristic of a *CBSS* (system, sub-system, sub-sub-systems up to the component level considering interactions) can be *reliability*, *maintainability*, *usability* etc. It can be inferred from the matrix that it is capable of representing the presence of systems and interactions among them. It does not include information about the attributes of the connections/interactions among sub-systems. The determinant of CSSM- *CBSS*, Det(C), is called *characteristic system structure polynomial* (CP- s). The CP-s of the matrix is shown below:

$$\text{Det}\,(C) = S^4 - S^2 - S \tag{2.4}$$

The CP-s of the matrix is invariant of the system (Deo, 2004) as it does not change by modifying labeling of systems (vertices) and it is the characteristic of the systems structure. It can be inferred that CSSM- *CBSS* is not an invariant of system, as new matrix can be obtained by changing labels of systems. Also, diagonal elements show that identical systems are present in the basic structure. This is one of the reasons that make the CP-s of CSSM- *CBSS* non-unique and incomplete representation of any real system. It has been identified in the literature (Deo, 2004) that many graphs belong to the same family known as *co-spectral* graphs on the basis of having same CP-s. To present distinct information of different sub-systems and interactions among them, a matrix called a '*variable characteristic system structure matrix'* (VCSSM- *CBSS*) is developed.

2.5.3 Variable Characteristic System Structure Matrix *(VCSSM- CBSS)*

A variable characteristic system structure matrix V_C is defined by taking into consideration distinct characteristics of sub-systems and their interconnections defined by *SSG*. Let the off-diagonal elements matrix S_O consists of e_{ij} rather than 1 to represent interaction/connectivity (system '*i*' is connected to system '*j'*). Let us also define diagonal matrix S_D with its variable diagonal elements S_i (i = 1, 2,…, 4) representing the characteristic structure features of four distinct sub-systems. The VCSSM- *CBSS*, $V_C = [S_D - S_O]$ is written as:

$$V_C = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \; \textit{Sub-systems} \\ \begin{bmatrix} S_1 & -e_{12} & 0 & 0 \\ -e_{21} & S_2 & -e_{23} & 0 \\ 0 & 0 & S_3 & -e_{34} \\ 0 & -e_{42} & 0 & S_4 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \tag{2.5}$$

The determinant of this (VCSSM- *CBSS*) is known as '*variable characteristic multinomial*' and is written as VCM- *CBSS*, the variable characteristic multinomial of the *CBSS*.

$$\mathrm{Det}(V_C) = S_1{*}S_2{*}S_3{*}S_4 - S_1{*}e_{23}{*}e_{34}{*}e_{42} - e_{21}{*}e_{12}{*}S_3{*}S_4 \tag{2.6}$$

The VCM- *CBSS* multinomial contains terms both of positive and negative signs. It is the comprehensive tool for analysis in symbolic form. While calculating VCM- *CBSS* value for CBSS analysis, some information about system, sub-systems, components and their nature of interaction is lost. This is due to the cancellation of some terms because of subtraction operation in the process of computing VCM- *CBSS*. In order to avoid loss of information during structural analysis and structural performance evaluation in critical cases, a new matrix function, which will retain all the multinomial terms with no subtraction operation is defined. This matrix preserves information about the system, sub-systems, components and their interconnectivities (Upadhyay, 2008; Upadhyay et al., 2009).

2.5.4 Variable Permanent System Structure Matrix *(VPSSM-CBSS)*

In order to describe proper characterization of *CBSS* systems as derived from combinatorial considerations, a permanent matrix *P*, is developed. The matrix function/permanent *Per(P)* of VPSSM- *CBSS* is capable of describing whole CBSS system i.e. system graph in a single multinomial equation (Jurkat and Ryser, 1996). Let the complete permanent matrix of four sub-systems of typical component based application with all possible interactions present be defined as:

$$P= \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \quad \textit{Sub-systems} \\ \left[\begin{array}{cccc} S_1 & e_{12} & e_{13} & e_{14} \\ e_{21} & S_2 & e_{23} & e_{24} \\ e_{31} & e_{32} & S_3 & e_{34} \\ e_{41} & e_{42} & e_{43} & S_4 \end{array}\right] \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \tag{2.7}$$

A variable permanent system structure matrix (VPSSM- *CBSS*), $V_p = [S_D + S_O]$, of *SSG* in Figure 2.3 is written as:

$$V_p = \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \quad \textit{Sub-systems} \\ \left[\begin{array}{cccc} S_1 & e_{12} & 0 & 0 \\ e_{21} & S_2 & e_{23} & 0 \\ 0 & 0 & S_3 & e_{34} \\ 0 & e_{42} & 0 & S_4 \end{array}\right] \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \tag{2.8}$$

It is a complete representation of CBSS, as it does not contain any negative sign. This means that it preserves all the structural information about dyads, loops of systems, or system attributes such as *reliability, maintainability,* etc., even in the numerical form. The only difference between VCSSM- *CBSS* and VPSSM- *CBSS* lies in the signs of off-diagonal elements. A unique expression i.e. permanent of VPSSM- *CBSS,* is computed by taking its determinant and converting all negative signs that appear in the determinant expression with positive signs. This expression is known as '*variable permanent CBSS function'* VPF- *CBSS* and is denoted as *Per(V_p)*:

$$\mathrm{Per}(V_p) = S_1{*}S_2{*}S_3{*}S_4 + S_1{*}e_{23}{*}e_{34}{*}e_{42} + e_{21}{*}e_{12}{*}S_3{*}S_4 \tag{2.9}$$

It can be inferred that the terms present in VCM- *CBSS* and VPF- *CBSS* are the same but they differ in the signs. In VCM- *CBSS* terms consist of both positive and negative sign. But VPF- *CBSS* only contains terms of positive sign. The above equation (multinomial) uniquely represents the CBSS of Figure 2.3 irrespective of labeling of sub-systems. Every term of these equations represents a sub-set of the CBSS system. It is possible to write these equations simply by visual inspection of the CBSS system of Figure 2.3 as every term corresponds to a physical sub-system of the complete system. All these distinct combinations of sub-systems and interactions of the macro system are shown graphically in Figure 2.4.

