AFRICAN JOURNAL OF COMPUTING & ICT
(IEEE NIGERIA COMPUTER CHAPTER)
ISSN 2006-1781

Vol.1, No.2, December 2008

CONTENTS

• Editorial Board

• Editor – in- Chief’s Introduction by Dele Oluwade

• On Set-Theoretic Fundamentals of Systems Integration by Adedeji Badru & O. Ibidapo-Obe

• Computational Analysis of GSM Security Algorithm by O.C. Abikoye & Dele Oluwade

• An Evaluation of Chidamber and Kemerer’s Lack of Cohesion in Method (LCOM) Metric Using Different Normalization Approaches by Ezekiel Okike & Adenike Ososian

• A Comparison of Fixed Cellular and GSM Telecommunication Networks by Oluemisi Ajibose & Dele oluwade

• On Some Combinatorial Results of Collapse and Properties of Height in Full Transformation Semigroups by A.O. Adeniji & S.O. Makanjuola

• Biodata of Authors in This Issue

• General Interest News

• Instruction to Authors, Subscribers and Advertizers

Editor-in-Chief: Dele Oluwade
**EDITORIAL BOARD**

The editorial board members are described below. Their respective areas of primary interest are also indicated.

**EDITOR – IN – CHIEF**

Dele Oluwade PhD, Senior Member IEEE
Federal University of Technology, Minna
E-mail: oluwade@sigmaxi.net, bamidele@aims.ac.za
(Computer Science & Information Technology; Discrete Modelling)

**(ASSOCIATE) EDITORS**

A.M. Awobode, PhD
University of Illinois at Urbana Champagne
Email: awobodea@yahoo.com
(Physics of Computation)

E.O. Ayoola, PhD
University of Ibadan
Email: eoayoola@ictp.it
(Mathematics of Computing; Numerical Analysis & Scientific Computation)

Emmanuel Oluremi Bejide PhD, MIEEE
University of Cape Town
Email: ebejide@ebe.uct.ac.za
(Communications Engineering)

Amos David, PhD
Universite Nancy 2
Email: Amos.David@loria.fr
(Computer Science; Strategic Information & Economic Intelligence Systems)

Khalil Ezzinbi, PhD
Cadi Ayyad University
Email: ezzinbi@ucau.ac.ma
(Mathematics of Computing)

A.O. Fakolujo PhD, Senior Member IEEE
University of Ibadan
Email: ao.fakolujo@mail.ui.edu.ng
(Communications Engineering)

Oliver E. Osuagwu PhD, MIEEE
Federal University of Technology, Owerri
Email: drosuagwu@yahoo.com
(Computer Science and Engineering; Data Communications & Network Engineering)

A.S. Ukpera, PhD
Obafemi Awolowo University, Ile-Ife
Email: aukpera@oaife.edu.ng
(Mathematics of Computing; Modelling)

Abdullahi Umar, PhD
King Fahd Univ. of Petroleum & Minerals
Email: aumar@kfupm.edu.sa
(Mathematics of Computing; Computer Algebra)

J. Michel Tchuendue, PhD
University of Tanzania
E-mail: jmt_biomaths@yahoo.com
(Mathematics of Computing; Mathematical Biology)

Submission of a manuscript and other editorial correspondence should be sent to the editor-in-chief electronically. Alternatively, it may be submitted through any other member of editorial board. In the latter case, the editor-in-chief should be copied for record purposes. Please see the inside back cover for details of Instruction to Authors, Subscribers and Advertisers.

**Editorial Assistant:** Ramatu Abdulsalam
(ramatayo@yahoo.com)
When the first edition of *African Journal of Computing and ICT* (i.e Vol.1, No.1) was published in June 2008, it was with the aim of contributing to scholarship in Africa via the dissemination of topical and relevant research papers within the field of computing and allied areas. It is gladdening to note that the journal has been receiving tremendous response from potential authors from Africa and abroad which tend to justify its establishment.

Although detailed guide to authors (as well as to subscribers and advertisers) is provided in the inner back cover, the journal house policy would be subject to review from time to time to accommodate local peculiarities, though a high level of standard would always be maintained. For instance, in lieu of the publication charge of NGN9,500, an author may be required to pay an assessment fee of NGN3,000 before the reviewers’ comments are sent, and a balance of NGN6,500 if/when his paper is eventually accepted. Also, if a review paper is submitted, then it must be particularly significant, topical and be written in a captivating style. The paper should not merely be a summary but should, among others, point out the way forward. The journal would be interested in publishing results of masters and doctorate studies/theses. As much as possible, the supervisor(s) and/or examiner(s) of the project/thesis would be required to be at least one of the reviewers. Such a paper would be indicated by the phrase ‘Thesis Communication’ on its first page. Rejoinders on any paper in the journal would always be welcome and published. In this edition, five papers are published. The first paper by the duo of Professor Adedeji Badru and Professor O. Ibidapo-Obe presents an insightful discourse on the importance and relevance of set theory to systems science and engineering integration. The paper is entitled ‘On Set-Theoretic Fundamentals of Systems Integration’. In ‘Computational Analysis of GSM Security Algorithm’, the authors, O.C. Abikoye and Dr. ‘Dele Oluwade studied the computational complexity of a combined user authentication and encryption key generation GSM algorithm and showed that the time complexity of the algorithm is constant. This paper is part of the Master in Computer Science thesis of the first author written under the supervision of the second author. The paper by Dr. Ezekiel Okike and Professor Adenike Osofisan is on software engineering measurement whereby the authors evaluated the lack of cohesion in method (LCOM) metric due to Chidamber and Kemerer. The paper contributed by Oluyemisi Ajibose and Dr. ‘Dele Oluwade presents a comparative examination of the fixed (wired and wireless) cellular network and GSM
telecommunication, especially in view of the tremendous recent growth of the latter in Africa. This paper is part of the Master in Computer Science thesis of the first author written under the supervision of the second author. The last paper is on discrete structures with focus on semigroups. In the paper, the authors A.O. Adeniji and Dr. S.O. Makanjuola presented some results on collapse and properties of height in full transformation semigroups.

The journal is grateful to the following intellectuals who assisted in reviewing submitted papers: Kunle Adegoke (Dr), R.A. Siyanbola (Dr), Amos David (Professor), Kazeem Gbolagade, Natawo Y. Goshwe (Engr), Oluwole Makinde (Professor), Wole Olatokun (Dr), Dele Oluwade (Dr), Olufade F.W. Onifade, Oliver Osuagwu (Professor) and Seyitan Osunade (Engr).
ON SET-THEORETIC FUNDAMENTALS OF SYSTEMS INTEGRATION

Adedeji Badru
Department of Systems & Engineering Management,
Graduate School of Engineering,
Air Force Institute of Technology,
Dayton, Ohio,
USA
Email: deji@badiru.com

&

O. Ibidapo-Obe
Department of Systems Engineering,
Faculty of Engineering,
University of Lagos,
Nigeria
Email: oyeibidapoobe@yahoo.com

Received 31 July, 2008
Communicated by Dele Oluwade

ABSTRACT

In spite of the much touted need to achieve systems integration, very little has been accomplished in practice. This is often so because there is no analytical basis for identifying and modeling the parameters that the sub-systems have in common. Consequently, attempts to achieve systems integration degenerates to mere conceptual formulations. Sometimes the conceptual formulations work, but most times they do not. The premise of this paper is that a rigorous set-theoretic formulation can help identify the dominant common elements of subsystems, thereby facilitating practical implementation that is amenable to quantitative optimization. The paper explores fundamental set theory and principles for modeling systems interactions.

Keywords: Systems Integration, Set Theory, Mathematical Modeling, Sub-system Analysis, Surface Project Integrals, Quantitative optimization
0. INTRODUCTION

In order for two systems to be integrated, they must overlap in scope or performance with respect to time, space, or physics; subsequently, this requires mathematical representations for manipulating the integrated parameters of the systems. In spite of the much touted need to achieve systems integration, very little has been accomplished in practice. This is often because there is limited analytical basis for modeling the parameters that the sub-systems have in common. Consequently, attempts to achieve systems integration degenerates to mere conceptual formulations. Sometimes the conceptual formulations work, but most times they do not. Omicini and Papadopoulos (2001) present systems integration as the coordination and management of interactions among entities of a system. The premise of this paper is that a rigorous set-theoretic formulation can help identify the dominant common elements of subsystems, thereby facilitating practical implementation that is amenable to quantitative optimization. The paper explores fundamental set theory and principles for modeling system interactions. A system is often defined as a collection of interrelated elements working together in synergy to produce a composite output that is greater than the sum of the individual outputs of the components. Systems integration is a desired state whereby sub-systems of the system are in proper consonance of operation and input-output interactions. Methodologies pertaining to the various characteristics of integrated systems are available in the literature. As a practical application example, Balkhi (1999) addressed integrated inventory systems with time-varying demand. The approach presented in this paper differs from those existing in the literature by focusing on set-theoretic modeling.

0.1 Theory of Sets and Systems

Set theory is the mathematical concept that governs sets, which represent collections of objects. Sets are defined by the elements contained in the collection and the respective memberships of the elements in the sets. Set theory provides the formalism in which mathematical objects are described. This makes set theory suitable for defining systems integration. In this conception, sets and set membership are fundamental concepts describing points and lines in Euclidean geometry. The properties of sets essential for modeling systems integration include:

- 1. Ordered Pairs
- 2. Relations
- 3. Functions
- 4. Set Cardinality

In this paper, standard set and matrix notations and symbols that are generally available in the literature are used for the mathematical formulations presented.

0.1.1 Ordered Pairs on Sets

The notion of ordered pair suggests that if \( a \) and \( b \) are sets, then the unordered pair \( \{ a, b \} \) is a set whose elements are exactly \( a \) and \( b \). The “order” in which \( a \) and \( b \) are put together plays no role. Thus, \( \{ a, b \} = \{ b, a \} \). For many applications, we need to pair \( a \) and \( b \) in a way making possible to “read off” which set comes...
“first” and which comes “second.” We denote this ordered pair of a and b by \((a, b)\); \(a\) is the first coordinate of the pair \((a, b)\), \(b\) is the second coordinate. To assess systems integration properties, the ordered pair has to be a set. It should be defined in such a way that two ordered pairs are equal if and only if their first coordinates are equal and their second coordinates are equal. This guarantees in particular that \((a, b) \neq (b, a)\) if \(a \neq b\). With this understanding, we can distinguish systems integration from the perspective of “push system” or “pull system” because they may not necessarily be identical.

**Basic definition of ordered pairs:** The basic definition of ordered pair, generally available in the literature specifies that \((a, b) = \{\{a\}, \{a, b\}\}\). If \(a \neq b\), \((a, b)\) has two elements, a singleton \(\{a\}\) and an unordered pair \(\{a, b\}\). We find the first coordinate by looking at the element of \(\{a\}\). The second coordinate is then the other element of \(\{a, b\}\). If \(a = b\), then \((a, a) = \{\{a\}, \{a, a\}\} = \{\{a\}\}\) has only one element. In any case, it seems obvious that both coordinates can be uniquely “read off” from the set \((a, b)\).

**Ordered Pair Theorem:** \((a, b) = (a', b')\) if and only if \(a = a'\) and \(b = b'\).

**Proof.** If \(a = a'\) and \(b = b'\), then, of course, \((a, b) = \{\{a\}, \{a, b\}\}\) = \(\{\{a'\}, \{a', b'\}\}\) = \((a', b')\). The other implication is more intricate. Let us assume that \(\{\{a\}, \{a, b\}\}\) = \(\{\{a'\}, \{a', b'\}\}\). If \(a \neq b\), \(\{a\}\) and \(\{a, b\}\) = \(\{a', b'\}\). So, first, \(a = a'\) and then \(\{a, b\} = \{a', b'\}\) implies \(b = b'\). If \(a = b\), \(\{a, a\}\) = \(\{\{a\}\}\). So \(\{a\}\) = \(\{a'\}\), \(\{a\}\) = \(\{a', b'\}\), and we get \(a = a' = b'\), so \(a = a'\) and \(b = b'\) holds in this case also. Ordered pairs can be extended to define ordered triples: \((a, b, c) = ((a, b), c)\). Similarly, we can define ordered quadruples: \((a, b, c, d) = ((a, b, c), d)\), and so on. These extensions are useful for modeling subsystems that number more than two.

**0.1.2 Set Relations**

Set relation is defined in terms of a binary relation, which is determined by specifying all ordered pairs of objects in that relation. **Relation Definitions:** A set \(R\) is a binary relation if all elements of \(R\) are ordered pairs, i.e., if for any \(z\) in \(R\) there exist \(x\) and \(y\) such that \(z = (x, y)\). It is customary to write \(xRy\) instead of \((x, y) \in R\). We say that \(x\) is in relation \(R\) with \(y\) if \(xRy\) holds. The set of all \(x\) which are in relation \(R\) with some \(y\) is called the domain of \(R\) and denoted by “Dom \(R\).” So Dom \(R\) = \(\{x \mid \text{there exists } y \text{ such that } xRy\}\). Dom \(R\) is the set of all first coordinates of ordered pairs in \(R\). The set of all \(y\) such that, for some \(x\), \(x\) is in relation \(R\) with \(y\) is called the range of \(R\), denoted by “ran \(R\).” So ran \(R\) = \(\{y \mid \text{there exists } x \text{ such that } xRy\}\). In modeling systems integration parameters, the relations mapped from one sub-system to another become defining characteristics of the integration.

**0.1.3 Functions on Sets**

A function represents a special type of relation. A function is a rule that assigns to any object \(a\), from the domain of the function, a unique object \(b\), the value of the function at “\(a\).” The function implies a relation where every object “\(a\)” from the domain is related to precisely one object in the range, namely, to the value of the function at \(a\). This makes it possible to characterize component-to-component relationships for systems integration purposes.

**Function Definitions:** A binary relation \(F\) is called a function (or mapping,
correspondence) if \(aFb_1\) and \(aFb_2\) imply \(b_1 = b_2\) for any \(a, b_1, \) and \(b_2\). In other words, a binary relation \(F\) is a function if and only if for every \(a\) from \(\text{Dom } F\) there is exactly one \(b\) such that \(aFb\). This unique \(b\) is called the value of \(F\) at \(a\) and is denoted \(F(a)\) or \(F_a\). Note that \(F(a)\) is not defined if \(a \not\in \text{Dom } F\). If \(F\) is a function with \(\text{Dom } F = A\) and range \(F \subseteq B\), we use the notations \(F : A \mapsto B, \langle F(a) \mid a \in A \rangle, \langle F_a \rangle_{a \in A}\) for the function \(F\). The range of the function \(F\) can then be denoted \(\{F(a) \mid a \in A\}\). The Axiom of Extensionality can be applied to functions as follows.

**Lemma.** Let \(F\) and \(G\) be functions. \(F = G\) if and only if \(\text{Dom } F = \text{Dom } G\) and \(F(x) = G(x)\) for all \(x \in \text{Dom } F\). A function \(f\) is called one-to-one or injective if \(a_1 \in \text{Dom } f, a_2 \in \text{Dom } f, \) and \(a_1 \neq a_2\) implies \(f(a_1) \neq f(a_2)\). In other words if \(a_1 \in \text{Dom } f, a_2 \in \text{Dom } f, \) and \(f(a_1) = f(a_2)\), then \(a_1 = a_2\).

### 0.1.4 Cardinality of Sets

The most obvious property of a set relates to how many elements it contains. It is a fundamental observation that we can define the statement “sets \(A\) and \(B\) have the same number of elements” without knowing anything about the elements.

**Cardinality Definitions:** Sets \(A\) and \(B\) have the same cardinality if there is a one-to-one function \(f\) with domain \(A\) and range \(B\). We denote this by \(|A| = |B|\). The cardinality of \(A\) is less than or equal to the cardinality of \(B\) (notation: \(|A| \leq |B|\)) if there is a one-to-one mapping of \(A\) into \(B\). Notice that \(|A| \leq |B|\) means that \(|A| = |C|\) for some subset \(C\) of \(B\). We also write \(|A| < |B|\) to mean that \(|A| \leq |B|\) and \(|A| \neq |B|\), i.e., that there is a one-to-one mapping of \(A\) onto a subset of \(B\), but there is no one-to-one mapping of \(A\) onto \(B\).

**Lemma:**

1. If \(|A| \leq |B|\) and \(|A| = |C|\), then \(|C| \leq |B|\).
2. If \(|A| \leq |B|\) and \(|B| = |C|\), then \(|A| \leq |C|\).
3. \(|A| \leq |A|\).
4. If \(|A| \leq |B|\) and \(|B| \leq |C|\), then \(|A| \leq |C|\).

**Cantor-Bernstein Theorem:** If \(|X| \leq |Y|\) and \(|Y| \leq |X|\), then \(|X| = |Y|\).

### I. SET-TO-SYSTEM AND SUBSET-TO-SUBSYSTEM RELATIONSHIPS

Based on the underlying principles of sets and relations, we can now take a system as a set and subsets as subsystems. Each system is represented by \(n\)-tuple as an extension of an ordered pair. For systems that are describable by a matrix, the eigenmodes, in order of magnitude, will be the \(n\)-th tuple and we take the dominant modes. This will generate a reduced order representation using singular value decomposition, which is addressed in the next section. Consider the domain of set relations such that if \(R\) is a relation from set \(A\) to set \(B\), then the domain of \(R\) is the set of all elements “\(a\)” belonging to \(A\) such that \(aRb\) for some element “\(b\)” belonging to \(B\). That is,

\[
\text{Dom}(R) = \{a \in A | (a, b) \in R \text{ for some } b \in B\}
\]  

(1.1)

Consider the two framed sets in Figure 1 (Badriu and Cheung, 2002). Let Frame \(A\) contain the subsets \(a1, a2, a3, a4, \) and \(a5\) while Frame \(B\) contains the subsets \(b1, b2, b3, b4, b5, b6, b7, \) and \(b8\). Let us define an “integrating” relation \(Z\) from \(A\) to \(B\) such that an element \(a\) belonging to \(A\) is related to an element \(b\) belonging to \(B\), if and only if there is a rule in \(B\) that has \(a\) as a premise and \(b\) as a complement.
From Figure 1, we can define the following relationships:

\[ Z = \text{Relation of relevant subset correspondence} \]

Domain of \( Z = \{a_1, a_2, a_3\} \)

Image of \( Z = \{b_2, b_5, b_6\} \)

(1.2)
Now suppose only the following relationships exist in Frame $B$:

\[ a_1 \equiv b_2 \]
\[ a_2 \equiv b_5 \]
\[ a_3 \equiv b_6, \quad (1.3) \]

where $\equiv$ represents reciprocity of parameter correspondence. The domain of $Z$ is then given by the set:

\[ \text{Dom}(Z) = \{a_1, a_2, a_3\}, \quad (1.4) \]

Since the elements $a_1$, $a_2$, and $a_3$ are the only elements of $A$ that can successfully integrate with elements in $B$. The image of the relation, $R$, is then defined as:

\[ \text{Im}(R) = \{b \in B \mid (a,b) \in R \text{ for some } a \in A\} \quad (1.5) \]

Thus, the image of the relation $Z$ (measure of system integration) is:

\[ \text{Im}(Z) = \{b_2, b_5, b_6\}, \quad (1.6) \]

which corresponds to the set of integrating relations that are effected in Frame $B$. A special kind of relation on integrating sets is parameter mapping in which there is one-to-one correspondence between elements in subsets (subsystems). An identity relation in a subsystem is a relation that integrates every element with itself such that:

\[ R = \{(a,a) \mid a \in A\}, \quad (1.7) \]

which is a reflexive (automatic) property indicating that every subsystem perfectly integrates with itself. In a similar vein, we can define a binary operation on a subsystem, $S$, whereby the Cartesian square $S \times S$ produces other elements not necessarily in $S$. For example, let $S$ be the set of words in a dictionary. An operation may be defined for the process of forming natural language statements or new concatenation of words from the set. The new set of word-statements, although not resident in $S$, do originate from the elements of $S$; thus signifying interrelationships (or integration) of the elements in $S$. As another example, from the set of natural numbers $N$, one could randomly form a sequence of numbers such as 3, 1, 4, 1, and 7. This sequence itself is not in the set $N$, although each term of the sequence is in $N$. Based on the definition of relations on sets, the binary operation approach can, potentially, be used to convey a measure of integration across sub-systems.

### 1.1 Integration Mapping of Subsets

If the set $C$ is a subset of the Cartesian product, $A \times B$, of the sets $A$ and $B$, then $C$ is an integration mapping from $A$ to $B$ such that for each $a$ belonging to $A$ there is exactly one $b$ belonging to $B$ for which $(a, b)$ belong to $C$. Thus, the mapping, $C$, is the collection of the elements of $A \times B$ that have one-to-one correspondence or integration with the elements of $A$ and $B$. The elements of the sets $A$, $B$, and $C$ are specific parameters of objects contained in the main set of interest. In the word-set example mentioned earlier, $C = A \times B$ since each element $a$ of $A$, in integration with each element $b$ of $B$, yields a unique (distinct) ordered pair of words. The ordered pair formation can be extended to $n$-tuple to represent a finite sequence (or ordered list) of elements pulled from multiple systems to form an integrated subset. Thus, $n$-tuples can be used to mathematically describe integrated objects that consist of specified elements of cooperating systems of dimension $n$.

### 1.2 Model Reduction Approach

In order to model systems integration, particularly for system of systems, it is helpful
to represent the component systems in a more tractable form. Many alternate representational forms are available in the literature. One approach is to develop representational equations of model reduction of integrated systems. Soong (1977) presents a model reduction of large scale systems using a multivariate linear regression approach. Ibidapo-Obe (1982a) presents the model reduction in another form based on a singular value decomposition approach, which is more efficient computationally. For example, consider an \(n\)th-order linear system \(S\) defined by

\[
S_1 : x = Ax + B
\]

where \(x\) is the \(n\)-dimensional state vector, \(A\) is an \(n \times n\) system matrix, and \(u\) is a \(p\)-dimensional input vector. Let \(z\) be an \(m\)-vector \((m < n)\) related to \(x\) by

\[
z = Cx.
\]

In model reduction, it is desirable to find an \(m\)-order system described by

\[
S_2 : \dot{z} = Fx + Gu
\]

The \(m \times n\) matrix \(C\) in Eq. (1.9) is the aggregation matrix and \(S_2\) is the aggregated system or the reduced model. It is easy to show that \(G = CB\) and that \(F\) must satisfy the matrix equation:

\[
FC = CA.
\]

\(S_1, S_2\) are relations whilst \(A, B, C, F, G\) are sets/subsets

Equation (1.11) defines an over-specified system of equation for the unknown matrix \(F\), and hence \(F\) must be approximated. In Soong (1977), a multivariate linear regression scheme is used to yield a “best” approximation for \(F\) in the form

\[
\hat{F} = CAC^T (CC^T)^{-1}
\]

where \(T\) and \(-1\) denote matrix transpose and matrix inverse, respectively. The rank of \(C\) is assumed to be \(m\). The result given by Eq. (1.12) is interpreted as a linear, unbiased, minimum-variance estimate of \(F\) and its form agrees with that given by Aoki (1968), following an ad hoc procedure. In addition, the covariance of \(\hat{F}\) is found to be

\[
\text{cov}(\text{vec}\hat{F}) = \sigma^2 [(CC^T)^{-1} \otimes I_m]
\]

and it is shown in Soong (1977) that this covariance matrix can be used for model reduction error assessment. In Eq. (1.13), the Kronecker product \(\otimes\) and the “vec” operator are defined as

\[
P \otimes Q = [P_{ij} \ Q]
\]

\[
\text{vec}(P) = [P_{1 \ 2 \ \cdots}]^T
\]

where \(P\) and \(Q\) are matrices of arbitrary dimensions and \(P_k\) is the \(k\)th column of \(P\).

### I.3 Singular Value Decomposition

From the computational point of view, it is desirable to circumvent the use of matrix inverses in Eqs. (1.12) and (1.13), particularly for systems having large aggregation matrices. In what follows, this is accomplished through the use of matrix singular value decomposition (SVD), which has found useful application in linear least squares problems (Golub and Reinsch, 1970; Golub and Van Loan, 1979). The SVD concept gives the Moore-Penrose pseudo-inverse of \(C\) as
C^* = VAU^T \quad (1.15)

where U and V are unitary matrices whose columns are the eigenvectors of matrices DD^T, and D^T D, respectively, and

\[
\Lambda = \begin{bmatrix}
\sigma_1^{-1} & 0 & \cdots & 0 \\
0 & \sigma_2^{-1} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_m^{-1} \\
0 & 0 & \cdots & 0
\end{bmatrix}
\]

where \(\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0\), called singular values, are the nonnegative square roots of the eigenvalues of \(D^T D\). A discussion of this decomposition and its properties can be found in Stewart (1973). Now, Eq. (1.11) gives

\[\hat{F} = CAC^+ \quad (1.16)\]

And, using SVD, we can write

\[\hat{F} = CA(V\Lambda U^T) \quad (1.17)\]

The matrix C can also be written in the form

\[D = U\hat{\alpha} V^T \quad (1.18)\]

where

\[
\sum = \begin{bmatrix}
\sigma_1 & 0 & \cdots & 0 \\
0 & \sigma_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_m \\
0 & 0 & \cdots & 0
\end{bmatrix}_{n \times n}
\]

and we have

\[\hat{F} = U\sum V^T AVU^T \quad (1.19)\]

Compared with Eq. (1.12), either Eq. (1.18) or Eq. (1.20) provides a more efficient method of computation for \(\hat{F}\) due to elimination of the matrix inverse. Similarly, advantages are realized in the calculation of \(\text{cov} (\text{vec} \hat{F})\). Following the SVD scheme,

\[
(DD^T)^{-1} = (D^T D)^+ = (U\Lambda U^T)(V\Lambda^2 U^T) \quad (1.20)
\]

Equation (1.13) now takes the form

\[\text{cov} (\text{vec} F) = \sigma [U\Lambda^2 U^T] \quad (1.21)\]

which is clearly of a simpler structure than Eq. (1.13).

### I.4 Sub-System Surface Projection Integrals

Using the concept of system reduction; we consider the projection of one sub-system onto the plane of another sub-system to identify common region, area, or parameters (dominant eigenvalues) for integration. The idea is to identify sub-sets that represent segments (sub-systems) of the systems to be integrated. Fischer-Cripps (2005) presents formulations of line integrals that are adapted for the systems modeling in this section. Consider a surface \(z = f(x,y)\) in Figure 2. Let a line increment on the curve \(C\) in the \(xy\) plane be given by \(\Delta s\).

The area of the band formed from the line increment to the surface is given by the product expressed as shown below:

\[f(x,y)\Delta s\]

The total area under the surface along the curve from \(P_1\) to \(P_2\) represents the sum as \(\Delta s\) goes to zero. That is,
\[
\lim_{n \to \infty, \Delta s \to 0} \sum f(x, y)\Delta s
\] (1.22)

This limit is called the line integral of \(f(x, y)\) along the curve \(C\) from \(P_1\) to \(P_2\). Note that \(C\) is used to represent an interval or group of intervals along the curve.

Figure 2. Line Integral of Surface Projection
Referring to Figure 2, the line integral is written as:

\[ \int_{P_1}^{P_2} f(x, y) \, ds = \int_{C} f(x, y) \, ds \]  \hspace{1cm} (1.23)

If the curve C can be represented as a vector function in terms of the path or arc length parameter \( s \), such that:

\[ \mathbf{r}(s) = x(s) \mathbf{i} + y(s) \mathbf{j} \]

Then the line integral becomes:

\[ \int_{P_1}^{P_2} f(x(s), y(s)) \, ds \]  \hspace{1cm} (1.24)

If the curve C is given in terms of the parameter \( t \) such that:

\[ \mathbf{r}(t) = x(t) \mathbf{i} + y(t) \mathbf{j} \]

Then, we have:

\[ \int_{C} f(x(t), y(t)) \, ds = \int_{P_1}^{P_2} f(x, y) \, dt \]  \hspace{1cm} (1.25)

where

\[ \frac{ds}{dt} = \sqrt{ \left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2 } \]  \hspace{1cm} (1.27)

The line integral can be extended to three dimensions where \( F = f(x, y, z) \). Then, the line integral is represented as:

\[ \int_{C} f(x, y, z) \, ds \]  \hspace{1cm} (1.27)

It is possible to express the line integral of \( F(x, y) \) along C with respect to \( x \) (or \( y \)) instead of the arc length \( s \). This gives the projected area on the xz plane (for \( \Delta x \)) or the yz plane (for \( \Delta y \)). That is, the curve segment \( \Delta s \) projects onto x-axis as \( \Delta x \) and onto y-axis as \( \Delta y \).

Consequently, in this case, the surface \( z \) may be represented by the function \( f(x, y) \) in the \( x \) direction and \( g(x, y) \) in the \( y \) direction. The line integral then becomes:

\[ \int_{C} F(x, y) \, ds = \int_{C} f(x, y) \, dx + \int_{C} g(x, y) \, dy \]

\[ = \int_{C} (f(x, y, z) \, dx + g(x, y, z) \, dy) \]  \hspace{1cm} (1.28)

For the general case for three dimensions, the line integral is given by the sum of the projected line integrals and the functions, \( f, g, \) and \( h \). Then, we have:

\[ \int_{C} F(x, y, z) \, ds = \int_{C} f(x, y, z) \, dx + \int_{C} g(x, y, z) \, dy + \int_{C} h(x, y, z) \, dz \]

\[ = \int_{C} (f(x, y, z) \, dx + g(x, y, z) \, dy + h(x, y, z) \, dz) \]  \hspace{1cm} (1.29)
The functions $f$, $g$, and $h$ may be components of a vector function $\mathbf{F}$, such that:

$$\mathbf{F} = f(x, y, z)\mathbf{i} + g(x, y, z)\mathbf{j} + h(x, y, z)\mathbf{k}$$

$$= F_1\mathbf{i} + F_2\mathbf{j} + F_3\mathbf{k}$$

Now, we have:

$$F_1dx + F_2dy + F_3dz = \left( \frac{dx}{ds} + \frac{dy}{ds} + \frac{dz}{ds} \right) ds$$

$$= \left( \mathbf{F}.\frac{dr}{ds} \right) ds$$

(1.31)

where:

$$dr = dx\mathbf{i} + dy\mathbf{j} + dz\mathbf{k}$$

and

$$\frac{dr}{ds} = \frac{dx}{ds}\mathbf{i} + \frac{dy}{ds}\mathbf{j} + \frac{dz}{ds}\mathbf{k}$$

Consequently, the line integral $\int_C \mathbf{F}.d\mathbf{r}$ can be written as $\int_C \mathbf{F}.d\mathbf{r}$, which is referred to as the circulation of $\mathbf{F}$ around $C$. In general, the value of the line integral depends upon the path of integration. If, however, the vector field $\mathbf{F}$ can be represented by:

$$\mathbf{F} = \nabla \phi,$$

(1.33)

Then the line integral is independent of the path. In this case, $\phi$ is called the scalar potential. The field $\mathbf{F}$ is said to be a conservative field and $d\phi$ is called an exact differential. Now, we have

$$\int_{P_1}^{P_2} \mathbf{F}.d\mathbf{r} = \int_{P_1}^{P_2} \nabla \phi.d\mathbf{r}$$

$$= \left. [ \phi(x, y, z) ] \right|_{P_1}^{P_2}$$

$$= \phi(x_2, y_2, z_2) - \phi(x_1, y_1, z_1)$$

(1.35)

The integral depends only on the end points $P_1$ and $P_2$ and not on the path joining them. This is verified by the fact that the form or equation of the curve $C$ is not defined in the above formulations. If the path of integration of $C$ is a closed curve, and if $\mathbf{F}$ is conservative, then we have:

$$\oint_C \mathbf{F}.d\mathbf{r} = 0,$$

(1.34)

where the circle in the integral sign denotes integration around a closed curve. For systems integration application, the area of integration can be used to represent the magnitude of overlap between sub-systems.

I.5 Set Projection and System Overlap

Systems overlap for integration purposes can further be represented as projection integrals by considering areas bounded by the common elements of sub-systems. In Figure 3, the projection of a flat plane onto the first quadrant is represented as area $A$ while Figure 4 shows the projection on an inclined plane as area $B$. 
\[ A = \int_{A_y}^{A_y} \int_{A_x}^{A_x} z(x, y) \, dy \, dx \quad (1.35) \]

\[ B = \int_{B_y}^{B_y} \int_{B_x}^{B_x} z(x, y) \, dy \, dx \quad (1.36) \]

Figure 3. Flat Plane Projection
The net projection encompassing the overlap of A and B is represented as area C, shown in Figure 5, and computed as:

\[ C = \int_{C} \int_{C} \mathcal{z}(x, y) dy dx \]  

(1.37)
Figure 5. Net projection area
II. TIME-VARIANT SYSTEMS INTEGRATION

Time-variant systems pose additional complexity to systems integration. This can be addressed by stochastic techniques. The problem of stochastic estimation in non-linear systems has been formulated using different mathematical models and techniques such as the stochastic differential equation approach and function-space methods. Ibidapo-Obe (1982b) presents a scheme, based on eigenmode expansion, for the estimation of system responses in distributed parameter systems. Although some of the expositions demonstrate organic generality of the proposed non-linear structures, the techniques available for the analysis and synthesis of these structures for stochastic processes do not yield solutions for the generalized models. An enhanced technique for optimal estimation of system parameters in a stochastic environment with non-linear sensors based on the eigenmode representation of generalized displacements and orthogonal expansion of the conditional joint probability distribution is now illustrated. A linear dynamical system shows the correspondence between the innovations process, the observations process and the integrated system response. The proposed algorithm which may be applied to several civil engineering and space structures, best described by the distributed parameter equation

\[ m(x)u_x(x,t) + Bu_t(x,t) + Cu(x,t) = F_0(x,t) \]  

(2.1)

with the initial conditions

\[ u(x,t_0) = u^0 \quad \text{and} \quad u_t(x,t_0) = u^0_t \]  

(2.2)

where \( m(x) \) is the mass per unit length (assumed unity); \( u(x,t) \) the vector of generalized displacements; \( B \) and \( C \) are in general spatial differential operators representing the damping and restoring forces respectively; and \( F_0(x,t) \) is the external impressed forces (wind loads, earthquake excitations, etc.).

II.1 Modal Canonical Representation

Let the displacement be expressed in terms of structural dominant mode shapes \( \{\phi_i(x)\}_{i=1}^N \) with associated frequencies \( \{\omega_i\}_{i=1}^N \)

\[ u(x,t) = \sum_{i=1}^{N} u_i(t)\phi_i(x); \quad F_0(x,t) = \sum_{i=1}^{N} F_i(t)\phi_i(t) \]  

(2.3)

so that

\[ u(x,t) = U^T(t)\Phi(x) \]  

and

\[ F_0(x,t) = F_0^T(t)\Phi(x) \]  

(2.4)

where

\[ U = (u_{11} u_{12} \ldots u_{1N})^T, \quad F = (F_0^0 F_0^1 \ldots F_0^N)^T \]  

(2.5)

and

\[ \Phi = (\phi_1, \phi_2, \ldots, \phi_N)^T \]  

(2.6)

Substituting eqn. (2.4) to eqn. (2.1) gives

\[ \ddot{U}^T(t)\Phi(x) + B\dot{U}^T(t)\Phi(x) + C U^T(t)\Phi(x) = F_0^T(t)\Phi(x) \]  

(2.7)

thus obtaining the \( i^{th} \) mode amplitude equation

\[ \ddot{u}_i(t) + 2\zeta_i \omega_i \dot{u}_i(t) + \omega_i^2 u_i(t) = F_0^i(t) \]  

(2.8)

where \( C\phi_i = \omega_i^2 \phi_i \) and \( B\phi_i = 2\zeta_i \omega_i \phi_i \) with appropriate initial conditions.
Equation (2.8) can now be put in the state space form
\[ \dot{q}(t) = Aq(t) + F \]  
(2.9)
such that
\[ q = (q_1, q_2)^T \]  
(2.10)
with
\[ q_1 = u, \quad q_2 = \dot{u}, \]
\[ A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\xi \omega \end{bmatrix} \]  
(2.11)
and
\[ F = (0, F^0)^T \]  
(2.12)
The sensors are assumed to have the form
\[ y(t) = z(t) + v(t); \quad z(t) = h[q(s); s \leq t] \]  
(2.13)
where
\[ E\{v(t)\} = 0; \quad E\{v(t)v^T(s)\} = R(t)\delta(t - s); \quad E\{z(t)\} = 0; \]
\[ f(t) \leq M < \infty; \quad E\{v(t)F^T\} = 0 \]
for all \( t \); \( \int E\{z(t)z^T(t)\}dt < \infty; \quad E\{v(t)F^T\} = 0 \)
and \( h \) is a functional of \( q(s) \). It is further assumed that the forcing function \( F \) is additively (preferably Gaussian) random.