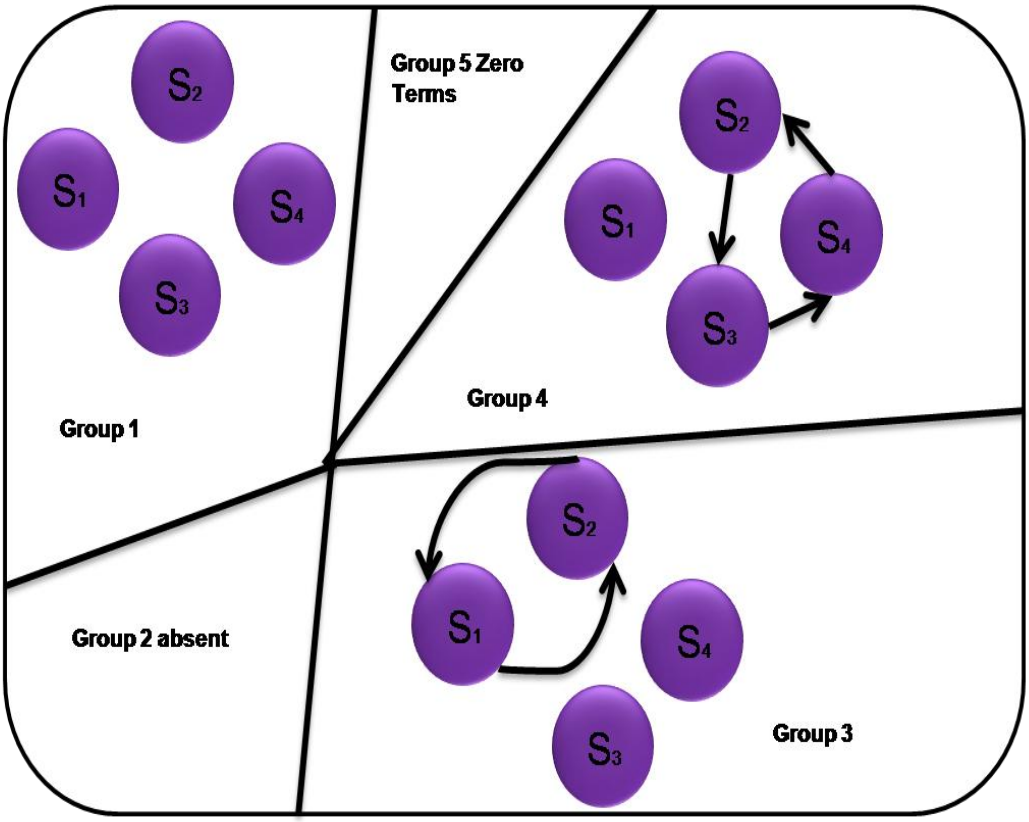

Figure 2.4 Combination of sub-systems and interactions for analysis purpose

The multinomial, i.e., the permanent function when arranged in decreasing order of the group of systems/sub-systems/vertices, S_i, it can be easily seen that arrangement reflects total (N + 1) groups. The first group contains terms with N unconnected S_i*'s***.** Each successive group has one less sub-system than the previous group and rest of the elements are the combination of dyads and loops. The last group does not contain any S_i in its terms. It contains only terms such as e_{ij}^2, $e_{ij}\ e_{jk}\ e_{ki}$, etc. This arrangement helps in identifying different critical components and links to improve quality concerns such as *reliability*, *usability*, *maintainability*, etc., of system. On a critical analysis of permanent function, expression 2.9, it is inferred that this multinomial contains only distinct sub-systems – S_i , dyads - e_{ij}^2 / $e_{ij}\ e_{jk}$ and loops – $e_{ij}\ e_{jk}\ \ldots e_{ni}$. A complete permanent function has been written in a systematic manner for unambiguous and unique interpretation. In short it can be represented as:

$$
\left.
\begin{aligned}
&\text{Per }(V_P) = f\,(S_{i,}\,, e_{ij}^{2}\,, e_{ij}\, e_{jk}\, e_{ki} \text{ etc}) \quad \{\text{ if } e_{ij} = e_{ji}\} \\
&\qquad = f\,(\text{Vertices, dyads, loops}) \quad = f\,(\text{structural components}) \\
&\text{Per }(V_P) = f'\,(S_{i,}\,, e_{ij}e_{ji}\,, e_{ij}\, e_{jk}\, e_{kl}\, e_{li}\,, e_{ij}\, e_{jk}\, e_{kl}\, e_{lm}\, e_{mi}\,) \quad \{\text{ if } e_{ij} \neq e_{ji}\} \\
&\qquad = f'\,(\text{Vertices, 2-vertex loops, loops}) = f'\,(\text{structural components})
\end{aligned}
\right\} \qquad (2.10)
$$

Following are the group specification for the typical component based web application, Figure 2.2, as computed through expression 2.9:

- Group 1: The first term (grouping) represents a set of N unconnected *CBSS* sub-systems, i.e., $S_1, S_2, \ldots, S_n$. For example, if the analysis is carried out for the *CBSS* i.e. typical web application, the first set is

/UI/Web Application Server/Database Server/Page Generator/

A slash helps in separating two entities. These entities can be handled concurrently with the help of concurrent teams.