II.2 Canonical Estimation Procedure

The least squares estimate of \( q(t) \) is
\[ \hat{q}(t|\tau) = E\{q(\tau)|v(s), 0 \leq s \leq \tau\} \]  
(2.14)
Assuming that the property of “causal equivalence” holds, the sensor equations may be transformed into a white Gaussian process \( v(t) \), called the strict sense innovations process with a simpler structure so that eqn. (2.14) can now be put in the form
\[ \hat{q}(t|\tau) = \int E\{q(t)v^T(s)\}v(\sigma), 0 \leq \sigma \leq \tau \} v(s)ds \]  
(2.15)
where \( v(t) = y(t) - \dot{z}(t), 0 \leq t < \tau \) and \( \int \) is the \( \text{Ito}^{(1951)} \) integral. A differential structure for eqn. (2.15) is obtained as
\[ \dot{\hat{q}}(t|\tau) + A\hat{q}(t|\tau) + K(t)v(t) \]  
(2.16)
where
\[ K(t) = E\{q(t)v(t)\}, 0 \leq \sigma \leq t \} \]
(2.17)
Conditional joint probability distribution functions are required to obtain explicit solution to the stochastic differential equation (2.16). A method for the evaluation of the probability densities is obtained in the form of a partition theorem (Lainiotis 1971).
\[ p(q_f^t) = A(tq)p(q_f^t) p(q)A(tq)dq \]  
(2.18)
with \( f(t) = \{y(t)\} \) and
\[ \Lambda(q) = \exp\left\{ \int \hat{h}(s,q)R^{-1}(s)\gamma(s)ds - \int \int h(s,q)R^{-1}(s)\gamma(s)ds \right\} \]  
(2.19)
where
\[ \hat{h}(s,q) = E\{h(s,q)|f^t\} \]  
(2.20)
and \( p(q_0) \) is the prior density function.

Equation (2.18) can be approximated by expanding the distributions \( A(t|q) \) in a Volterra power series (Barrett and Lampard 1955) with functional terms
\[ \Lambda(q) = \sum_{k=0}^{\infty} \Lambda_k(q_f^t) \]  
(2.21)
where
such that \( g_i(...) \), \( i = 1, \ldots, n \) are the integral kernels describing the system and \( v(\cdot) \) is the innovations process. It has been assumed that in order to satisfy the requirements of physical realizability, the kernels are zero for any argument less than zero. The kernels can be identified in the following manner:

(i) First order kernel: \( g_1(t) \)

Let \( v(t) = A\delta_0(t) \) be an impulse of strength \( A \) so that

\[
A(t|q) = Ag_1(t) + A^2 g_2(t,t) + A^3 g_3(t,t,t) + \ldots
\]

and

\[
g_1(t_1) = \left. \frac{\partial \Lambda(t_1,q)}{\partial A} \right|_{A=0}
\]

(ii) Second order kernel: \( g_2(t,t) \)

Let \( P(t) = A\delta_0(t) + B\delta_0(t+\tau) \) be two impulses at \( t \) and \( t+\tau \) of strength \( A \) and \( B \) respectively where \( A \) and \( B \) are specified constants as above: then

\[
\Lambda(t\theta) = Ag_1(t) + Bg_1(t+\tau) + A^2 g_2(t,t) + A^3 g_3(t,t,t) + \ldots
\]

and the second order kernel

\[
g_2(t,t+\tau) = \left. \frac{1}{2} \frac{\partial^2 \Lambda(t_1|q)}{\partial A\partial B} \right|_{A=0, B=0}
\]

Higher order kernels can be computed in similar manners. The system state estimates can now be obtained as

\[
\Lambda(q) = \frac{1}{t!} \int_0^\infty \cdots \int_0^\infty v(t_1, \tau_1, \tau_2, \ldots, \tau_i) q(t_1) \prod_{k=1}^i v(t-k) \prod_{k=1}^i J(t, \eta_k, \eta_0) k(\tau) v(\tau) d\tau
\]

where

\[
J(t, \eta_0) = \exp \left\{ A(t - t_0) \right\}
\]

represents the transition/fundamental matrix; \( k(\tau) \) having been approximated using the eqns. (2.18) and (2.21).

II.3 Dynamic System Example

An hypothetical linear dynamical system is solved below, using the Kalman filtering equations, to illustrate the close correlation (in a non-statistical sense) between the observations process, the innovations process and the systems state. It has been assumed that the distributed parameter system is reducible to a linear ordinary differential equation system as follows

System dynamics:

\[
\dot{x}(t) = 0.5x(t) + u(t), \quad x(0) = 0
\]

Observation process:

\[
y(t) = x(t) + v(t)
\]

Figure 6 represents the observations process, which is simply taken here as the system response plus a white Gaussian noise. Figure 7 is the innovations representation. Figure 8 is the innovations auto-correlation, which indicates that the process is a white noise. Finally, Fig. 9 shows the system state, \( x(t) \). An approximation technique becomes imperative in the non-linear case since the innovations sequence generated is not a true innovations process and hence a scheme for adaptive improvement has to be employed.
Figure 6. Systems Observation Process
Figure 7. Systems Innovation Process
Figure 8. Autocorrelation for innovations process
Figure 9. System Response
III. CONCLUSIONS

Systems integration is of general interest as an avenue to enhance organizational cohesion and achieve operational efficiency. However, the process of integrating sub-systems of a massively complex system is very intractable. This is because many underlying factors impinge upon the efficacy of an integrated system. Such factors may include technological, human, psychological, ethical, contractual, economic, and time-dependent issues pertinent for each sub-system to be integrated. So far, there is no quantitative modeling approach to assess the degree of systems integration. The set-theoretic approach presented in this paper attempts to fill that void. It represents the first step to further research in developing quantitative measures for systems integration. In such further research, one must consider various parameters that characterize a system as may be deterministic, stochastic, dynamic, static, continuous, or discrete so that quantitative modeling of integration can be applied. Simulation modeling can also be developed for assessing integration potential for systems involving several components that are too intractable to study analytically.

REFERENCES


ON SET-THEORETIC FUNDAMENTALS OF SYSTEMS INTEGRATION

Adedeji Badru  
Department of Systems & Engineering Management,  
Graduate School of Engineering,  
Air Force Institute of Technology,  
Dayton, Ohio,  
USA  
Email: dejibadru.com

&

O. Ibidapo-Obe  
Department of Systems Engineering,  
Faculty of Engineering,  
University of Lagos,  
Nigeria  
Email: oyeibidapoobe@yahoo.com

Received 31 July, 2008  
Communicated by Dele Oluwade

ABSTRACT

In spite of the much touted need to achieve systems integration, very little has been accomplished in practice. This is often so because there is no analytical basis for identifying and modeling the parameters that the sub-systems have in common. Consequently, attempts to achieve systems integration degenerates to mere conceptual formulations. Sometimes the conceptual formulations work, but most times they do not. The premise of this paper is that a rigorous set-theoretic formulation can help identify the dominant common elements of subsystems, thereby facilitating practical implementation that is amenable to quantitative optimization. The paper explores fundamental set theory and principles for modeling systems interactions.

Keywords: Systems Integration, Set Theory, Mathematical Modeling, Sub-system Analysis, Surface Project Integrals, Quantitative optimization
0. INTRODUCTION

In order for two systems to be integrated, they must overlap in scope or performance with respect to time, space, or physics; subsequently, this requires mathematical representations for manipulating the integrated parameters of the systems. In spite of the much touted need to achieve systems integration, very little has been accomplished in practice. This is often because there is limited analytical basis for modeling the parameters that the sub-systems have in common. Consequently, attempts to achieve systems integration degenerates to mere conceptual formulations. Sometimes the conceptual formulations work, but most times they do not. Omicini and Papadopoulos (2001) present systems integration as the coordination and management of interactions among entities of a system. The premise of this paper is that a rigorous set-theoretic formulation can help identify the dominant common elements of subsystems, thereby facilitating practical implementation that is amenable to quantitative optimization. The paper explores fundamental set theory and principles for modeling system interactions. A system is often defined as a collection of interrelated elements working together in synergy to produce a composite output that is greater than the sum of the individual outputs of the components. Systems integration is a desired state whereby sub-systems of the system are in proper consonance of operation and input-output interactions. Methodologies pertaining to the various characteristics of integrated systems are available in the literature. As a practical application example, Balkhi (1999) addressed integrated inventory systems with time-varying demand. The approach presented in this paper differs from those existing in the literature by focusing on set-theoretic modeling.

0.1 Theory of Sets and Systems

Set theory is the mathematical concept that governs sets, which represent collections of objects. Sets are defined by the elements contained in the collection and the respective memberships of the elements in the sets. Set theory provides the formalism in which mathematical objects are described. This makes set theory suitable for defining systems integration. In this conception, sets and set membership are fundamental concepts describing points and lines in Euclidean geometry. The properties of sets essential for modeling systems integration include:

- 1. Ordered Pairs
- 2. Relations
- 3. Functions
- 4. Set Cardinality

In this paper, standard set and matrix notations and symbols that are generally available in the literature are used for the mathematical formulations presented.

0.1.1 Ordered Pairs on Sets

The notion of ordered pair suggests that if \( a \) and \( b \) are sets, then the unordered pair \( \{a, b\} \) is a set whose elements are exactly \( a \) and \( b \). The “order” in which \( a \) and \( b \) are put together plays no role. Thus, \( \{a, b\} = \{b, a\} \). For many applications, we need to pair \( a \) and \( b \) in a way making possible to “read off” which set comes
“first” and which comes “second.” We denote this ordered pair of \(a\) and \(b\) by \((a, b)\); \(a\) is the first coordinate of the pair \((a, b)\), \(b\) is the second coordinate. To assess systems integration properties, the ordered pair has to be a set. It should be defined in such a way that two ordered pairs are equal if and only if their first coordinates are equal and their second coordinates are equal. This guarantees in particular that \((a, b) \neq (b, a)\) if \(a \neq b\). With this understanding, we can distinguish systems integration from the perspective of “push system” or “pull system” because they may not necessarily be identical.

**Basic definition of ordered pairs:** The basic definition of ordered pair, generally available in the literature specifies that \((a, b) = \{(a), \{a, b\}\}\). If \(a \neq b\), \((a, b)\) has two elements, a singleton \(\{a\}\) and an unordered pair \(\{a, b\}\). We find the first coordinate by looking at the element of \(\{a\}\). The second coordinate is then the other element of \(\{a, b\}\). If \(a = b\), then \((a, a) = \{(a), \{a\}\} = \{\{a\}\}\) has only one element. In any case, it seems obvious that both coordinates can be uniquely “read off” from the set \((a, b)\).

**Ordered Pair Theorem:** \((a, b) = (a', b')\) if and only if \(a = a'\) and \(b = b'\).

**Proof.** If \(a = a'\) and \(b = b'\), then, of course, \((a, b) = \{(a), \{a, b\}\} = \{(a'), \{a', b'\}\} = (a', b')\). The other implication is more intricate. Let us assume that \((a, b) = \{(a'), \{a', b'\}\}\). If \(a \neq b\), \(\{a\}\) and \(\{a, b\}\) \(\neq \{a', \{a', b'\}\}\). If \(a = b\), then \(\{(a), \{a, b\}\} = \{(a'), \{a', b'\}\}\). So, first, \(a = a'\) and then \(\{a, b\} = \{a', b'\}\) implies \(b = b'\). If \(a = b\), \(\{(a), \{a, a\}\} = \{\{a\}\}\). So \(\{a\} = \{a', \{a', b'\}\}\), and we get \(a = a' = b'\), so \(a = a'\) and \(b = b'\) holds in this case also. Ordered pairs can be extended to define ordered triples: \((a, b, c) = ((a, b), c)\). Similarly, we can define ordered quadruples: \((a, b, c, d) = ((a, b, c), d)\), and so on. These extensions are useful for modeling subsystems that number more than two.

**0.1.2 Set Relations**

Set relation is defined in terms of a binary relation, which is determined by specifying all ordered pairs of objects in that relation. **Relation Definitions:** A set \(R\) is a binary relation if all elements of \(R\) are ordered pairs, i.e., if for any \(z\) in \(R\) there exist \(x\) and \(y\) such that \(z = (x, y)\). It is customary to write \(xRy\) instead of \((x, y) \in R\). We say that \(x\) is in relation \(R\) with \(y\) if \(xRy\) holds. The set of all \(x\) which are in relation \(R\) with some \(y\) is called the domain of \(R\) and denoted by “Dom \(R\)”. So \(Dom R = \{x \mid \text{there exists } y \text{ such that } xRy\}\). \(Dom R\) is the set of all first coordinates of ordered pairs in \(R\). The set of all \(y\) such that, for some \(x\), \(x\) is in relation \(R\) with \(y\) is called the range of \(R\), denoted by “ran \(R\)” So \(run R = \{y \mid \text{there exists } x \text{ such that } xRy\}\). In modeling systems integration parameters, the relations mapped from one sub-system to another become defining characteristics of the integration.

**0.1.3 Functions on Sets**

A function represents a special type of relation. A function is a rule that assigns to any object \(a\), from the domain of the function, a unique object \(b\), the value of the function at “\(a\)” The function implies a relation where every object “\(a\)” from the domain is related to precisely one object in the range, namely, to the value of the function at \(a\). This makes it possible to characterize component-to-component relationships for systems integration purposes.

**Function Definitions:** A binary relation \(F\) is called a function (or mapping,
correspondence) if \( aFb_1 \) and \( aFb_2 \) imply \( b_1 = b_2 \) for any \( a, b_1, \) and \( b_2 \). In other words, a binary relation \( F \) is a function if and only if for every \( a \) from Dom \( F \) there is exactly one \( b \) such that \( aFb \). This unique \( b \) is called the value of \( F \) at \( a \) and is denoted \( F(a) \) or \( F_a \). Note that \( F(a) \) is not defined if \( a \notin \text{Dom} \ F \). If \( F \) is a function with \( \text{Dom} \ F = A \) and \( \text{range} \ F \subseteq B \), we use the notations \( F : A \mapsto B \), \( F(a) | a \in A \), \( F_a | a \in A \), for the function \( F \). The range of the function \( F \) can then be denoted \( \{F(a) | a \in A\} \). The Axiom of Extensionality can be applied to functions as follows.

**Lemma.** Let \( F \) and \( G \) be functions. \( F = G \) if and only if \( \text{Dom} \ F = \text{Dom} \ G \) and \( F(x) = G(x) \) for all \( x \in \text{Dom} \ F \). A function \( f \) is called one-to-one or injective if \( a_1 \in \text{Dom} \ f, \ a_2 \in \text{Dom} \ f, \) and \( a_1 \neq a_2 \) implies \( f(a_1) \neq f(a_2) \). In other words if \( a_1 \in \text{Dom} \ f, \ a_2 \in \text{Dom} \ f, \) and \( f(a_1) = f(a_2) \), then \( a_1 = a_2 \).

0.1.4 Cardinality of Sets

The most obvious property of a set relates to how many elements it contains. It is a fundamental observation that we can define the statement “sets \( A \) and \( B \) have the same number of elements” without knowing anything about the elements.

**Cardinality Definitions:** Sets \( A \) and \( B \) have the same cardinality if there is a one-to-one function \( f \) with domain \( A \) and range \( B \). We denote this by \( |A| = |B| \). The cardinality of \( A \) is less than or equal to the cardinality of \( B \) (notation: \( |A| \leq |B| \)) if there is a one-to-one mapping of \( A \) into \( B \). Notice that \( |A| \leq |B| \) means that \( |A| = |C| \) for some subset \( C \) of \( B \). We also write \( |A| < |B| \) to mean that \( |A| \leq |B| \) and \( |A| \neq |B| \), i.e., that there is a one-to-one mapping of \( A \) onto a subset of \( B \), but there is no one-to-one mapping of \( A \) onto \( B \).

**Lemma:**
1. If \( |A| \leq |B| \) and \( |A| = |C| \), then \( |C| \leq |B| \).
2. If \( |A| \leq |B| \) and \( |B| = |C| \), then \( |A| \leq |C| \).
3. \( |A| \leq |A| \).
4. If \( |A| \leq |B| \) and \( |B| \leq |C| \), then \( |A| \leq |C| \).

**Cantor-Bernstein Theorem:** If \( |X| \leq |Y| \) and \( |Y| \leq |X| \), then \( |X| = |Y| \).

I. SET-TO-SYSTEM AND SUBSET-TO-SUBSYSTEM RELATIONSHIPS

Based on the underlying principles of sets and relations, we can now take a system as a set and subsets as subsystems. Each system is represented by \( n \)-tuple as an extension of an ordered pair. For systems that are describable by a matrix, the eigenmodes, in order of magnitude, will be the \( n \)-th tuple and we take the dominant modes. This will generate a reduced order representation using singular value decomposition, which is addressed in the next section. Consider the domain of set relations such that if \( R \) is a relation from set \( A \) to set \( B \), then the domain of \( R \) is the set of all elements “\( a \)” belonging to \( A \) such that \( aRb \) for some element “\( b \)” belonging to \( B \). That is,

\[
\text{Dom}(R) = \{a \in A | (a, b) \in R \text{ for some } b \in B\}
\]

(1.1)

Consider the two framed sets in Figure 1 (Badiru and Cheung, 2002). Let Frame \( A \) contain the subsets \( a1, a2, a3, a4, \) and \( a5 \) while Frame \( B \) contains the subsets \( b1, b2, b3, b4, b5, b6, b7, \) and \( b8 \). Let us define an “integrating” relation \( Z \) from \( A \) to \( B \) such that an element \( a \) belonging to \( A \) is related to an element \( b \) belonging to \( B \), if and only if there is a rule in \( B \) that has \( a \) as a premise and \( b \) as a complement.
Figure 1. Relations on System Subsets

From Figure 1, we can define the following relationships:

\[ Z = \text{Relation of relevant subset correspondence} \]

Domain of \( Z = \{a1, a2, a3\} \)

Image of \( Z = \{b2, b5, b6\} \)

(1.2)
Now suppose only the following relationships exist in Frame B:

\[ a1 \sqdot b2 \]
\[ a2 \sqdot b5 \]
\[ a3 \sqdot b6, \]

where \( \sqdot \) represents reciprocity of parameter correspondence. The domain of \( Z \) is then given by the set:

\[ \text{Dom}(Z) = \{a1, a2, a3\} \]  

Since the elements \( a1, a2, \) and \( a3 \) are the only elements of \( A \) that can successfully integrate with elements in \( B \). The image of the relation, \( R \), is then defined as:

\[ \text{Im}(R) = \{b \in B \mid (a, b) \in R \text{ for some } a \in A\} \]  

Thus, the image of the relation \( Z \) (measure of system integration) is:

\[ \text{Im}(Z) = \{b2, b5, b6\} \]  

which corresponds to the set of integrating relations that are effected in Frame B. A special kind of relation on integrating sets is parameter mapping in which there is one-to-one correspondence between elements in subsets (subsystems). An identity relation in a subsystem is a relation that integrates every element with itself such that:

\[ R = \{(a, a) \mid a \in A\} \]  

which is a reflexive (automatic) property indicating that every subsystem perfectly integrates with itself. In a similar vein, we can define a binary operation on a subsystem, \( S \), whereby the Cartesian square \( S \times S \) produces other elements not necessarily in \( S \). For example, let \( S \) be the set of words in a dictionary. An operation may be defined for the process of forming natural language statements or new concatenation of words from the set. The new set of word-statements, although not resident in \( S \), do originate from the elements of \( S \); thus signifying interrelationships (or integration) of the elements in \( S \). As another example, from the set of natural numbers \( N \), one could randomly form a sequence of numbers such as 3, 1, 4, 1, and 7. This sequence itself is not in the set \( N \), although each term of the sequence is in \( N \). Based on the definition of relations on sets, the binary operation approach can, potentially, be used to convey a measure of integration across sub-systems.

### 1.1 Integration Mapping of Subsets

If the set \( C \) is a subset of the Cartesian product, \( A \times B \), of the sets \( A \) and \( B \), then \( C \) is an integration mapping from \( A \) to \( B \) such that for each \( a \) belonging to \( A \) there is exactly one \( b \) belonging to \( B \) for which \((a, b)\) belong to \( C \). Thus, the mapping, \( C \), is the collection of the elements of \( A \times B \) that have one-to-one correspondence or integration with the elements of \( A \) and \( B \). The elements of the sets \( A, B, \) and \( C \) are specific parameters of objects contained in the main set of interest. In the word-set example mentioned earlier, \( C = A \times B \) since each element \( a \) of \( A \), in integration with each element \( b \) of \( B \), yields a unique (distinct) ordered pair of words. The ordered pair formation can be extended to \( n \)-tuple to represent a finite sequence (or ordered list) of elements pulled from multiple systems to form an integrated subset. Thus, \( n \)-tuples can be used to mathematically describe integrated objects that consist of specified elements of cooperating systems of dimension \( n \).

### 1.2 Model Reduction Approach

In order to model systems integration, particularly for system of systems, it is helpful...
to represent the component systems in a more tractable form. Many alternate representational forms are available in the literature. One approach is to develop representational equations of model reduction of integrated systems. Soong (1977) presents a model reduction of large scale systems using a multivariate linear regression approach. Ibidapo-Obe (1982a) presents the model reduction in another form based on a singular value decomposition approach, which is more efficient computationally.

For example, consider an nth-order linear system $S$ defined by:

$$S_1: \dot{x} = Ax + B$$  \hspace{1cm} (1.8)

where $x$ is the n-dimensional state vector, $A$ is an $n \times n$ system matrix, and $u$ is a p-dimensional input vector. Let $z$ be an m-vector ($m < n$) related to $x$ by

$$z = Cx.$$  \hspace{1cm} (1.9)

In model reduction, it is desirable to find an $m \times m$-order $S$ described by a system $S_2$:

$$S_2: \dot{z} = Fz + Gu$$  \hspace{1cm} (1.10)

The $m \times n$ matrix $C$ in Eq. (1.9) is the aggregation matrix and $S_2$ is the aggregated system or the reduced model. It is easy to show that $G = CB$ and that $F$ must satisfy the matrix equation:

$$FC = CA.$$  \hspace{1cm} (1.11)

$S_1$, $S_2$ are relations whilst $A$, $B$, $C$, $F$, $G$ are sets/subsets

Equation (1.11) defines an over-specified system of equation for the unknown matrix $F$, and hence $F$ must be approximated. In Soong (1977), a multivariate linear regression scheme is used to yield a “best” approximation for $F$ in the form

$$\hat{F} = CA(C^T(C^T)^{-1})^T,$$  \hspace{1cm} (1.12)

where $T$ and $-1$ denote matrix transpose and matrix inverse, respectively. The rank of $C$ is assumed to be $m$. The result given by Eq. (1.12) is interpreted as a linear, unbiased, minimum-variance estimate of $F$, and its form agrees with that given by Aoki (1968), following an ad hoc procedure. In addition, the covariance of $\hat{F}$ is found to be

$$\text{cov}(\text{vec}\hat{F}) = \sigma^2[(CC^T)^{-1} \otimes I_m].$$  \hspace{1cm} (1.12)

and it is shown in Soong (1977) that this covariance matrix can be used for model reduction error assessment. In Eq. (1.13), the Kronecker product $\otimes$ and the “vec” operator are defined as

$$P \otimes Q = [P_{ij} \ Q]$$

$$\text{vec}(P) = [P_{1\ 2 \ \cdots \ k}]^T$$  \hspace{1cm} (1.13)

where $P$ and $Q$ are matrices of arbitrary dimensions and $P_k$ is the kth column of $P$.

### I.3 Singular Value Decomposition

From the computational point of view, it is desirable to circumvent the use of matrix inverses in Eqs. (1.12) and (1.13), particularly for systems having large aggregation matrices. In what follows, this is accomplished through the use of matrix singular value decomposition (SVD), which has found useful application in linear least squares problems (Golub and Reinsch, 1970; Golub and Van Loan, 1979).

The SVD concept gives the Moore-Penrose pseudo-inverse of $C$ as
where $U$ and $V$ are unitary matrices whose columns are the eigenvectors of matrices $DD^T$, and $D^TD$, respectively, and

$$
\sum = \begin{bmatrix}
\sigma_1 & 0 & \cdots & 0 \\
0 & \sigma_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_m
\end{bmatrix}_{n \times n}
$$

and we have

$$
F^* = U \sum V^T A \Lambda U^T
$$

(1.15)

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$, called singular values, are the nonnegative square roots of the eigenvalues of $D^TD$. A discussion of this decomposition and its properties can be found in Stewart (1973). Now, Eq. (1.11) gives

$$
\hat{F} = CAC^+
$$

(1.16)

And, using SVD, we can write

$$
\hat{F} = CA(V\Lambda U^T)
$$

(1.17)

The matrix $C$ can also be written in the form

$$
D = U A \mathbf{V}^T
$$

(1.18)

where

$$
(DD^T)^{-1} = (D^T)^+ D^+ = (U \Lambda^2 U^T)(V\Lambda V^T)
$$

(1.19)

Equation (1.13) now takes the form

$$
\text{cov}(\text{vec}\hat{F}) = \sigma [U \Lambda U \otimes I_m]
$$

(1.21)

which is clearly of a simpler structure than Eq. (1.13).

### I.4 Sub-System Surface Projection Integrals

Using the concept of system reduction; we consider the projection of one sub-system onto the plane of another sub-system to identify common region, area, or parameters (dominant eigenvalues) for integration. The idea is to identify sub-sets that represent segments (sub-systems) of the systems to be integrated. Fischer-Cripps (2005) presents formulations of line integrals that are adapted for the systems modeling in this section. Consider a surface $z = f(x,y)$ in Figure 2. Let a line increment on the curve $C$ in the $xy$ plane be given by $\Delta s$.

The area of the band formed from the line increment to the surface is given by the product expressed as shown below:

$$
f(x, y)\Delta s
$$

The total area under the surface along the curve from $P_1$ to $P_2$ represents the sum as $\Delta s$ goes to zero. That is,
This limit is called the line integral of \( f(x, y) \) along the curve C from P1 to P2. Note that C is used to represent an interval or group of intervals along the curve.

\[
\lim_{n \to \infty; \Delta s \to 0} \sum f(x, y)\Delta s
\]  

(1.22)

Figure 2. Line Integral of Surface Projection
Referring to Figure 2, the line integral is written as:

$$\int_{p_1}^{p_2} f(x, y)ds = \int_{c} f(x(t), y(t)) \frac{ds}{dt} dt$$  \hspace{1cm} (1.25)

If the curve C can be represented as a vector function in terms of the path or arc length parameter $s$, such that:

$$r(s) = x(s)i + y(s)j$$

Then the line integral becomes:

$$\int_{p_1}^{p_2} f(x(s), y(s))ds$$  \hspace{1cm} (1.24)

If the curve C is given in terms of the parameter $t$ such that:

$$r(t) = x(t)i + y(t)j$$

Then, we have:

$$\int_{p}^{p_2} f(x, y)ds$$

It is possible to express the line integral of $F(x, y)$ along C with respect to $x$ (or $y$) instead of the arc length $s$. This gives the projected area on the xz plane (for $\Delta x$) or the yz plane (for $\Delta y$). That is, the curve segment $\Delta s$ projects onto x-axis as $\Delta x$ and onto y-axis as $\Delta y$. Consequently, in this case, the surface $z$ may be represented by the function $f(x, y)$ in the $x$ direction and $g(x, y)$ in the $y$ direction. The line integral then becomes:

$$\int_{c} F(x, y)ds = \int_{c} f(x, y)dx + \int_{c} g(x, y)dy$$

$$= \int_{c} (f(x, y, z)dx + g(x, y, z)dy)$$  \hspace{1cm} (1.28)

For the general case for three dimensions, the line integral is given by the sum of the projected line integrals and the functions, $f$, $g$, and $h$. Then, we have:

$$\int_{c} F(x, y, z)ds = \int_{c} f(x, y, z)dx + \int_{c} g(x, y, z)dy + \int_{c} h(x, y, z)dz$$

$$= \int_{c} (f(x, y, z)dx + g(x, y, z)dy + h(x, y, z)dz)$$  \hspace{1cm} (1.29)
The functions \( f, g, \) and \( h \) may be components of a vector function \( \mathbf{F} \), such that:

\[
\mathbf{F} = f(x, y, z)\mathbf{i} + g(x, y, z)\mathbf{j} + h(x, y, z)\mathbf{k}
\]

Now, we have:

\[
F_1 dx + F_2 dy + F_3 dz = \left( \frac{dx}{ds} + F_2 \frac{dy}{ds} + F_3 \frac{dz}{ds} \right) ds = \mathbf{F} \cdot \frac{dr}{ds} ds
\]

(1.31)

where:

\[
\frac{dr}{ds} = dx\mathbf{i} + dy\mathbf{j} + dz\mathbf{k}
\]

and

\[
\frac{dr}{ds} = \frac{dx}{ds}\mathbf{i} + \frac{dy}{ds}\mathbf{j} + \frac{dz}{ds}\mathbf{k}
\]

Consequently, the line integral \( \int_{C} \mathbf{F} \cdot dr \) can be written as \( \int_{P_1}^{P_2} \mathbf{F} \cdot dr \), which is referred to as the circulation of \( \mathbf{F} \) around \( C \). In general, the value of the line integral depends upon the path of integration. If, however, the vector field \( \mathbf{F} \) can be represented by:

\[
\mathbf{F} = \nabla \phi,
\]

(1.33)

Then the line integral is independent of the path. In this case, \( \phi \) is called the scalar potential. The field \( \mathbf{F} \) is said to be a conservative field and \( d\phi \) is called an exact differential. Now, we have

\[
\int_{P_1}^{P_2} \mathbf{F} \cdot dr = \int_{P_1}^{P_2} \nabla \phi \cdot dr
\]

\[
= \int_{C} \nabla \phi \cdot d\mathbf{r} = \int_{C} \left( \frac{\partial \phi}{\partial x} \mathbf{i} + \frac{\partial \phi}{\partial y} \mathbf{j} + \frac{\partial \phi}{\partial z} \mathbf{k} \right) \cdot (dx\mathbf{i} + dy\mathbf{j} + dz\mathbf{k})
\]

\[
= \int_{C} \left( \frac{\partial \phi}{\partial x} dx + \frac{\partial \phi}{\partial y} dy + \frac{\partial \phi}{\partial z} dz \right) ds
\]

(1.35)

The integral depends only on the end points \( P_1 \) and \( P_2 \) and not on the path joining them. This is verified by the fact that the form or equation of the curve \( C \) is not defined in the above formulations. If the path of integration of \( C \) is a closed curve, and if \( \mathbf{F} \) is conservative, then we have:

\[
\int_{C} \mathbf{F} \cdot dr = 0,
\]

(1.34)

where the circle in the integral sign denotes integration around a closed curve. For systems integration application, the area of integration can be used to represent the magnitude of overlap between sub-systems.

### I.5 Set Projection and System Overlap

Systems overlap for integration purposes can further be represented as projection integrals by considering areas bounded by the common elements of sub-systems. In Figure 3, the projection of a flat plane onto the first quadrant is represented as area \( A \) while Figure 4 shows the projection on an inclined plane as area \( B \).

Badru & Ibidapo-Obe, Systems Integration 14

\[ A = \int_{A_x}^{A_x} \int_{A_y}^{A_y} z(x, y) \, dy \, dx \]  \hspace{1cm} (1.35)

\[ B = \int_{B_x}^{B_x} \int_{B_y}^{B_y} z(x, y) \, dy \, dx \]  \hspace{1cm} (1.36)

Flat Plane Projection

Figure 3. Flat Plane Projection
The net projection encompassing the overlap of A and B is represented as area C, shown in Figure 5, and computed as:

\[ C = \int_{C_{0}} \int_{C_{1}} z(x, y) dy dx \]  \hspace{1cm} (1.37)
Figure 5. Net projection area
II. TIME-VARIANT SYSTEMS INTEGRATION

Time-variant systems pose additional complexity to systems integration. This can be addressed by stochastic techniques. The problem of stochastic estimation in non-linear systems has been formulated using different mathematical models and techniques such as the stochastic differential equation approach and function-space methods. Ibidapo-Obe (1982b) presents a scheme, based on eigenmode expansion, for the estimation of system responses in distributed parameter systems. Although some of the expositions demonstrate organic generality of the proposed non-linear structures, the techniques available for the analysis and synthesis of these structures for stochastic processes do not yield solutions for the generalized models. An enhanced technique for optimal estimation of system parameters in a stochastic environment with non-linear sensors based on the eigenmode representation of generalized displacements and orthogonal expansion of the conditional joint probability distribution is now illustrated. A linear dynamical system shows the correspondence between the innovations process, the observations process and the integrated system response. The proposed algorithm which may be applied to several civil engineering and space structures, best described by the distributed parameter equation

\[ m(x)u''(x,t) + Bu_t(x,t) + Cu(x,t) = F_0(x,t) \]  

(2.1)

with the initial conditions

\[ u(x,t_0) = u^0 \quad \text{and} \quad u_t(x,t_0) = u^0_t \]  

(2.2)

where \( m(x) \) is the mass per unit length (assumed unity); \( u(x,t) \) the vector of generalized displacements; \( B \) and \( C \) are in general spatial differential operators representing the damping and restoring forces respectively; and \( F_0(x,t) \) is the external impressed forces (wind loads, earthquake excitations, etc.).

II.1 Modal Canonical Representation

Let the displacement be expressed in terms of structural dominant mode shapes \( \{ \phi(x) \}_{i=1}^N \) with associated frequencies \( \{ w_i \}_{i=1}^N \)

\[ u(x,t) = \sum_{i=1}^{N} u_i(t) \phi_i(x); \quad F_0(x,t) = \sum_{i=1}^{N} F_i(t) \phi_i(t) \]  

(2.3)

so that

\[ u(x,t) = U^T(t) \Phi(x) \]  

(2.4)

and

\[ F_0(x,y) = F_0^T(t) \Phi(x) \]  

(2.5)

where

\[ U = (u \ u \ ... \ u) \}_{1,2,...,N}^T, \quad F = (F^0, F^0, ..., F^0) \]  

(2.6)

Substituting eqn. (2.4) to eqn. (2.1) gives

\[ \ddot{U}^T(t) \Phi(x) + B \dot{U}^T(t) \Phi(x) + C U^T(t) \Phi(x) = F_0^T(t) \Phi(x) \]  

(2.7)

thus obtaining the \( i^{th} \) mode amplitude equation

\[ \ddot{u}_i(t) + 2 \xi_i \omega_i \dot{u}_i(t) + \omega_i^2 u_i(t) = F_0^0(i) \]  

(2.8)

where \( C \phi_i = \omega_i^2 \phi_i \) and \( B \phi_i = 2 \xi_i \omega_i \phi_i \) with appropriate initial conditions.
Equation (2.8) can now be put in the state space form

\[ \dot{q}(t) = Aq(t) + F \]  

such that

\[ q = (q_1, q_2)^T \]  

with

\[ q_1 = u, \quad q_2 = \dot{u}, \]  

\[ A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\xi\omega_i \end{bmatrix} \]  

and

\[ F = \left(0, F^0_0\right)^T \]  

(2.12)

The sensors are assumed to have the form

\[ y(t) = z(t) + v(t); \quad z(t) = h[q(s); s \leq t] \]  

(2.13)

where

\[ \Lambda(q) = \exp\left\{ \int_{0}^{s} h(s, q) R_{i}(s) y(s) ds - \frac{1}{2} \int_{0}^{s} h(s, q)^2 ds \right\} \]  

(2.19)

where

\[ h(s, q) = E \{ h(s, q) \} \]  

(2.20)

and \( p(q_0) \) is the priori density function.