If the entity 2 i.e. S_2 (web application server) characteristic is considered for failure analysis then an in-depth study may reveal that this attributes to: overloading, deadlock, processing of improper business logic, synchronization, system crash etc. By the application of appropriate techniques, the failure mode and effect of the web application server can be reduced. Using a similar approach, the failure mode and effects of other sub-systems are also considered.

- Group 2: This group is absent as a particular sub-system has no interaction with itself (absence of self-loops) i.e. none of the sub-system is connected to itself.
- Group 3: Each term of the third grouping represents a set of two-element *CBSS* system loops (i.e., $S_{ij}\, S_{ji}$)and shows interaction between sub-system *i* and sub-system *j*. Along with this sub-system measure of the remaining (N-2) unconnected sub-systems are also considered. For example, in current application following set is present:

/(UI Web Application Server)/Database Server/Page Generator/

The above set shows that a combined effect of *UI* and *web application server* as an entity is considered for analysis. This is further extended by incorporating analytical results of remaining unconnected sub-systems.

Group 4: Each term of the fourth grouping represents a set of three-element CBSS sub-system interaction loops ($e_{ij}\ e_{jk}\ e_{ki}$ or its pair $e_{kj}\ e_{ji}$) and the system measure of the remaining (N-3) unconnected elements. In the example this group has one term. Here sub-system S_2, S_3 and S_4 has interactions. Following representation can be seen:

/UI/(Web Application Server Database Server Page Generator)/

- Group 5: The fifth grouping contains two sub-groups. The terms of the first sub-grouping consist of two-element CBSS sub-system interaction loops (i.e., $e_{ij}\ e_{ji}$ and $e_{kl}\ e_{lk}$) and CBSS constituent e_m .The terms in the second grouping are a product of four-element CBSS sub-system interaction loops (i.e., $e_{ij}\ e_{jk}\ e_{kl}\ e_{li}$) or its pair (i.e., $e_{il}\ e_{lk}\ e_{kj}\ e_{ji}$)and CBSS constituent S_m. This group in the example has zero terms.

Different terms of permanent function i.e. expression 2.9 of CBSS system represent different sub-sets of the system, see Figure 2.4. As these terms consist of structural terms S_i, $\boldsymbol{e_{ij}^2}/e_{ij}\ e_{ji}$, $e_{ij}\ e_{jk}\ e_{kl}\ e_{li}/e_{il}\ e_{lk}\ e_{kj}\ e_{ji}$ etc., global CBSS system solution providers offer different alternative solutions for each of these structural sub-systems. If there are *n* distinct terms in the permanent function, there are *n* ways of designing and analyzing the CBSS system. If the system is already in place, the *SWOT* (Strength-Weakness-Opportunities-Threats) analysis can help in improving the existing system. If the designer is using the developed methodology to develop optimum design solution at the conceptual stage, it can be done in the presence of available standard solutions of concerns for e.g. sub-systems (S_i), dyads (e_{ij}^2) and loops ($e_{ij}\ e_{jk}\ e_{kl}\ e_{li}$) etc. Thus the physical representation of Figure 2.4 helps in analyzing and designing CBSS system comprehensively using the structural modular approach. This permanent function and its interpretations become the basis of modular analysis and design of CBSS system.

2.6 Evaluation of V_P

The diagonal elements of the matrix in equation (2.8) correspond to the four sub-systems that constitute a CBSS system (*component based web application*). The values of these diagonal elements $S_1, S_2 \ldots S_4$ are calculated as:

$$S_1 = \mathrm{Per}(V_P S_1) \qquad S_2 = \mathrm{Per}(V_P S_2) \qquad S_3 = \mathrm{Per}(V_P S_3) \qquad S_4 = \mathrm{Per}(V_P S_4) \tag{2.11}$$

where, $V_P S_1$, $V_P S_2$, $V_P S_3$, and $V_P S_4$ are the variable permanent matrices for four sub-systems of the *CBSS* system. The procedure for calculating values of $V_P S_1$, $V_P S_2 \ldots V_P S_4$ is the same as for calculating Per (V_p) of expression 2.9. For this purpose, the sub-systems of CBSS system are considered, and the procedure given below is followed:

- ***Step 1***: The schematics of these sub-systems are drawn separately by considering their various sub-sub-systems up to the component level.
- ***Step 2***: The degree of interactions, interconnections, dependencies, connectivity, etc., among different sub-sub-systems are identified.
- ***Step 3***: Repeat steps 1 and 2 until the component level is achieved.