Equation (2.18) can be approximated by expanding the distributions \( \Lambda(t | q) \) in a Volterra power series (Barrett and Lampard 1955) with functional terms

\[ \Lambda(q) = \sum_{i=0}^{\infty} \Lambda_i(q) \]  

(2.21)

where
such that $g_i(...)$, $i = 1, ..., n$ are the integral kernels describing the system and $v(t)$ is the innovations process. It has been assumed that in order to satisfy the requirements of physical realizability, the kernels are zero for any argument less than zero. The kernels can be identified in the following manner:

(i) First order kernel: $g_1(t)$

Let $v(t) = A\delta_0(t)$ be an impulse of strength $A$ so that

$$A(t|q) = Ag_1(t) + A^2 g_2(t,t) + A^3 g_3(t,t,t) + ...$$

and

$$g_1(t_1) = \frac{\partial^2 A(t_1 q)}{\partial A^2} \bigg|_{A=0}$$

(ii) Second order kernel: $g_2(t,t)$

Let $P(t) = A\delta_0(t) + B\delta_0(t+t) \tau$ be two impulses at $t$ and $t+t$ of strength $A$ and $B$ respectively where $A$ and $B$ are specified constants as above; then

$$A(t\theta) = Ag(t) + Bg(t+\tau) + A^2 g(t,t) + A^3 g(t,t,t) + ...$$

and the second order kernel

$$g_2(t, t+\tau) = \frac{1}{2} \frac{\partial^2 A(t_1 q)}{\partial A \partial B} \bigg|_{A=0, B=0}$$

Higher order kernels can be computed in similar manners. The system state estimates can now be obtained as

$$\Lambda(q) = \frac{1}{t} \int_0^t d\tau_1 \cdots \int_0^t d\tau_i g_i(\tau_1, \tau_2, \cdots, \tau_i) \prod_{k=1}^i v(t-\tau_k)$$

(2.22)

where

$$J(t|q_0) = \exp \{ A(t-t_0) \}$$

represents the transition/fundamental matrix; $k(\tau)$ having been approximated using the eqns. (2.18) and (2.21).

**II.3 Dynamic System Example**

An hypothetical linear dynamical system is solved below, using the Kalman filtering equations, to illustrate the close correlation (in a non-statistical sense) between the observations process, the innovations process and the systems state. It has been assumed that the distributed parameter system is reducible to a linear ordinary differential equation system as follows

**System dynamics:**

$$\dot{x}(t) = 0.5x(t) + u(t), \quad x(0) = 0$$

(2.28)

**Observation process:**

$$y(t) = x(t) + v(t)$$

(2.29)

Figure 6 represents the observations process, which is simply taken here as the system response plus a white Gaussian noise. Figure 7 is the innovations representation. Figure 8 is the innovations auto-correlation, which indicates that the process is a white noise. Finally, Fig. 9 shows the system state $x(t)$. An approximation technique becomes imperative in the non-linear case since the innovations sequence generated is not a true innovations process and hence a scheme for adaptive improvement has to be employed.
Figure 6. Systems Observation Process
Figure 7. Systems Innovation Process
Figure 8. Autocorrelation for innovations process
Figure 9. System Response
III. CONCLUSIONS

Systems integration is of general interest as an avenue to enhance organizational cohesion and achieve operational efficiency. However, the process of integrating sub-systems of a massively complex system is very intractable. This is because many underlying factors impinge upon the efficacy of an integrated system. Such factors may include technological, human, psychological, ethical, contractual, economic, and time-dependent issues pertinent for each sub-system to be integrated. So far, there is no quantitative modeling approach to assess the degree of systems integration. The set-theoretic approach presented in this paper attempts to fill that void. It represents the first step to further research in developing quantitative measures for systems integration. In such further research, one must consider various parameters that characterize a system as may be deterministic, stochastic, dynamic, static, continuous, or discrete so that quantitative modeling of integration can be applied. Simulation modeling can also be developed for assessing integration potential for systems involving several components that are too intractable to study analytically.

REFERENCES


ADVERTIZE YOUR PRODUCTS AND SERVICES IN THE

AFRICAN JOURNAL OF COMPUTING AND ICT

(ISSN 2006 -1781)

FOR DETAILED INFORMATION ON SUBSCRIPTION AND ADVERTIZEMENT RATES, PLEASE CONTACT THE PRINCIPAL CONSULTANT
(dewadeconsult@yahoo.com)
COMPUTATIONAL ANALYSIS OF GSM SECURITY ALGORITHM

O.C. Abikoye  
Department of Computer Science,  
University of Ilorin,  
Ilorin, Kwara State,  
Nigeria  
Email:kemi_adeoye@yahoo.com

&  
‘Dele Oluwade*  
P.O.Box 20253, U.I.P.O.,  
Ibadan, OYO 200005,  
Nigeria  
Email:deleoluwade@yahoo.com

(*Present institutional address: Department of Library and Information Technology, Federal University of Technology, Minna, Niger State, Nigeria)

Received 10 May 2008

Thesis Communication

ABSTRACT

The security and authentication mechanisms incorporated in the Global System of Mobile Network (GSM) make it the most secure mobile communication standard currently available, particularly in comparison to the analog systems. GSM security and encryption algorithm are used to provide authentication and radio link privacy to users on GSM network. Security in GSM consists of the following aspects: subscriber identity authentication, subscriber identity confidentiality, signaling data confidentiality, and user data confidentiality. These various security aspects are addressed by the user authentication algorithm(called A3), the signal encryption algorithm(A5) and the encryption key generation algorithm(A8). Both A3 and A8 algorithms are similar in functionality and are commonly implemented as a single algorithm called COMP128. In this paper, a computational time complexity analysis of COMP 128 is performed. It is shown that the running time of user authentication algorithm COMP128 is constant, meaning that the number of executions of basic operations in COMP128 algorithm is fixed and so the total time is bounded by a constant.

Keywords and Phrases: Computational complexity, GSM, Security, Algorithm, A3, A8, COMP128
I INTRODUCTION

GSM (Global System of Mobile Network) is one of the most widely used mobile telephone system. As communication with a mobile phone occurs over a radio link it is susceptible to attacks that passively monitor the airways (radio paths). The security and authentication mechanisms incorporated in the GSM make it the most secure mobile communication standard currently available, particularly in comparison to the analog systems. GSM security and encryption algorithm are used to provide authentication and radio link privacy to users on GSM network. Security in GSM consists of the following aspects: subscriber identity authentication, subscriber identity confidentiality, signaling data confidentiality, and user data confidentiality. Several security functions were built into GSM to safeguard subscriber privacy. The GSM specification addresses three key security requirements: authentication, confidentiality and anonymity. There are three proprietary algorithms used to achieve authentication and confidentiality. These are known as A3, A5 and A8 [4]. While A3 is used to authenticate the SIM (Subscriber Identity Module) for access to the network (user authentication algorithm), both A5 and A8 achieve confidentiality by scrambling the data sent across the airways; A5 is the signal encryption algorithm and A8 is the encryption key generation algorithm. Anonymity is achieved by use of temporary identities (TMSI). The A3 and A8 algorithms of GSM are key-dependent one-way hash functions. The GSM A3 and A8 algorithms are similar in functionality and are commonly implemented as a single algorithm called COMP128. To the best of the authors’ knowledge, even though these algorithms are very important, no computational analysis has hitherto been performed on them. In this paper, a computational time complexity analysis of the combined algorithm, COMP 128, is performed. This is a GSM standard authentication algorithm which may be used by all networks which do not wish to develop a proprietary algorithm.

Analyzing an algorithm has come to mean predicting the resources that the algorithm requires. Occasionally resources such as memory, communication bandwidth or logic gates are of primary concern but most often it is computational time that is always measured. Generally by analyzing several candidates’ algorithms for a problem, a most efficient one can be easily identified. Such analysis may indicate more than one viable candidate, but several inferior algorithms are usually discarded. The performance of any program/algorithm can be analyzed. By the performance of a program/algorithm we mean the amount of computer memory and time needed to run a program. There are two approaches to determine the performance of a program. One is analytical and the other is experimental. In computational analysis analytical methods are used, while in performance analysis experiments are conducted. The space complexity of a program is the amount of memory it needs to run to completion while the time complexity of a program is the amount of computer time it needs to run to completion [3]. Time complexity is always needed to provide an
upper limit on the amount of time the program will run, a satisfactory real-time response and also to decide on which solution to use if there are many alternatives based primarily on the performance difference among these solutions.

In general, the time taken by an algorithm grows with the size of the input, so it is traditional to describe the running time of a program as a function of the size of its input. The best notion for input size depends on the problem being studied for many problems such as sorting or computing Discrete Fourier Transform, the most natural measure is the number of items in the input. For example, the array size \( n \) for sorting. For many other problems, such as multiplying two integers, the best measure of input size is the total number of bits needed to represent the input in ordinary binary notation. The running time of an algorithm on a particular input is the number of primitive operations or “steps” executed. It is convenient to define the notion of step so that it is as machine-independently as possible.

In [7], the computational complexity of an algorithm useful in numerical analysis was studied and it was shown that the algorithm has a linear time complexity. In another paper [8], an algorithm for finding the error pattern of a uniform digital code was proposed and analyzed. It was shown that the (asymptotic) time complexity of the PASCAL-like algorithm is also of linear time. In the present paper, it is shown that the running time of COMP128 is constant, meaning that the number of executions of basic operations in COMP128 algorithm is fixed and so the total time is bounded by a constant.

II. COMP128 ALGORITHM

In this section, the COMP 128 algorithm is described. The algorithm has been extracted from [4].

(1) (Load RAND into last 16 bytes of input)

\[
\begin{align*}
&\text{FOR } i \text{ from } 16 \text{ to } 31 \\
&\quad X[i] = \text{rand}[i] \\
&\text{END FOR}
\end{align*}
\]

(2.1) (Loop eight times)

\[
\begin{align*}
&\text{FOR } i \text{ from } 1 \text{ to } 8 \\
&\quad I = n \\
&\quad \text{(Load key into first 16 bytes of input)} \\
&\quad \text{FOR } j \text{ from } 0 \text{ to } 15 \\
&\quad\quad X[j] = \text{key}[j] \\
&\quad \text{END FOR}
\end{align*}
\]

(2.2) (Perform substitutions)

\[
\begin{align*}
&\text{FOR } j \text{ from } 0 \text{ to } 4 \\
&\quad \text{FOR } k \text{ from } 0 \text{ to } 2^j - 1 \\
&\quad\quad \text{FOR } l \text{ from } 0 \text{ to } 2^{4j} - 1 \\
&\quad\quad\quad m = 1 + k \times 2^{5j} \\
&\quad\quad\quad n = m + 2^{4j}
\end{align*}
\]
\[ y = (x[m] + 2 * x[n]) \mod 2^0 \]

\[ z = (2 * x[m] + x[n]) \mod 2^9 \]

\[ x[m] = \text{table \[j,y\]} \]
\[ x[n] = \text{table \[j,z\]} \]

\text{END FOR}
\text{END FOR}
\text{END FOR}

(2.3) (Form bits from bytes)
\text{FOR } j \text{ from 0 to 31}
\quad \text{FOR } k \text{ from 0 to 7}
\quad \text{bit \[4*j+k\] = the \(8-k\)th bit of byte \text{j}}
\quad \text{END FOR}
\text{END FOR}

(2.4) (Permutation but not on the last loop)
\text{IF } (i < 8) \text{ THEN}
\quad \text{FOR } j \text{ from 0 to 15}
\quad \quad \text{FOR } k \text{ from 0 to 7}
\quad \quad \text{next bit = } (8 * j + k) \mod 128
\quad \quad \text{Bit } k \text{ of } x[j + 16] = \text{bit[next_bit]}
\text{END FOR}
\text{END FOR}
\text{END IF}

The vector \(x[\ ]\) consists of 32 nibbles, the first 8 of which are taken as the output SRES.

\text{III. COMPUTATIONAL TIME COMPLEXITY OF COMP 128 ALGORITHM}

The time complexity of a program depends on the factors that the space complexity depends on. A program will run faster on a computer capable of executing \(10^9\) instructions per second than on the one that can execute only \(10^7\) instructions per second. Some compilers will take less time than others to generate the corresponding computer code. Smaller problem instances will generally take less time than larger instances.

The time taken by a program \(p\) is the sum of the compile time and the run (execution time). One way to estimate the time complexity of a program or method is to select one or more operations i.e through the use of operation counts. Such as add, multiply and compare and to determine how many of each is done. The success of this method depends on the ability to identify the operations that contribute most to all the time complexity.

Another way to estimate the time complexity of a method or program is through the use of step count. In step count method, we attempt
to account for the time spent in all parts of the program/method. The step count is a function of the instance characteristics. Although any specific instance may have several characteristics (e.g. the number of inputs, the number of outputs, the magnitude of the inputs and outputs), the number of step is computed as a function of some subset of these.

Let \( f(n) \) and \( g(n) \) be two non-negative functions. Then \( g(n) \) is said to be asymptotically bigger than \( f(n) \) iff \( f(n)/g(n) \to 0 \) as \( n \to \infty \) [3]. The notation \( f(n) = O(g(n)) \), where \( O \) is called ‘big oh’, is used when \( f(n) \) is asymptotically smaller than or equal to \( g(n) \) (which means \( g(n) \) is asymptotically greater than or equal to \( f(n) \)) i.e \( g(n) \) is an upper bound for \( f(n) \). More rigorously, \( f(n) = O(g(n)) \) iff there exists positive constants \( c \) and \( n_0 \) such that \( f(n) \leq cg(n) \), for all \( n \geq n_0 \). Apart from the ‘big oh’ which is the most frequently used asymptotic notation, other notations include omega (\( \Omega \)), theta (\( \Theta \)), ‘little oh’ (\( o \)) and little omega (\( \omega \)). Now, \( f(n) = \Omega(g(n)) \) if \( f(n) \) is asymptotically bigger than or equal to \( g(n) \). This means \( g(n) \) is an asymptotic lower bound for \( f(n) \). \( f(n) = \Theta(g(n)) \) iff there exist positive constants \( c_1 \) and \( c_2 \), and \( n_0 \), such that \( c_1g(n) \leq f(n) \leq c_2g(n) \), for all \( n > n_0 \). i.e \( f(n) \) can be bounded both from above and below by the same function \( g(n) \), which implies that both of the following are valid:

\[
\begin{align*}
&f(n) = \Omega(g(n)); f(n) = O(g(n)) \\
&f(n) = \Theta(g(n)) \\
&f(n) = \omega(g(n)); f(n) = o(g(n)) \\
&f(n) = \Omega(g(n)); f(n) = \omega(g(n))
\end{align*}
\]

The computational time complexity analysis of COMP128 algorithm is performed below.

1. Assignment takes a constant time \( \Theta(k) \);
   we have \( n\Theta(k) = \Theta(nk) \) where \( n=16 \)
   \[
   T_1 = \Theta(nk)
   \]

2.1 assignment takes a constant time \( \Theta(k) \)
   \[
   n \cdot j \Theta(k) \text{ where } n=8 \text{ and } j = 16
   \]
   which give at most \( \Theta(n^2 k) \)
   \[
   T_2 = \Theta(n^2 k)
   \]

2.2 assignment, addition, multiplication and modulo each take constant times \( \Theta(k) \),
   where \( n \cdot j \cdot k \cdot l \Theta(k) = \Theta(n^4 k) \),
   \[
   j = 4, k = 2^4 = 16, l = 2^4 = 16
   \]
   \[
   T_3 = \Theta(n^4 k)
   \]

2.3 assignment takes a constant time \( \Theta(k) \)
   where \( n \cdot j \cdot k \Theta(k) = \Theta(n^3 k) \) where \( n = 8, k = 8, j = 32 \)
   \[
   T_4 = \Theta(n^3 k)
   \]

2.4 assignment, addition, multiplication modulo take constant times \( \Theta(k) \)
   where \( n \cdot j \cdot k \Theta(k) = \Theta(n^3 k) \) where \( j=16, n = 8, k = 8 \)
   \[
   T_5 = \Theta(n^3 k)
   \]
Finally, \( T(n) = T_1 + T_2 + T_3 + T_4 + T_5 \)

\[
= \Theta(nk) + \Theta(n^2k) + \Theta(n^4k) + \Theta(n^3k) + \Theta(n^5k)
\]

\[
= \Theta(n^4k) \quad \text{where} \quad n \leq 32
\]

\[
= O(n^4k) \quad n = 32
\]

since \( n \) has an upper bound which is 32 i.e we cannot have more than 32 loop and so the running time is constant.

\[
T(n) = \Theta(k)
\]

**IV CONCLUSION AND DISCUSSION**

The security mechanisms specified in the GSM standard make it the most secure cellular telecommunications system available. The use of authentication, encryption, and temporary identification numbers ensures the privacy and anonymity of the system’s users, as well as safeguarding the system against fraudulent use. Even GSM systems with the A5/2 encryption algorithm or even with no encryption are inherently more secure than analog systems due to their use of speech coding, digital modulation, and TDMA channel access.

The running time of user authentication COMP128 algorithm is constant, \( T(n) = \Theta(k) \), meaning that the number of executions of basic operations in COMP128 algorithm is fixed, hence, the total time is bounded by a constant.

The total memory requirement for the data and program storage of COMP128 algorithm depends on the total bytes occupied by the number of microprocessor instructions codes, the bytes of external memory, the bytes occupied by the variables used in the instruction codes.

In conclusion, GSM has experienced a lot of security breaches, despite all the breaches GSM is by far more secure than previous analog cellular systems and continues to be the most secure public wireless standard in the world.

**POSTSCRIPT**

This paper is part of the M.Sc (Computer Science) project of the first author [1] written under the supervision of the second author.

**REFERENCES**


[6] David Margrave, George Mason University, "GSM Security and Encryption";
Abikoye & Oluwade, Computational Analysis  33


ADVERTIZE YOUR PRODUCTS AND SERVICES IN THE

AFRICAN JOURNAL OF COMPUTING AND ICT

(ISSN 2006-1781)

FOR DETAILED INFORMATION ON SUBSCRIPTION AND ADVERTIZEMENT RATES, PLEASE CONTACT THE PRINCIPAL CONSULTANT
(dewadeconsult@yahoo.com)
AN EVALUATION OF CHIDAMBER AND KEMERER’S LACK OF COHESION IN METHOD (LCOM) METRIC USING DIFFERENT NORMALIZATION APPROACHES

1Ezekiel Okike, Member IEEE, and 2Adenike Osofisan

Department of Computer Science,
University of Ibadan,
Nigeria.
E-mail:1euokike@gmail.com;
2ao.osofisan@mail.ui.edu.ng

Received 02 June 2008
Communicated by Amos David

ABSTRACT

Chidamber and Kemerer first defined a cohesion measure for object-oriented software – the Lack of Cohesion in methods (LCOM) metric. However, the LCOM metric is not a normalized measure, hence the measure does not satisfy all the properties for cohesion measures. This paper presents three approaches which could be used to normalize the LCOM metric. The approaches are the Sigmoid LCOM (SLCOM) normalization, the Bowles LCOM (BLCOM) normalization and the Best-Fit LCOM (BFLCOM) normalization. These approaches were evaluated using field data from a Java based industrial system. Class cohesion was measured using the LCOM metric. The three normalization approaches were applied to the measured LCOM metric values. The result of the study suggests that the Sigmoid LCOM may not be a good transformation for the LCOM metric, the Bowles and Best-fit normalized LCOM measures may be used with the LCOM, while the Best-fit LCOM seem to be the best normalization approach for the LCOM metric.

Keywords—Normalization of measures, Class cohesion, LCOM metric.
I. INTRODUCTION

The term cohesion is defined as the “intramodular functional relatedness” in software [1]. This definition, considers the cohesion of each module in isolation: how tightly bound or related its internal elements are. Hence, cohesion as an attribute of software modules capture the degree of association of elements within a module, and the programming paradigm used determines what is an element and what is a module. In the object-oriented paradigm, for instance, a module is a class and hence cohesion refers to the relatedness among the methods of a class. Usually, cohesion is categorized ranging from the weakest form to the strongest form in the following order: coincidental, logical, temporal, procedural, communicational, sequential and functional.

A coincidentally cohesive module is one whose elements contribute to activities with no meaningful relationship to one another. An example is to have unrelated statements bundled together in a module. Such a module would be hard to understand what it does and can not be reused in another program. A logically cohesive module is one whose elements contribute to activities of the same general category in which the activity or activities to be executed are selected from outside the module. A logically cohesive module does any of several different related things, hence, presenting a confusing interface since some parameters may be needed only sometimes. A temporally cohesive module is one whose elements are involved in activities that are related in time. That is, the activities are carried out at a particular time. The elements occurring together in a temporally cohesive module do diverse things and execute at the same time. A procedurally cohesive module is one whose elements are involved in different and possibly unrelated activities in which control flows from each activity to the next. Procedurally cohesive modules tend to be composed of pieces of functions that have little relationship to one another (except that they are carried out in a specific order at a certain time). A communicationally cohesive module is one whose elements contribute to activities that use the same input or output data. A sequentially cohesive module is one whose elements are involved in activities such that output data from one activity serve as input data to the next. Some authors identify this as informational cohesion. A functionally cohesive module contains elements that all contribute to the execution of one and only one problem-related task. The elements do exactly one thing or achieve one goal. A module exhibits one of these forms of cohesion depending on the skill of the designer. However, functional cohesion is generally accepted as the best form of cohesion in software design. Functional cohesion is the most desirable because it performs exactly one action or achieves a single goal. Such a module is highly reusable, relatively easy to understand (because you know what it does) and is maintainable. In this paper, the term “cohesion” refers to functional cohesion. Functional cohesion is the most desirable in software design. This is because a functionally cohesive module performs exactly one action or achieves a single goal. Such a module is highly reusable, relatively easy to understand (because you know what it does) and is maintainable. Several measures of cohesion have been defined in both the procedural and object-oriented paradigms. Most of the cohesion measures defined in the object-oriented paradigm are inspired from the Lack of Cohesion in methods (LCOM) metric defined by Chidamber and Kemerer. However, the major problems with the LCOM metric are the presence of outliers and the fact that the metric values are not normalized values. In this paper, the Lack of Cohesion in methods (LCOM) metric is critically examined. Three normalization approaches are used to evaluate the metric, out of which two are proposed for use with the LCOM metric. The rest of the paper is organized as follows:
Section 2 presents a summary of the approaches to measuring cohesion in procedural and object-oriented programs. Section 3 examines the Chidamber and Kemerer LCOM metric. Section 4 examines three normalization approaches for the LCOM metric. Section 5 concludes the paper and suggests that the Best-fit Normalized LCOM (BFNLCOM) and the Bowles normalized LCOM (BNLCOM) approaches may be used with the LCOM metric. However, the BFNLCOM seem to be the best normalized LCOM approach for the LCOM metric.

II. MEASURING COHESION IN PROCEDURAL AND OBJECT-ORIENTED PROGRAMS

II.1 Measuring cohesion in procedural programs

Procedural programs are those with procedure and data declared independently. Examples of purely procedure oriented languages include C, Pascal, Ada83, Fortran and so on. In this case, the module is a procedure and an element is either a global value which is visible to all the modules or a local value which is visible only to the module where it is declared. As noted in [2], the approaches taken to measure cohesiveness of this kind of programs have generally tried to evaluate cohesion on a procedure by procedure basis, and the notational measure is one of “functional strength” of procedure, meaning the degree to which data and procedures contribute to performing the basic function. In other words the complexity is defined in the control flow. Among the best known measures of cohesion in the procedural paradigm are discussed in [3] and [4].

II.2 Measuring cohesion in object-oriented systems

In the Object Oriented languages, the complexity is defined in the relationship between the classes and their methods. Several measures exist for measuring cohesion in Object-Oriented systems [7,8,9,10,11,12]. Most of the existing cohesion measures in the object-oriented paradigm are inspired from the Lack of Cohesion in Methods (LCOM ) metric [5,6]. Some examples include LCOM3, Connectivity model, LCOM5, Tight Class Cohesion (TCC), and Low Class Cohesion (LCC), Degree of Cohesion in class based on direct relation between its public methods (DCD) and that based on indirect methods (DCI), Optimistic Class cohesion (OCC) and Pessimistic Class Cohesion (PCC). In the following section, we examine the desirable properties of cohesion measures, the LCOM metric and propose three normalization strategies for the metric.

II.3 Desirable properties of cohesion measures

The following are the basis for the list of properties of cohesion measures [19]:

(i) Normalization: A module can have no cohesion or can have maximum cohesion. Intuitively, cohesion can be expected to be non-negative and normalized so that the measure is independent of the size of the module. A cohesion metric for a given module must be in a specific interval [0, max]. Cohesion (module) = 0 if there is no relationship between the module components and there is no evidence that they should be encapsulated together. Cohesion (module) = max if and only if all components of the module are maximally related. Normalization thus allows for meaningful comparisons between the cohesion of different modules as the value of cohesion would always be within the same interval.

(ii) Non-decreasing monotonicity for non-cohesive components: Adding intra-module relationships cannot decrease the module cohesion. Additional internal relationships in modules cannot decrease cohesion since they are supposed to be additional evidence to encapsulate module components together.

(iii) Cohesive modules: The cohesion of a module obtained by putting together two
unrelated modules is not greater than the maximum of the two original modules. When two or more modules showing no relationships between them are merged, the cohesion cannot increase because seemingly unrelated elements are encapsulated together.

(iv) Non-negativity: The cohesion of a module cannot be negative. Even in the worst case, when none of the module components are related, the cohesion of the module is 0.

(v) Symmetry: The cohesion of a module does not depend on the convention chosen to represent the relationship between its elements. The cohesion of a module should not be sensitive to representation convention with respect to the direction of the arrow representing component relationships. A relation between module components m1 and m2, represented as either m1→m2 or m2→m1 is equivalent. The relationships between the module elements are not affected by these two equivalent conventions, so a cohesion measure should be insensitive to this.

Other properties include [18]:

(vi) Renaming: If P is a renaming of Q, cohesion (P) = cohesion (Q). According to this property, renaming does not affect cohesion

(vii) Transitive: Consider modules m1, m2 and m3 such that,

cohesion(m1) < cohesion(m2), and

cohesion(m2) < cohesion(m3), then

cohesion(m1) < cohesion(m3).

(viii) A Cohesion measure should not be “too coarse”, that is, there exist modules m1 and m2 such that cohesion(m1) ≠ cohesion(m2). According to this property, a measure should be able to distinguish between the cohesion of two or more modules.

III. THE LACK OF COHESION IN METHODS (LCOM) METRIC.

The LCOM metric is based on the number of disjoint sets of instance variables that are used by the method. Its definition is given as follows [6].

Definition.

Consider a class C1 with n methods M1, M2,..., Mn. Let \{Ii\} = set of instance variables used by method Mi.

There are n such sets \{I1\}, ..., \{In\}. Let P = \{(Ii, Ij) | Ii ∩ Ij = ∅\} and Q = \{(Ii, Ij) | Ii ∩ Ij ≠ ∅\}. If all n sets \{I1\}, ..., \{In\} are φ then let P = φ

\[ \text{LCOM} = \begin{cases} |P| - |Q|, & \text{if } |P| > |Q| \\ 0, & \text{otherwise} \end{cases} \]

Example: Consider a class C with three methods M1, M2 and M3. Let \{I1\} = \{a,b,c,d,e\} and \{I2\} = \{a,b,e\} and \{I3\} = \{x,y,z\}. \{I1\} ∩ \{I2\} is nonempty, but \{I1\} ∩ \{I3\} and \{I2\} ∩ \{I3\} are null sets. LCOM is (the number of null intersections – number of non empty intersections), which in this case is 1.

The theoretical basis of LCOM uses the notion of degree of similarity of methods. The degree of similarity of two methods M1 and M2 in class C1 is given by:

\[ σ( ) = \{I1\} ∩ \{I2\} \]

where \{I1\} and \{I2\} are sets of instance variables used by M1 and M2. The LCOM is a count of the number of method pairs whose similarity is 0 (i.e., σ( ) is a null set) minus the count of method pairs whose similarity is not zero. The larger the number of similar methods, the more cohesive the class, which is consistent with the traditional notions of cohesion that measure the inter relatedness between portions of a program. If none of the methods of a class display any instance behaviour, i.e. do not use any instance variables, they have no similarity and the LCOM value for the class will be zero. The LCOM value provides a measure of the relative disparate nature of methods in the class. A smaller number of disjoint pairs (elements of set P) implies greater similarity of methods. LCOM is intimately tied to the instance variables and methods of a class,
and therefore is a measure of the attributes of an object class.

pairs of methods in a class having no common attribute references, \(|P|\), minus the number of pairs of similar methods, \(|Q|\). However, if \(|P| < |Q|\), LCOM is set to zero. Again, it is not stated whether inherited methods and attributes are included or not. The above definition is refined in [14] as follows:

**Definition 4.**

Let \(P = \phi\), if \(AR(m) = \phi \forall m \in M_I(c)\)

\[\{\{m_1,m_2\} | m_1,m_2 \in M_I(c) \land m_1 \neq m_2 \land AR(m_1) \cap AR(m_2) \cap A_I(c) = \phi\}, \] else

Let \(Q = \{\{m_1,m_2\} | m_1,m_2 \in M_I(c) \land m_1 \neq m_2 \land AR(m_1) \cap AR(m_2) \cap A_I(c) \neq \phi\}\)

Then \(LCOM_2(c) = |P| - |Q|, if |P| > |Q|; = 0\) otherwise

The definition of LCOM2 has been widely discussed in the literature [6,9,11,14,19]. LCOM2 of many classes are set to be zero although different cohesions are expected.

**III.1 Remarks**

In general the Lack of Cohesion in Methods (LCOM) measures the dissimilarity of methods in a class by instance variable or attributes. Chidamber and Kemerer’s interpretation of the metric is that LCOM = 0 indicates a cohesive class. However, for LCOM >0, it implies that instance variables belong to disjoint sets. Such a class needs splitting into 2 or more classes to make it cohesive.

**III.2 Venn diagram illustrations of different ways methods share attributes**

Cases 1-5 below provide illustrations of different ways the methods of a class may share instance variables. For simplicity of understanding, venn diagrams are used in these illustrations. Each Venn diagram represents a method by the set of instance variables it uses. The ovals represent the methods and the small circles in the ovals represent instance variables.
3.2. A pair of methods $<M2, M3>$ that do share instance variables.

**Figure 3.3.** A pair of methods $<M1, M2>$; $<M3, M4>$ share an instance variable each.
In the case of figure 3.2, there are two disjoint sets of methods. There are three methods in the class namely M1, M2, M3, only two of which (M2 and M3) share an instance variable. There are two pairs of methods that do not share instance variables <M1,M2>, <M1,M3>. Therefore the number of pairs of disjoint sets P = 2. There is one pair of methods (M2,M3) that share an instance variable. Hence Q = 1. Therefore LCOM = P - Q = 2 - 1 = 1

**Case 3:** An illustration of four methods M1, M2, M3, M4 where a pair of methods M1 and M2; M3 and M4 share an instance variable each.

As before, in Figure 3.3, there are two disjoint sets of methods where each set has a pair of methods that share an instance variables. There are four pairs of methods that do not share instance variables namely <M1,M3>, <M1,M4>, <M2,M3>, and <M2,M4>. Hence P = 4. There are two pairs of methods that share an instance variable namely <M1,M2>, <M3,M4>. Hence Q = 2. Therefore LCOM = 4 - 2 = 2

**Case 4:** An illustration of a class with four methods M1,M2,M3,M4 where only three of the four methods M2,M3,M4 share an instance variable.

In the case of figure 3.4, there are two disjoint sets of methods where there is one method that does not share instance variables with any method and the other three methods share an instance variable. There are three pairs of methods that do not share instance variables namely <M1,M2>, <M1,M3>, <M1,M4>. Hence P = 3. There are also three pairs of methods that share an instance variable namely <M2,M3>, <M2,M4>, <M3,M4>. Hence Q = 3. Therefore LCOM = 3 - 3 = 0.
Case 5. (a): Three sequentially linked methods shared by instance variables

![Diagram of three sequentially linked methods](image)

**Figure 3.5**. Three Sequentially linked methods

Case 5 (b): The minimally cohesive class, where n methods are sequentially linked by shared instance variables.

![Diagram of n sequentially linked methods](image)

**Figure 3.6.** N Sequentially liked methods
Consider three sequentially linked methods M1, M2, M3 as in figure 3.5, case 5a above. Methods M1 and M2 share an instance variable, and methods M2 and M3 share an instance variable. Method M2 shares different instance variable with methods M1 and M3. There is one pair of methods that do not share any instance variable, namely, method pair <M1, M3>. Hence, P = 1. There are two pairs of methods that share an instance variable, namely <M1,M2>, <M2,M3>. Hence, Q = 2. Therefore, \( LCOM = 1 - 2 = -1 \). However, from the definition of LCOM, \( LCOM = 0 \).

If a case with five methods sequentially linked such as case 5b figure 3.6 above is considered, there will be six pairs of methods that do not share an instance variable namely <M1,M3>, <M1,M4>, <M1,M5>, <M2,M4>, <M2,M5>, <M3,M5>. Hence, P = 6. There are four pairs of methods that share an instance variable, namely <M1,M2>, <M2,M3>, <M3,M4>, <M4,M5>. Hence, Q = 4. Therefore, \( LCOM = 6 - 4 = 2 \).

### III.2.1 Remark

In general

\[
P = \left(\frac{n}{2}\right) - (n - 1)
\]

(1)

\[
Q = n - 1
\]

(2)

so that LCOM

\[
P - Q = \left(\frac{n}{2}\right) - 2(n - 1)
\]

(3)

**Explanation**

From (1) and (2)

\[
P - Q = \left(\frac{n}{2}\right) - (n - 1) - (n - 1)
\]

\[
= \left(\frac{n}{2}\right) - 2n - 2
\]

\[
= \left(\frac{n}{2}\right) - 2(n - 1)
\]

\[
= \frac{n!}{(n - 2)!2!} - 2(n - 1)
\]

(4)

From (4), for \( n < 5 \), \( LCOM = 0 \) indicating that classes with less than 5 methods are equally cohesive. For \( n \geq 5 \), \( 1 < LCOM < n \), implying that classes with 5 or more methods need to be split.

### III.3 Class design and LCOM computation

Figure 3.7 presents a class x written in C++.

```
Class x {
    Int A,B,C,D,E,F;
    Void f() {…uses A,B,…}
    Void g() {…uses D,E,…}
    Void h() {…uses E,F,…}
};
```

**Figure 3.7.** Class design showing LCOM computation
The Lack of Cohesion in Methods (LCOM) for class \( x = 1 \), calculated as follows:

There are two pairs of methods accessing no common instance variables namely \(<f, g>, <f, h>\). Hence \( P = 2 \). One pair of methods shares variable \( E \), namely, \(<g, h>\). Hence, \( Q = 1 \). Therefore, LCOM is \( 2 - 1 = 1 \).