Digraph representations (like Figure 2.3) of four sub-systems are first drawn separately to obtain their matrix equations (like equation (2.8)) i.e. $V_p S_i$ and then their permanent functions Per($V_P S_i$), S_i, $i = 1,\ldots,4$ are obtained. The off-diagonal terms e_{ij} ($i, j = 1,2,\ldots,4$) of matrix equation (2.8) gives the interactions between the systems S_i and S_j . Depending upon the type of structural analysis, system can be represented as multinomial, graph and matrix or by some analytical model. To get the exact degree of interactions, interconnections, dependencies, connectivity, etc. between sub-systems or sub-sub-systems one has to consider the views of technical team experts. The final decision on the values of S_i and interactions may be taken on the recommendations of the team.

2.7 Variable Permanent Structural Quality Matrix (VPSQM-CBSS)

On the lines of equation 2.9 various quality matrices of CBSS can be developed and analyzed simultaneously. The quality of CBSS is defined as:

$$Q(\text{CBSS}) = f\,(\text{ quality of its sub-systems considering all levels of interactions}) \tag{2.12}$$

For considering reliability of CBSS, equation 2.9 can be transformed to

$$V_{pR} = \begin{array}{c} \begin{array}{ccccl} 1 & 2 & 3 & 4 & \textit{Sub-systems} \end{array} \\ \begin{bmatrix} R_1 & r_{12} & 0 & 0 \\ r_{21} & R_2 & r_{23} & 0 \\ 0 & 0 & R_3 & r_{34} \\ 0 & r_{42} & 0 & R_4 \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \qquad (2.13)$$

$$\mathrm{Per}(V_{pR}) = R_1{*}R_2{*}R_3{*}R_4 + R_1{*}r_{23}{*}r_{34}{*}r_{42} + r_{21}{*}r_{12}{*}R_3{*}R_4 \qquad (2.14)$$

In equation (2.13) diagonal elements represents *reliability* of sub-systems and off diagonal elements represents *reliabilities* of interconnections. Similarly other quality characteristics such as *usability, performance, maintainability* etc., can be transformed and considered respectively. Permanent expression, equation (2.14), of the quality characteristics can be developed based upon the method presented in section 2.5.4 (see equation (2.9)). Likewise equation (2.14) permanent expression of other quality characteristics can be developed. By putting values of diagonal and off-diagonal elements in quality permanent expressions such as equation (2.14) respective characteristics ***index*** can be computed, for example by putting *reliabilities* of the elements in equation (2.14), *reliability* ***index*** of complete ***CBSS*** can be computed. This can be compared with other alternative design solutions and then later ranked on ***index*** value. It is to be noted that the developed methodology will also allow designers, quality analysts to consider all quality characteristics concurrently. This can be achieved by transforming equation (2.8) by putting quality characteristics in its diagonal positions and their relative importance or trade off measures in its off diagonal positions. Let the ***CBSS*** quality characteristics of interest be – Reliability (Q_1), Usability (Q_2) and Maintainability (Q_3). For concurrent consideration these characteristics will be represented as equation (2.15) where diagonal elements represents three characteristics and off-diagonal elements represent relative importance of one characteristic over other characteristics. Each quality characteristic value can be computed on the lines of equation (2.13) and respective permanent expression, like equation (2.14), can be computed to get the ***Index*** value. This index value will be placed in respective diagonal position in equation (2.15). By computing the permanent expression of equation (2.15) quality ***Index*** can be generated.

$$V_{pQ} = \begin{array}{c} \begin{array}{ccc} Q_1 & Q_2 & Q_3 \end{array} \text{ Characteristic} \\ \begin{bmatrix} Q_1 & q_{12} & q_{13} \\ q_{21} & Q_2 & q_{23} \\ q_{31} & q_{32} & Q_3 \end{bmatrix} \begin{array}{c} Q_1 \\ Q_2 \\ Q_3 \end{array} \end{array} \tag{2.15}$$

Thus it can be seen that developed methodology is a powerful tool on the hands of designers, analysts, developers to take early decisions on the quality of CBSS and also compare and select alternative designs.

2.8 Generalization of Methodology

Suppose a system consists of N sub-systems (due to involvement of N-tier architecture, vertical/horizontal/diagonal extension) in place of the developed four sub-systems then the most general way of matrix representation is shown below. This matrix is also known as the '*general variable permanent matrix*' (VPGSSM- *CBSS*) corresponding to the N sub-systems.

$$\begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & . & . & . & . & N \end{array} \text{ Sub-systems} \\ \begin{bmatrix} S_1 & e_{12} & e_{13} & . & . & . & . & e_{1N} \\ e_{21} & S_2 & e_{23} & . & . & . & . & e_{2N} \\ e_{31} & e_{32} & S_3 & . & . & . & . & e_{3N} \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ e_{N1} & e_{N2} & e_{N3} & . & . & . & . & S_N \end{bmatrix} \begin{array}{c} 1 \\ 2 \\ 3 \\ . \\ . \\ . \\ . \\ N \end{array} \end{array} \tag{2.16}$$

Permanent for the above matrix, i.e., $Per(V_{PN})$is called '*variable permanent General function*' (VPGSSF- *CBSS*).The general permanent function, VPF_N, for the above matrix is written in sigma form as:

$Per(V_{PN})$ =

$$\prod_{x=1}^{N} S_x + \sum_i \sum_j \sum_k \sum_l \ldots \sum_N \left(e_{ij} e_{ji}\right) S_k S_l \ldots S_N + \sum_i \sum_j \sum_k \sum_l \ldots \sum_N \left(e_{ij} e_{jk} e_{ki} + e_{ik} e_{kj} e_{ji}\right) S_l S_m \ldots S_N$$

$$+ \left(\sum_i \sum_j \sum_k \sum_l \ldots \sum_N \left(e_{ij} e_{jk}\right)\left(e_{kl} e_{lk}\right) S_m \ldots S_N + \sum_i \sum_j \sum_k \sum_l \ldots \sum_m \left(\left(e_{ij} e_{jk} e_{kl} e_{li}\right) + \left(e_{il} e_{lk} e_{kj} e_{ji}\right) \right) S_m \ldots S_N \right)$$

$$+ \sum_i \sum_j \sum_k \sum_l \ldots \sum_N \left(\begin{array}{l} e_{ij} e_{ji} \left(e_{kl} e_{lm} e_{mk} + e_{km} e_{ml} e_{lk}\right) S_n S_o \ldots S_N + \\ \sum_i \sum_j \sum_k \sum_l \ldots \sum_m \left(e_{ij} e_{jk} e_{kl} e_{lm} e_{mi} + e_{im} e_{ml} e_{lk} e_{kj} e_{ji}\right) S_n S_o \ldots S_N \end{array} \right)$$

$$+ \ldots \tag{2.17}$$

The number and composition of groups and sub-groups will be the same as discussed earlier. So it is possible to write the permanent function of any CBSS system in (***N + 1***) groups. It may be noted that a permanent function will contain $N!$ terms only, provided e_{ij} are not 0. In certain cases, designers and/or developers team may decide that some of e_{ij} are 0 because of insignificant influence of one sub-system over the other sub-system. Substitutions of corresponding e_{ij} equal to 0 in general permanent function (equation (2.16)) or in general VPF_N (expression 2.17) gives the exact number of terms with the modified permanent function.

2.9 Structural Identification and Comparison of CBSS - Web Applications

Typical component based web application architecture is represented as a system consisting of four sub-systems, which affects properties and performance of finished CBSS. This four sub-system CBSS is modeled as a multinomial i.e. a permanent function. Different CBSS web applications developed using different sub-systems and technologies will have a different number of terms in different groups and sub-groups of their permanent functions because of change in structure and interactions. The similarity or dissimilarity in the structure between two CBSS systems is obtained by comparing their permanent functions. Using the methodology, the identification of CBSS system architecture and its comparison with other CBSS system architecture can be based on the analysis carried out with the help of VPF-CBSS. Two CBSS system architectures are similar from sub-systems and their interactions point of view only if their digraphs are isomorphic. Two CBSS system architecture digraphs are isomorphic if they have identical VPF- *CBSS*. This means that the set of number of terms in each grouping/sub-grouping of two CBSS systems is the same. Based on this, a CBSS identification set for any complete CBSS is written as:

$$\left[\left(M_1 / M_2 / M_3 / M_4 / M_{51} + M_{52} / M_{61} + M_{62} / \ldots\right)\right] \tag{2.18}$$

Where, M_i represents the structural property of a system. It can be interpreted as the total number of terms in i^{th} grouping, M_{ij} represents the total number of terms in the j^{th} sub-group of i^{th} grouping. In case there is no sub-grouping, the M_{ij} is the same as M_i ; the sub-groupings are arranged in the decreasing order of size (i.e., number of elements in a loop). In general, two CBSS products may not be isomorphic from the viewpoint of architecture of sub-systems and interactions among sub-systems. A comparison is also carried out on the basis of the coefficient of similarity. The coefficient is derived from the structure, i.e., VPF-CBSS and it compares two CBSS products or a set/family of CBSS products on the basis of similarity or dissimilarity. If the value of distinct terms in the j^{th} sub-grouping of the i^{th} grouping of VPF- *CBSS* of two CBSS products under consideration are denoted by M_{ij} and M'_{ij}, then the following criteria is proposed (Liu et al., 2004, Upadhyay et al., 2009): The coefficient of similarity and dissimilarity are calculated using number of terms only.

Criterion 1: The coefficient of dissimilarity C_{d-1} is defined as:

$$C_{d-1} = \frac{1}{Y_1} \sum_i \sum_j \psi_{ij} \tag{2.19}$$

Where, $Y_1 = \max\left[\sum_i \sum_j \left| M_{ij} \right| \ and \ \sum_i \sum_j \left| M'_{ij} \right| \right]$

When sub-groupings are absent $M_{ij} = M_i \ and \ M'_{ij} = M'_i \ and \ \psi_{ij} = \left| M_{ij} - M'_{ij} \right|$ when the sub-groupings exists and $\psi_{ij} = \left| M_i - M'_i \right|$, when the sub-groupings are absent.

Criterion 1 is based on the sum of the difference in number of terms in different sub-groups and groups of VPF- *CBSS* of two structurally distinct *CBSS* architecture. There may be a case when some $\sum_i \sum_j \psi_{ij}$ is zero though two systems are structurally different. This situation may arise when some of the differences are positive while some other differences are negative such that $\sum_i \sum_j \psi_{ij}$ becomes zero. To improve the differentiating power, another criterion is defined.