### III.4 Critique of LCOM metric

The LCOM metric has been criticized for not satisfying all the desirable properties of cohesion measures. For instance, the LCOM metric values are not normalized values \([11,13]\). In addition, this metric is not able to distinguish between the structural cohesiveness of two classes, in the way in which the methods share instance variables \([8]\). Consider figures 3.1-3.6 (cases 1-5). According to Chidamber and Kemerer’s metric, the classes in case 1 and case 2 should be split since for these cases \( \text{LCOM} = 1 \). The class in case 3 definitely needs to be split (\( \text{LCOM} = 2 \)) while the class in case 4 is considered to be cohesive class (\( \text{LCOM} = 0 \)) and hence does not need to be split. Adding a method to an existing method as in case 1 resulting in case 2, however, did not change the cohesion of the class. Adding a method that shares an instance variable with existing method should decrease the value of LCOM. However, adding a method to case 2 give conflicting results. In case 3 it increases the LCOM and in case 4 it decreases it. Also in case 5, where \( n \) methods are sequentially “linked” by shared instance variable, an examination of different values of \( n \) indicate that LCOM varies with the value of \( n \). For \( n \geq 5 \), the dynamic increase in the value of LCOM when increasing the number of methods by one does not seem to be justifiable. Hence, a connectivity metric to be used in conjunction with the LCOM metric is proposed in \([8]\). The value of the connectivity metric always lies between 0 and 1. In this paper, we address the problems of outliers and un normalized values common with the LCOM metric. We apply normalization techniques to the LCOM metric and define a Best-Fit normalized LCOM (BFNLCOM) whose value always lies between 0 and 1.

### IV. NORMALIZATION OF SOFTWARE MEASURES

#### IV.1 Related Work

The normalization of measures is an important aspect of software measurement. Measures with outlier values need to be normalized in order to get numbers between \([0,1]\). However, as stated in \([15]\) the normalization of a measure \( u \) to a normalized measure \( u' \) can lead to a completely new measure with other properties.

In \([16]\) a measure \( IS \) is defined in order to capture the complexity of a module derived from the interactions (coupling) with other modules. The measure \( IS \) was the original measure. In order to get numbers between 0 and 1 the measure \( CM \) was defined as follows:

\[
CM = 1 - \frac{1}{1 + IS}
\]

where \( CM = \text{Module Complexity}, IS = \text{Complexity from coupling} \)

In this normalization, the obtained values for \( CM \) are in the range \([0,1]\).

In neural networks a Sigmoid neural layer uses the Sigmoid function to determine its activation. The sigmoid function \( y(u) \) is defined as follows:

\[
y(u) = \frac{1}{1 + e^{-u}} ; \text{(the usual value of} \ e = 2.7183, u = x \text{ _values)}
\]

One thing about the sigmoid layer is that only positive values are returned between 0.5 and 1. The Sigmoid function is so called because it looks like an \text{s} as shown in figure 4.1.
When applied to software measurement, a sigmoid layer for the attribute being measured may be identified only if the expected transformations are in the range [0.5,1]. Although both the approach in [16] and the Sigmoid approach may be used to normalize a measure between the ranges [0,1] and [0.5,1] respectively, in [15] the following important questions were suggested in order to decide whether to use a normalization approach:

(i) Do the measure being normalized measure the same qualitative aspect?
(ii) Is the normalizing measure really a normalization of the measure being normalized?
(iii) After normalization, does the normalization agree with the interpretations of the original measure?

IV.2 Normalizing Chidamber and Kemerer LCOM metric

The major problems with Chidamber and Kemerer’s LCOM metric are the presence of outliers and the fact that the measure is not normalized (standardized).

(a) LCOM is not a standardized (normalized) measure

In correlation analysis, correlation measures the degree or strength of relationship between two quantities say x, and y. However, because the two quantities could be measured in different units, there is need to standardize the unit of measurement. To achieve this, the correlation Cor (x, y) is divided by the probability prob (x,y):

\[
Cor.(x,y) \div prob(x,y) = -1 \leq prob(x,y) \leq 1
\]

The resulting value is said to be standardized or normalized and lie between -1 and 1. However cohesion measures are non negative, hence standardization (normalization) with regards to cohesion should lie between 0 and 1. The values obtained from LCOM are most often outside the standard range, and therefore are not normalized values. Table 4.1 below contains results using Chidamber and Kemerer metrics taken from a Java based system as reported in [17]. In the table, the metrics from Chidamber and Kemerer suite are [6]: Lack of Cohesion in methods (LCOM), Coupling Between Object Classes (CBO), Response For a Class (RFC), Weighted Methods Per Class (WMC), Depth of Inheritance (DIT) and Number of Children (NOC). Two other metrics are used (not part of the Chidamber and Kemerer metrics) namely: Number of Public Methods (NPM) and Afferent Coupling (CA) [20]. The choice of these metrics as additional to the Chidamber and Kemerer suite of metrics is informed by the need to have a metric to measure the number of public methods in a class as well as the number of other classes using a specific class.

In table 4.1 the LCOM values are highlighted. These values are not standardized (normalized) and have outliers. For instance the classes with cohesion (LCOM) values 91, 370, 78, and even 6075. According to Chidamber and Kemerer, classes whose LCOM values are not zero needs to be split into two or more classes to
make them cohesive. However, we point out that in reality this is not always the case. At best what you can achieve by splitting an outlier class is a reduced outlier value. In addition, the authors in [8] queried why a class with LCOM=1 should be split? In other words, LCOM=1 could be a cohesive class which the LCOM metric seem to have ignored.

**Table 4.1.** Illustration of LCOM values with outliers and not standardized (unnormalized)

<table>
<thead>
<tr>
<th>Class Name</th>
<th>WM</th>
<th>DIT</th>
<th>NOC</th>
<th>CBO</th>
<th>RFC</th>
<th>LCOM</th>
<th>CA</th>
<th>NPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>XlmException</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>HandlerBase</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>91</td>
<td>0</td>
<td>14</td>
</tr>
<tr>
<td>SAXDriver</td>
<td>33</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>86</td>
<td>370</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>XmlHandler</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>78</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>XmlParser</td>
<td>11</td>
<td>1</td>
<td>8</td>
<td>18</td>
<td>2</td>
<td>607</td>
<td>5</td>
<td>27</td>
</tr>
<tr>
<td>JspXMLParser</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>18</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Compiled Exception</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>JspServlet$Page</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>14</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>JspServlet$MapEntry</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>JspMsg</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>JspFactory$Impl$1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Source:** [17]

Interpreting this result using Chidamber and Kemerer’s approach shows that of the 10 classes, 4 are cohesive (i.e LCOM = 0) while 6 classes are not cohesive. Following from the earlier explanation, we suggest that cohesiveness (LCOM) ∈ [0,1] since there is no basis for splitting a class whose LCOM=1 and this agrees with the view in [8]. Furthermore, to normalize or standardize Chidamber and Kemerer’s LCOM metric, we define and evaluate three normalization approaches as explained in the following section.

### IV.2.1 Bowles Normalized LCOM

Applying Bowless approach to LCOM normalization, a Bowless normalized function for the Chidamber and Kemerer LCOM metric (BNLCOM) may be defined as follows:

$$BNLCOM = 1 - \left( \frac{1}{1 + LCOM} \right).$$

The measure BNLCOM (BowlessNormalized LCOM) is a numerical modification of the measure LCOM (Lack of Cohesion in Methods) without changing the empirical meaning of the measure. It measures the same qualitative statement. Table 4.2 below shows results of applying the BNLCOM measure on sample class data from a Java based system. The LCOM and BNLCOM columns are highlighted for comparison as shown below. The BNLCOM column values are normalized values. Hence $0 \leq BNLCOM \leq 1$ is a valid.
Since LCOM is an inverse measure, the smaller the value of LCOM, the higher the cohesion. This is in line with Chidamber and Kemerer’s interpretation of the LCOM. Cohesion values of the BNLCOM are normalized values, hence BNLCOM normalization is appropriate for the LCOM metric. However, considering the nature of the transformation of 1 to 0.5, the classes with LCOM =1 may have been affected (outlier transformation not important).

IV.2.2 Sigmoid Normalized LCOM (SNLCOM)

Using a Sigmoid function, a Sigmoid normalized cohesion (SNLCOM) for Chidamber and Kemerer’s Lack of Cohesion in Methods (LCOM) is defined as the LCOM metric whose values are passed through the Sigmoid function layer. Using a sigmoid function such as \( y(u) = \frac{1}{1+e^{-u}} \), the Sigmoid normalized cohesion metric for LCOM (SNLCOM) is defined as follows:

\[
SNLCOM = Y(LCOM) = \frac{1}{1+e^{-LCOM}}
\]

The result of applying the sigmoid normalized cohesion is shown in table 4.3. Although the SNLCOM column is normalized, the resulting values are in the range [0.5,1] as in table 4.3. This is not considered as appropriate normalization for the Chidamber and Kemerer LCOM metric because of the nature of the transformations where 0 is transformed to 0.5 as this will affect the authors interpretation of the LCOM metric. A Best-fit normalized LCOM (BFNLCOM) considered very appropriate for the Chidamber and Kemerer LCOM metric is proposed in this paper as discussed in section IV.2.3.
Table 4.3 Illustration of Sigmoid normalized LCOM (SNLCOM) cohesion with standardized (normalized) values and no outliers for LCOM values. Hence 0.5 ≤ SLCOM ≤ 1.

<table>
<thead>
<tr>
<th>Class Name</th>
<th>W</th>
<th>M</th>
<th>T</th>
<th>O</th>
<th>C</th>
<th>B</th>
<th>F</th>
<th>R</th>
<th>LCOM</th>
<th>SNLCOM</th>
<th>C</th>
<th>A</th>
<th>P</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmException</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td>0.8</td>
<td>807</td>
<td>98</td>
<td>1</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>HandlerBase</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>91</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAXDriver</td>
<td>33</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>86</td>
<td>37</td>
<td></td>
<td></td>
<td>1</td>
<td>0</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XmlHandler</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>78</td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XmlParser</td>
<td>11</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>2</td>
<td>75</td>
<td>1</td>
<td>1</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JspXMLParser</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>18</td>
<td>0</td>
<td></td>
<td></td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CompiledException</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td>0.7</td>
<td>310</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>JspServletPage</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>14</td>
<td>15</td>
<td>4</td>
<td></td>
<td></td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JspServletMapEntry</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JspMsg</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>0.5</td>
<td>9</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JspFactoryImpl$1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td>0.7</td>
<td>310</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Source: [17]

IV.2.3 Proposal of a Best-Fit Normalized LCOM (BFNLCOM)

According to Chidamber and Kemerer’s definition of LCOM:

\[ \text{LCOM} = \begin{cases} \frac{|P| - |Q|}{|P|}, & \text{if } |P| > 0 \\ 0, & \text{otherwise} \end{cases} \]

LCOM = 0 indicates a cohesive class while for LCOM ≥ 1 such classes may be split into two or more classes to make them cohesive. A further interpretation of the metric using descriptive statistics suggests that for cohesive classes, a low median value of 0.00 means that at least half of the classes have cohesive methods. In this work, a median value of 1 is considered also low to be a cohesive class. A normalized LCOM measure which agrees with this interpretation is defined as follows:

\[ \text{BFNLCOM} = \begin{cases} 0, & \text{LCOM} = 0 \\ \frac{1}{\text{LCOM}}, & \text{otherwise} \end{cases} \]

Using the BFNLCOM (Best-Fit Normalized LCOM) metric, it was possible to eliminate the presence of unexplainable outliers from Chidamber and Kemerer’s LCOM metric and also standardize (normalize) the LCOM values in agreement with Chidamber and Kemerer’s interpretation of the LCOM metric in this work. Table 4.4 shows the result of applying the BFNLCOM metric as defined above to normalize the LCOM values.
Table 4.4 Illustration of Best-Fit Normalized LCOM (BFNLCOM) cohesion with standardized (normalized) values and no outliers for LCOM values.

<table>
<thead>
<tr>
<th>Class Name</th>
<th>W</th>
<th>M</th>
<th>C</th>
<th>I</th>
<th>T</th>
<th>N</th>
<th>O</th>
<th>C</th>
<th>B</th>
<th>F</th>
<th>C</th>
<th>RF</th>
<th>LCOM</th>
<th>BFNLCOM</th>
<th>CA</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmEEception</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.5</td>
<td>0.0109</td>
<td>0.027</td>
<td>0.027</td>
</tr>
<tr>
<td>HandlerBase</td>
<td>14</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>SAXDriver</td>
<td>33</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>86</td>
<td>32</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XmlHandler</td>
<td>13</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
<td>0.0128</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>XmlParser</td>
<td>11</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>18</td>
<td>60</td>
<td>75</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>0.0016</td>
<td>1.0</td>
<td>27</td>
</tr>
<tr>
<td>JspXMLParser</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>18</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CompiledEception</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>JspServletPage</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>14</td>
<td>15</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>JspServletMapEntry</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>JspMessage</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>JspFactoryImplS1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Source [17]

This paper is the first to apply the BFNLCOM and BNLCOM normalizations in order to eliminate outliers from Chidamber and Kemerer’s LCOM metric, and hence make the LCOM values much like the normalized cohesion values in [8,10,11]. The benefit of applying BFNLCOM normalization to Chidamber and Kemerer’s LCOM metric is not only that the BFNLCOM standardizes (normalizes) LCOM values, and remove outlier values, BFNLCOM values retain classes with LCOM=0 and LCOM =1 in the resulting transformation. This kind of transformation is considered much better than the others. Hence the admissible value of cohesion for BFNLCOM is the same for the original LCOM. Hence, LCOM, BFNLCOM ∈ [0,1]. For instance, from table 4.4 the outlier values 91, 370, 78, and 6075 were completely reduced to ≤ 1.

4.2.4 Comparison of BFNLCOM, BNLCOM and SNLCOM with LCOM

Table 4.5 gives the values obtained after applying the normalization approaches discussed in this chapter.
### Table 4.5 Comparing normalization approaches for LCOM

<table>
<thead>
<tr>
<th>Class Name</th>
<th>LCOM</th>
<th>BFNLCOM</th>
<th>BNLCOM</th>
<th>SNLCOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlExceptiion</td>
<td>2</td>
<td>0.5</td>
<td>0.666</td>
<td>0.880</td>
</tr>
<tr>
<td>HandlerBase</td>
<td>91</td>
<td>0.0109</td>
<td>0.989</td>
<td>1</td>
</tr>
<tr>
<td>SAXDriver</td>
<td>370</td>
<td>0.0027</td>
<td>0.997</td>
<td>1</td>
</tr>
<tr>
<td>XmlHandle</td>
<td>78</td>
<td>0.0128</td>
<td>0.987</td>
<td>1</td>
</tr>
<tr>
<td>XmlParser</td>
<td>607</td>
<td>0.0001</td>
<td>0.999</td>
<td>1</td>
</tr>
<tr>
<td>JspXMLParser</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>CompiledExcep</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.731</td>
</tr>
<tr>
<td>JspServlet$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>JspServlet$MapEntry</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>JspM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>JspFactoryImpl$</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.731</td>
</tr>
</tbody>
</table>

The table gives the LCOM, BFNLCOM, BNLCOM and SNLCOM values obtained from some ten classes taken from samples of the Java based system used in this paper. The BFNLCOM transformation best fits the LCOM values (outlier value transformation not important). This is because both BFNLCOM and LCOM support earlier explanation that for cohesiveness, BFNLCOM, LCOM ∈ [0,1], as there is no justification to consider a class whose LCOM = 1 un cohesive. This paper agrees with other researchers that cohesion values lie within a range [0,1]. The inclusion of LCOM = 1 as cohesive in this work is explained by the fact that there was no evidence to the fact that the class whose LCOM = 1 was improperly designed [17]. A critical observation of the BFNLCOM, BNLCOM, and SNLCOM columns reveal that the BFNLCOM normalizations is the best normalization approaches for LCOM. SNLCOM approach is not appropriate since using this approach the cohesive range is [0.5,1]. Using BFNLCOM and BNLCOM normalization transformations, Chidamber and Kemerer’s original interpretation of their LCOM metric suggesting that a median value of zero (using descriptive statistics) indicates that at least half of the classes have cohesive methods is retained. However this paper further suggests that the median ∈ [0,1] for cohesive classes.

### IV.3 Outliers in LCOM metric

Table 4.1 shows LCOM values with outliers. Chidamber and Kemerer had suggested that when a class is not cohesive the class should be split into two or more classes to make the class cohesive. In this paper, splitting outlier classes only reduced the outlier values and never made such classes cohesive, and there are no evidences to suggest that such classes were improperly designed. This is illustrated by taking data from the first eight classes of the studied system in the experimental study (Okike and Osofisan, 2007) as shown in figures 4.7 (a) and (b). The fourth class with LCOM value 196 was used as highlighted in figure 4.7 (a) below. The class was split into two classes (ClassMetrics and Classinfo) to see if the outlier value 196 would become a cohesive class in the 0 or even 1 range. The result as shown in figure 4.7 (b) indicated that the outlier value was only reduced to 63 while the LCOM value for Classinfo was 34 which is still an outlier and un cohesive. Figures 4.7(a) and 4.7 (b) show the output after running a Chidamber and Kemerer metric tool ckjm discussed in [20] on the selected classes. After the word ckjm is the class name being analyzed by the tool followed by the corresponding metrics for the class: WMC, DIT, NOC, CBO, RFC, LCOM, CA , NPM in that order. The metric values after the class name correspond to these metrics respectively.
[eokike@visitor2 build]$ java -jar ckjm-1.6.jar
/tmp/gr/spinellis/ckjm/*.class
gr.spinellis.ckjm.ClassMetricsContainer 3 1 0 3 18 0 2 2
gr.spinellis.ckjm.MethodVisitor 11 1 0 24 0 1 8
gr.spinellis.ckjm.CkjmOutputHandler 1 1 0 1 1 0 3 1
gr.spinellis.ckjm.ClassMetrics 24 1 0 33 196 6 23
gr.spinellis.ckjm.MetricsFilter 7 1 0 6 30 11 2 5
gr.spinellis.ckjm.ClassVisitor 13 1 0 14 71 34 2 9
gr.spinellis.ckjm.ClassMap 3 1 0 1 21 0 0 2
gr.spinellis.ckjm.PrintPlainResults 2 1 0 2 8 0 1 2
[eokike@visitor2 build]$

**Figure 4.7 (a).** Splitting an outlier class

gr.spinellis.ckjm.ClassMetrics 14 1 0 0 16 63 1 13
JavaParser$ButtonHandler 3 1 0 2 5 3 1 1
Classidentifier 3 1 0 0 7 0 1 3
JavaParser 5 1 0 3 59 2 1 3
gr.spinellis.ckjm.Classinfo 1 2 1 0 1 27 34 0 12
JavaParser$1 0 1 0 0 0 0 2 0

**Figure 4.7 (b).** Result of splitting an outlier class

Sources: [17]

In this paper, there is no evidence to suggest that a class whose LCOM=1 was improperly designed and therefore needs splitting. It is suggested that the basis for splitting a class should be whenever the number of methods (number of public methods NPM) is greater than or equal to 5. This agrees with the observation section 3.2.1.