Criterion 2: The coefficient of dissimilarity C_{d-2} is defined as:

$$C_{d-2} = \frac{1}{Y_2} \sum_i \sum_j \psi'_{ij} \tag{2.20}$$

Where $Y_2 = \max\left[\sum_i \sum_j \left(M_{ij}\right)^2 \; and \; \sum_i \sum_j \left(M'_{ij}\right)^2 \right]$

When sub-groupings are absent $M_{ij} = M_i \; and \; M'_{ij} = M'_i \; and \; \psi'_{ij} = \left|M_{ij}^2 - M'^2_{ij}\right|$ when the sub-groupings exists and $\psi'_{ij} = \left|M_i^2 - M'^2_i\right|$, when the sub-groupings are absent.

Criterion 2 is based on the sum of the squares of the difference in number of terms in different sub-groups and groups of VPF- *CBSS* of two structurally distinct CBSS architectures. It shows that ψ'_{ij} (criterion 2) is much larger than ψ_{ij} (criterion 1). To further increase the differentiating power another criterion 3 is defined.

Criterion 3: The coefficient of dissimilarity C_{d-3} based on criterion 1 is defined as:

$$C_{d-3} = \left[\frac{1}{Y_3} \sqrt{\sum_i \sum_j \psi_{ij}} \right] \tag{2.21}$$

Where ψ_{ij} is the same as described in criterion 1 and

$$Y_3 = \max\left[\sqrt{\sum_i \sum_j \left|M_{ij}\right|} and \sqrt{\sum_i \sum_j \left|M'_{ij}\right|} \right]$$

When sub-groupings are absent $M_{ij} = M_i \; and \; M'_{ij} = M'_i$. Criterion 3 is derived from criterion 1.

Criterion 4: The coefficient of dissimilarity C_{d-4} based on criterion 2 is defined as:

$$C_{d-4} = \left[\frac{1}{Y_3} \sqrt{\sum_i \sum_j \psi'^2_{ij}} \right] \tag{2.22}$$

Where ψ'_{ij} is the same as described in criterion 2 and

$$Y_4 = \max\left[\sqrt{\sum_i \sum_j \left(M_{ij}\right)^2} \; and \; \sqrt{\sum_i \sum_j \left(M'_{ij}\right)^2} \right]$$

When sub-groupings are absent $M_{ij} = M_i$ *and* $M'_{ij} = M'_i$. Criterion 4 is derived from criterion 2. This can further increase the differentiating power. Using the above equations the coefficient of similarity is given as

$$C_{s-1} = 1 - C_{d-1}; C_{s-2} = 1 - C_{d-2}; C_{s-3} = 1 - C_{d-3}; C_{s-4} = 1 - C_{d-4} \tag{2.23}$$

where C_{s-1}, C_{s-2}, C_{s-3} and C_{s-4} are the coefficient of similarity between two CBSS architectures under consideration based on criterion 1, criterion 2, criterion 3 and criterion 4.

Using above-mentioned criteria, comparison of two or family of CBSS system architectures is carried out. Two CBSS architectures are isomorphic or completely similar from a structural point of view, if structural identification set for the two systems are exactly the same. This means the number of terms/ items in each grouping/ sub-grouping are exactly the same. The structural identification set equation (2.18) for the system shown in Figure 2.2 is obtained by considering its structure graph, Figure 2.3, and VPF- *CBSS* as /1/0/1/1/0/.

It may be noted that the coefficient of similarity and dissimilarity lies in the range between 0 and 1. If two *CBSS* architectures are isomorphic or completely similar, their coefficient of similarity is 1 and the coefficient of dissimilarity is 0. Similarly, if two *CBSS* architectures are completely dissimilar, their coefficient of similarity is 0 and the coefficient of dissimilarity is 1. Based on aforementioned criteria for comparing different *CBSS* a systems structure index (I_{SS}) is defined as:

I_{SS}: f (Coefficient of Similarity or Dissimilarity) (2.24)

2.10 Case Study - CBSS Web Application

A variation of typical component based web application is considered to validate the developed methodology. The new *component based web application architecture/structure* is shown in Figure 2.5. Due to the rise of users for accessing data, existing web application server capability is extended by adding another web (application) server. The dashed line in Figure 2.5 represents enabling of second web server path to access the response for the client request. The respective digraph for Figure 2.5 is shown in Figure 2.6.

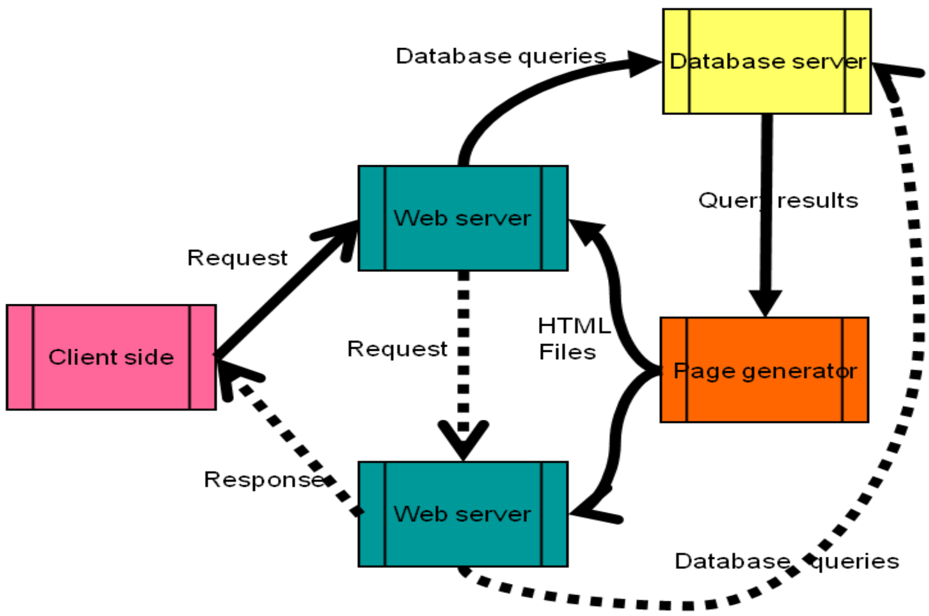

Figure 2.5 Variation of typical component based web application

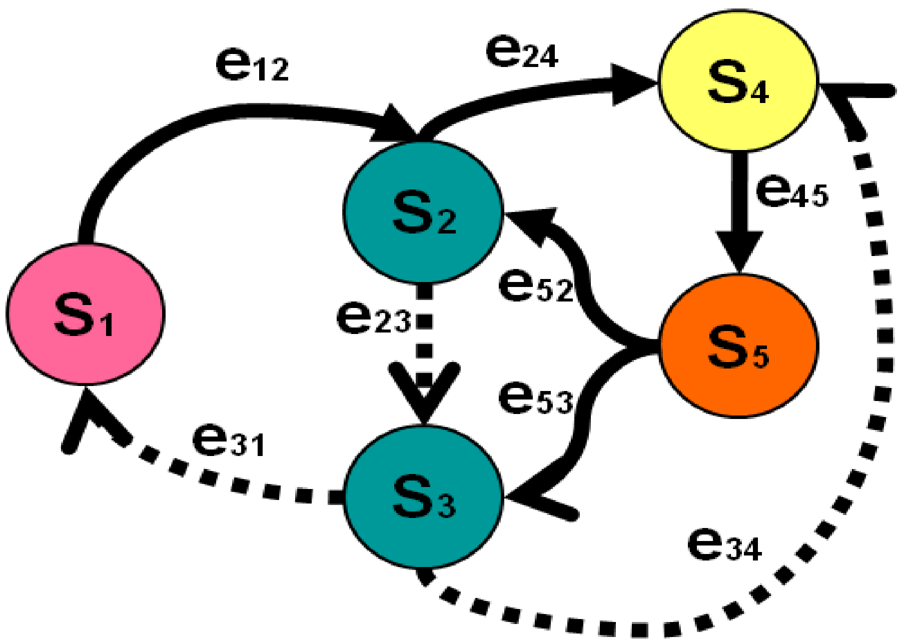

Figure 2.6 Equivalent digraph model for Figure 2.5

The permanent matrix for the Figure 2.6 is written as follows:

$$
\begin{array}{cc}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} & \textit{Sub-systems} \\
V_{p'} = \begin{bmatrix} S_1 & e_{12} & 0 & 0 & 0 \\ 0 & S_2 & e_{23} & e_{24} & 0 \\ e_{31} & 0 & S_3 & e_{34} & 0 \\ 0 & 0 & 0 & S_4 & e_{45} \\ 0 & e_{52} & e_{53} & 0 & S_5 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\end{array}
\qquad (2.25)
$$

The permanent function for the above matrix is written as:

$$
\begin{aligned}
Per(V_{p'}) = {} & S_1*S_2*S_3*S_4*S_5+e_{34}*e_{45}*e_{53}*S_1*S_2 + e_{24}*e_{45}*e_{52}*S_1*S_3 + e_{12}*e_{23}* e_{31}*S_4*S_5 \\
& + e_{23}*e_{34}*e_{45}*e_{52} *S_1 +e_{31}*e_{12}*e_{53}*e_{24}*e_{45}
\end{aligned}
\qquad (2.26)
$$

Based on equation (2.26), the structure identification for the above permanent function can be written as /1/0/0/3/1/1/. Table 2.1 shows description of terms of various grouping and sub-groupings for both the CBSS web application.

Group No.	1st ***CBSS Web Application*** **Architecture**	**2nd *CBSS Web Application* Architecture**
1	1	1
2	0	0
3	1	0
4	1	3
5	0	1
6	-----	1
Total	3	6

Table 2.1 Structural identification set

The coefficients of similarity/dissimilarity for these two systems are calculated by using four different criteria and are given below, Table 2.2:

Criterion	Coefficient of dissimilarity	Dissimilarity index value of 1st *CBSS Web Application* Architecture *w.r.t* 2nd *CBSS Web Application* Architecture	Coefficient of similarity (index value)
1	$C_{d\text{-}1}$	0.83	$C_{s\text{-}1}$ = 1-$C_{d\text{-}1}$ = 0.17
2	$C_{d\text{-}2}$	0.91	$C_{s\text{-}2}$ = 1-$C_{d\text{-}2}$ = 0.09
3	$C_{d\text{-}3}$	0.92	$C_{s\text{-}3}$ = 1-$C_{d\text{-}3}$ = 0.08
4	$C_{d\text{-}4}$	0.95	$C_{s\text{-}4}$ = 1-$C_{d\text{-}4}$ = 0.05

Table 2.2 Coefficient of dissimilarity and similarity (index value)

This shows that criterion 4 has much larger value (coefficient of dissimilarity index value) as compared to criterion 1, 2 and 3. This demonstrates larger differentiating capacity of criterion 4 over criterion 1, 2 and 3. After comparing these two system structure graphs, it is found that the second graph has one more node as compared to first system graph and few more interactions among nodes (in order to achieve the functionality). These additions cause a major change in the structural complexity, which is directly reflected in the similarity/dissimilarity coefficient as calculated.

It has been shown by a number of researchers that quality of any system is dependent on its architecture/structure consisting of its structural components and interactions among them.

2.11 Step-by-Step Procedure

The step-by-step procedure is mentioned which can permit industry, research community and other stakeholders to modify, extend and improve quality of their CBSS products. This methodology will also help in identifying various choices of available designs depending upon interaction/interdependencies between systems and their sub-systems and so on. Various marketing and strategic decisions can also be taken as per the competitiveness of CBSS products in the global market. It will also give an insight to researchers, designers and developers to identify, select and create critical systems integration process. A generalized procedure for the complete design and analysis of CBSS system architecture is summarized below:

- ***Step 1***: Consider the desired CBSS product. Study the complete CBSS system and its sub-systems, and also their interactions.
- ***Step 2***: Develop a block diagram (schematics) of the CBSS system as per the concept shown in Figure 2.2, considering its sub-systems and interactions along with assumptions, if any.
- ***Step 3***: Develop a systems structure graph of the CBSS system on the lines of Figure 2.7 with sub-systems as nodes and edges for interconnection between the nodes.
- ***Step 4***: Develop the matrix equation as per the concept mentioned in equation (2.8) and multinomial representations as per the expression (2.9) of CBSS system.
- ***Step 5*:** Evaluate functions/values of diagonal elements from the permanent functions of distinct sub-systems equation based on the lines of the equation (2.11) of the CBSS and repeat Steps 2 – 4 for each sub-system.
- ***Step 6***: Identify the functions/values of off-diagonal elements/interconnections at different levels of hierarchy of the CBSS amongst systems, sub-systems, sub-sub-systems, etc.
- ***Step 7***: Calculate CBSS identification set as per the concept defined in the equation (2.18). Carry out architectural similarity and dissimilarity with potential candidates to take appropriate decisions on the lines of the equation (2.24).
- ***Step 8***: Carry out modular design and analysis of CBSS products while purchasing off the shelf from the global market.

The values (or functions) of interactions e_{ij} (i ,j = 1,2,…,n) between different sub-systems $S_1,S_2,...,S_n$ can be written as a multinomial or a matrix, depending upon the type of interaction/reaction between the two sub-systems. The sub-sub-systems can again be treated as systems, as every sub-sub-system is a system in itself. Following the above procedure, these sub-systems can be broken down into sub-sub-systems and different graphs, matrices, and permanent representations can be obtained. Depending upon the depth of the required analysis, the process could be taken to the constituent level. In certain cases, it may be possible to evaluate e_{ij} *'s* experimentally or using available mathematical models. With the help of this data, complete multinomial for the CBSS system can be evaluated. Using/available standard modules of CBSS architectural sub-systems (e.g. dyads and loops of different sub-systems) in the global market, designers can develop alternative designs of CBSS products and carry out analysis and improvement of existing CBSS products.

2.12 Usefulness of the Developed Methodology

Different stakeholders in the component based software development project are benefited by the developed methodology. The methodology is dynamic in nature as sub-systems/components and interactions, which appear as variables in different models may be changed without any difficulty. It also helps to develop a variety of CBSS systems providing optimum quality characteristics under different industrial/organizational component based applications. Thus, the approach helps to express the CBSS system in quantitative terms, which has more often been expressed in qualitative terms. The procedure also helps to compare different CBSS systems in terms of its characteristics and rate them for particular application domain. It is hoped that this methodology will provide a new direction in the research attempts towards global projects of quantitative structure activity relationship and quantitative structure properties relationship. The present work is an attempt towards the development of complete methodology for virtual integration (Choi and Chan, 2004) of CBSS components/sub-system as well as virtual design of complete CBSS system architecture. The developed methodology is a powerful tool in the hands of system analysts, designers, decision makers and developers. Using this and morphological chart/tree, system analysts, decision makers and designers can generate alternative design solutions and select the optimum one. Similarly, this method can be exploited to improve quality and reduce cost and time-to-market in the software industry. It is also possible to exploit the methodology to extend the useful product life in the software industry market by making strategic changes in the CBSS systems architecture. This methodology gives a comprehensive knowledge to the user about CBSS systems architecture and helps in the selection of right sub-system or component at the right time and at right cost from the global market. Component designers and developers will also get to know about the critical parameters that need to be taken care of in order to improve the quality of components.

2.12 Concluding Remarks

In this chapter, a methodological framework is developed using system methodology, CE principles and graph theoretic approach to model and analyze component based software system. The framework helps in representing CBSS structural information, including its sub-systems, their sub-sub-systems (up to component level) and their interconnections concurrently. The methodology also allows to compare two or family of CBSS on the basis of system’s characteristics expression.

The methodological framework developed in this chapter will be used in carrying out quality analysis, design and evaluation of software components which will yield good quality CBSS**.** From chapter 3 to chapter 6 the methodology is exploited for identifying *usability, maintainability* and *reliability* characteristics of a software component by concurrently considering its respective sub-characteristics and attributes. Also chapter 7 focuses on building of concurrent framework for considering different quality characteristics concurrently. The applicability of the developed methodology is demonstrated with the help of case studies. It is expected that in order to achieve high quality CBSS a repetitive process of the developed methodological framework must run for each sub-system using bottom up procedure starting from the components.