**IV.4 Comparing LCOM, BFNLCOM, BNLCOM with TCC, LCC, Co, DCD, DCI**

Table 4.6 (a) below gives the descriptive statistics for LCOM, BFNLCOM and BNLCOM.

<table>
<thead>
<tr>
<th>Source: [17]</th>
</tr>
</thead>
</table>

From tables 4.6 (a) above the BFNLCOM and BNLCOM values are normalized values within the range [0,1]. These compare very well with the normalized values of Co, Coh, TCC, LCC, DC\textsubscript{D} and DC\textsubscript{I} described in [11] as shown in table 4.6 (b) below. This field experiments involved six systems. In the six systems, Chidamber and Kemerer’s LCOM metric are the LCOM1 and LCOM2. These measures are not normalized values, while the variant measures of Co, Coh, TCC, LCC, DC\textsubscript{D} and DC\textsubscript{I} are normalized. The first two systems in the field experiment (system 1 and system 2) and their measured cohesion descriptive statistics are shown in table 4.6(b) below. The descriptive statistics for Co, Coh, TCC, LCC, DC\textsubscript{D} and DC\textsubscript{I} compare very well with the descriptive statistics values for BFNLCOM and BNLCOM in table 4.6 (a). Comparing descriptive statistics values for BFNLCOM and BNLCOM in table 4.6 (a) with the
descriptive statistics values for CO, Coh, DCD, DCI in table 4.6 (b), the values are within the range [0,1] unlike the LCOM1, LCOM2 values (Chidamber and Kemerer’s LCOM metric) whose descriptive statistics values are [0, maximum] where maximum could be any number.

**Table 4.6 (b) Badri’s result**

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>LCOM1</th>
<th>LCOM2</th>
<th>Coh</th>
<th>Coo</th>
<th>TCC</th>
<th>LCC</th>
<th>DCD</th>
<th>DCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys 1 Min</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sys 1 Max</td>
<td>54 86</td>
<td>54 07</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.52</td>
</tr>
<tr>
<td>Sys 1 Mea</td>
<td>13 9.7</td>
<td>11 9.1</td>
<td>0.1</td>
<td>1</td>
<td>0.3</td>
<td>0.4</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>Sys 1 Med</td>
<td>15.00</td>
<td>9.00</td>
<td>0</td>
<td>0.2</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.33</td>
</tr>
<tr>
<td>Sys 1 Std</td>
<td>43 1.1</td>
<td>40 6.0</td>
<td>0</td>
<td>0.2</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
<td>0.43</td>
</tr>
<tr>
<td>Sys 2 Min</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sys 2 Max</td>
<td>13 99</td>
<td>13 99</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.42</td>
</tr>
<tr>
<td>Sys 2 Mea</td>
<td>24 4.4</td>
<td>12 9.0</td>
<td>0.1</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.34</td>
</tr>
<tr>
<td>Sys 2 Med</td>
<td>10 00</td>
<td>6.00</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>0.36</td>
</tr>
<tr>
<td>Sys 2 Std</td>
<td>13 07.1</td>
<td>86 1.6</td>
<td>0.1</td>
<td>0</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Source: [11]
BFNLCOM and BNLCOM values are standardized (Normalized) and are better than the LCOM values. With the NLCOM and BNLCOM applied to LCOM, LCOM measures satisfy the required properties for cohesion measures as suggested in [13,18,19].

V. CONCLUSION

This paper examined the LCOM metric defined by Chidamber and Kemerer and identified the major problems of the metric as the presence of outliers and unnormalized values. As a solution to these problems, three normalization approaches were proposed and evaluated using empirical data. Consequently, two of the normalization approaches namely the Best-Fit Normalized LCOM (BFNLCOM) and the Bowles Normalized LCOM (BNLCOM) were found suitable for the LCOM metric. However, from empirical evaluation, the BFNLCOM is the best fit normalization approach for the LCOM metric.

ACKNOWLEDGEMENTS

The work in this paper is part of the result from a thesis in the Department of Computer Science, submitted to the Faculty of Science in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY of the UNIVERSITY OF IBADAN, 2007. The thesis was supervised by Dr A. O. Osofisan.

REFERENCES

A COMPARISON OF FIXED CELLULAR AND GSM TELECOMMUNICATION NETWORKS

Oluyemisi Ajibose
West African Bioethics Training Programme,
Center for Bioethics,
P.O.Box 29822,
Secretariat Post Office,
Ibadan, Oyo State,
Nigeria
Email: yemisiaji@yahoo.com

&

Dele Oluwade*
P.O. Box 20253, U.I.P.O.,
Ibadan, OYO 200005,
Nigeria
Email: deleoluwade@yahoo.com

(*Present institutional address: Department of Library and Information Technology, Federal University of Technology, Minna, Niger State, Nigeria)

Received 21 October 2006

Thesis Communication

ABSTRACT
The economic development of any nation is affected by the type of telecommunication networks available. Until recently, the market share of telecommunications infrastructure in Africa is dominated by fixed cellular network. With the popularity of GSM network, it becomes necessary to examine features of its architecture. This paper presents a comparison of the fixed cellular network and GSM network available in Nigeria based on certain factors such as deployment difficulty, complexity of telephone sets, grade of service for intra-village call traffic, flexibility, co-channel interference and complexity of RSUs. It is observed that deployment of wireless telecommunication services is easier and more flexible than the wired services. However, the wired network offers a better grade of service. Hardware infrastructures in GSM networks which are equivalent to those in wireless/wireline networks are also highlighted.

KEYWORDS AND PHRASES: Fixed wired network, Fixed wireless network, GSM network.
I. INTRODUCTION

The function of the telecommunication sector of any nation is to provide an efficient and affordable information transmission service from one point to another. Transmission of telecommunication services can be through wired (copper, fiber) or wireless means.

The fixed cellular network consists of wired and wireless architectures. GSM architecture is a wireless architecture that enables mobility of users. The fixed wireless is simply an alternative to wired communications, it does not need mobility, instead users are provided with cost effective telecommunications from fixed locations. Until recently, the market share of telecommunications infrastructure in developing countries and Africa is dominated by fixed cellular architecture. Among the countries of Africa, Nigeria is currently the fastest growing telecommunication environment, and the eighth fastest growing environment in the world [5]. Some of the past and present network operators in Nigeria are: NITEL (which provide fixed wired services); MTN, ZAIN(formerly known as Vmobile/Econet), Globacom, M-tel and Etisalat provide GSM services while Multilinks, Starcom and MTS provide fixed wireless services. In particular, MTN—which is one of the first telecommunication companies to be licensed in 2001—is regarded to be the largest telecommunication company in Nigeria. There exists a total of 57 million subscribers connected to all the five GSM networks in Nigeria.

There are many factors that impede the practical deployment of telecommunication technology in developing countries. Some of the major deployment difficulties have been identified as [3]: high initial capital costs, incompatibility of equipment, obsolescence of affordable equipment, different service quality requirement, and inadequate spectrum availability. Considering their ease of deployment and cost effectiveness for thin call traffic situations, wireless technologies appear to be an alternative choice for small communities in short time [1]. However, wired technology is well proven and cost effective for the higher subscriber densities.

Generally, the poisson, exponential and erlang distributions are frequently used to analyze telecommunication architectures. The poisson is widely used to measure the call arrival, the erlang to determine the traffic in a network while the exponential determines the call holding time [3].

In [3], a comparison of fixed wired and fixed wireless architectures was presented. In the present paper, this comparison is extended to GSM architecture based on the same factors described in [3]. In addition, hardware infrastructures in GSM network which are equivalent to those in wireless/wireline cellular networks are highlighted.

II. FIXED CELLULAR NETWORK ARCHITECTURE

Fixed Cellular Network refers to a system or devices that are situated in a fixed location such as an office or home [6]. Fixed networks can be categorized into two viz: fixed wired architecture and fixed wireless architecture. In fixed wired architecture, cables are used as physical connection between the terminals and the switch center. In fixed wireless there is no physical connection between the devices. Fixed wireless devices normally derive their electrical power from utility mains in which point-to-point signal transmissions occur through the air over a terrestrial microwave platform rather than through copper or fiber cables. The advantages of fixed wireless include the ability to connect with users in remote areas without the need for laying
cables and the capacity for broad bandwidth that is not impeded by fiber or cable capacity.

Fixed cellular network architecture consists of several neighbouring areas that are grouped to form a macrocell which has single base station (BS) [3]. The base stations are linked through high-speed wireline or wireless trunks or through links to a satellite. The subscribers within an area are connected to the remote switching unit (RSU) which manages all the intra-village call traffic. The RSUS are linked to the base station through wireless links. The macrocells are often widely separated from each other and the coverage of the BSs and RSUs do not overlap. A high degree of frequency reuse is possible leading to high spectrum utilization efficiency.

In the wired architecture, the subscribers are connected through traditional wired links. The subscriber equipment is just the conventional telephone set which is powered from a central battery in the RSU. An advantage of this architecture is that intra-village call traffic does not use up any radio spectrum. Also, intra-village calls are not blocked. However, inter-village and long distance call still requires a wireless channel between the RSU and the BS.

For the wireless architecture, there exist wireless links between BS and RSUs and between RSUs and subscribers. This architecture has the advantage of ease of deployment but has the limitations of higher complexity and cost of subscriber equipment, higher call blocking and lower grade of service for inter-village calls when intra-village call traffic is high. As the telephone sets in the architecture are to be powered locally their maintenance cost are higher, and if powered from a.c. power source, the telephone service is affected during power failures [3].

III. GSM NETWORK ARCHITECTURE

GSM is the first digital mobile communication system to enable international roaming and ISDN service characteristics [4]. GSM is an open standard for services, infrastructure and communication independent of the individual countries, network operators and producers, and flexible to the requirements of the individual user.

The GSM architecture consists of the Mobile Station (MS), Base Station Subsystem (BSS) and the Network Subsystem (NSS). The mobile station (MS) is carried by the subscriber, specifically; the MSs are connected to the BSS and the BSS to the NSS through wireless links. This implies that communication from one unit to another unit in GSM is wireless. The telephone set used in GSM is called mobile phone. The mobile station is made up of the mobile equipment (ME) which is carried by the mobile station; it is anonymous and consists of the physical equipment such as the radio transceivers, display and digital signal processors. It also contains the subscriber identity module (SIM) which stores the subscriber’s information. The MS contains a battery that can be recharged.

The Base station subsystem is composed of two parts viz: the Base Transceiver Station (BTS) and the Base Station Controller (BSC); these communicate across the standard Abis interface. The Base transceiver station houses the radio transceivers that define a cell and handle the radio link protocols with the mobile station. The Base Station Controller manages the radio resources for one or more BTSs. It handles radio-channel setup, frequency hopping, and handovers. The BSC is the connection between the mobile station and the NSS. The NSS has the Mobile Switching Center as the central component. The A
interface is between the BSS and the MSC. The A interface manages allocation of suitable radio resources to the MSs and mobility management. The BTSs, whose coverage overlap, are widely separated from each other. Transmission over Air interface is digital.

GSM has the advantage of subscriber mobility, open interface, unique services like Short Message Service (SMS), web browsing, email and Multimedia Messaging Service (MMS).

IV COMPARISON OF FIXED CELLULAR NETWORK AND GSM NETWORK ARCHITECTURES

Table 1 below shows a comparison of fixed cellular network and GSM network architectures in terms of certain factors.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Fixed Wired</th>
<th>Fixed Wireless</th>
<th>GSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Deploymen nt Difficulty</td>
<td>High</td>
<td>Low</td>
<td>Lower</td>
</tr>
<tr>
<td>2 Complexity of telephone sets</td>
<td>Low</td>
<td>High</td>
<td>Higher</td>
</tr>
<tr>
<td>3 Grade of service for intra-village call</td>
<td>Very Good</td>
<td>Fair</td>
<td>Good</td>
</tr>
<tr>
<td>4 Overall grade of service</td>
<td>Good</td>
<td>Fair</td>
<td>Fair</td>
</tr>
<tr>
<td>5 Flexibility</td>
<td>Poor</td>
<td>Good</td>
<td>Better</td>
</tr>
<tr>
<td>6 Co channel interference</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>7 Complexity of RSUs</td>
<td>Medium</td>
<td>High</td>
<td>Higher</td>
</tr>
</tbody>
</table>

Table 1: Comparison of fixed cellular and GSM network architectures

The above can be interpreted as follows: there is lower ease of deployment difficulty in GSM. Also, the complexity of GSM telephone set is more enhanced than the fixed cellular telephone set. There is no intra-village call traffic in GSM but communication exists between the different components. The overall grade of service in GSM can be assumed to be fair compared with fixed wireless (almost the same grade of service). To guarantee flexibility, open interfaces are specified in GSM between particular network elements. This way, network operators can be supplied by different producers. Nevertheless, the interfaces’ functionality is very well specified, to guarantee a smooth data transmission. GSM architecture is designed to accommodate minimal co-channel interference. Minimizing co-channel interference is a goal in any cellular system, since it allows better service for a given cell size, or the use of smaller cells, thus increasing the overall capacity of the system. GSM does not use the term Remote Switching Unit (RSU), however it is a MOBILE Switching Centre (MSC) which acts like a normal switching node of the normal telephones of the landlines and in addition provides all the functionality needed to handle a mobile subscriber. Table 2 shows some hardware infrastructures in rural wireless/wireline architecture and the equivalent infrastructures in GSM architecture.
Table 2: Equivalent Hardware Infrastructures in Wireless/Wireline and GSM Architecture

<table>
<thead>
<tr>
<th>HARDWARE CONCEPTS IN RURAL WIRELESS/WIRELINE ARCHITECTURE</th>
<th>EQUIVALENT CONCEPTS IN GSM ARCHITECTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Village subscribers (VS)</td>
<td>* Mobile station (MS)</td>
</tr>
<tr>
<td>* Base station (BS)</td>
<td>* Base station controller (BSC)</td>
</tr>
<tr>
<td>* Remote switching unit (RSU)</td>
<td>* MS, Um interface &amp; BSS</td>
</tr>
<tr>
<td>* Macrocell</td>
<td>* Mobile service switching centre (MSC)</td>
</tr>
</tbody>
</table>

V CONCLUSION

In this paper a comparative study of two telecommunication network architectures namely fixed cellular (wireless and wired) architecture and GSM architecture has been carried out. The fixed wireless and GSM network architectures are easily deployed to any area. On the other hand, the fixed wired network architecture has a better grade of service compared to the other two.

POSTSCRIPT

This paper is part of the M.Sc (Computer Science) project work of the first author carried out under the supervision of the second author [2].

REFERENCES

[5] Ernest Ndukwe (2008), ‘From Telecommunications Back Waters To A Regional Hub: Tracking the Role of the Regulator in Nigeria’s Telecom Revolution’, delivered at the 9th Distinguished Electrical and Electronics Engineer Yearly Lecture, organized by the Nigerian Institution of Electrical and Electronics Engineers (NIEEE), a division of the Nigerian Society of Engineers
ADVERTIZE YOUR PRODUCTS AND SERVICES IN THE

AFRICAN JOURNAL OF COMPUTING AND ICT

(ISSN 2006 -1781)

FOR DETAILED INFORMATION ON SUBSCRIPTION AND ADVERTIZE-
MENT RATES, PLEASE CONTACT THE PRINCIPAL CONSULTANT
(dewadeconsult@yahoo.com)
ON SOME COMBINATORIAL RESULTS OF COLLAPSE AND PROPERTIES OF HEIGHT IN FULL TRANSFORMATION SEMIGROUPS

A.O. Adeniji
Dept. of Mathematics, Statistics and Computer Science,
University of Abuja,
Gwagwalada, Abuja,
Nigeria
E-mail address: adeniji4love@yahoo.com

&

S.O. Makanjuola
Dept. of Mathematics,
University of Ilorin,
Ilorin
Nigeria
E-mail address: sanmimaks@yahoo.com

Received 24 May, 2008

Communicated by Dele Oluwade

ABSTRACT

In a Transformation Semigroup S, an element \( \alpha \) in S is collapsible, \( c(\alpha) \) if there exists a number \( q = |t\alpha^{-1}| \geq 2 \) where \( t \) is an element in the image of \( \alpha \). In this note we obtain formula for the total number of collapsible elements for \( |t\alpha^{-1}| = 2 \) and \( |t\alpha^{-1}| = 3 \) for \( n \geq 2 \) in \( T_n \), the full transformation semigroup. Also \( q \) is defined for some height \( h(\alpha) = |Im\alpha| \).

Keywords: Collapse, Height, Full Transformation Semigroup, Idempotent, Semigroup
I INTRODUCTION
Among ‘naturally occurring’ semigroups, the full transformation semigroup \( T = T_X \), consisting of all self-maps of a non-empty set \( X = X_n \), has been perhaps the most intensively studied\[7\]. For standard terms in semigroup theory see, for example,\[4\].

A groupoid \((S, \mu)\) is defined as a non-empty set \( S \) on which a binary operation \( \mu \) by which we mean a map \( \mu : S \times S \rightarrow S \) is defined. We say that \((S, \mu)\) is a Semigroup if the operation \( \mu \) is associative, that is to say, if, for all \( x, y \) and \( z \) in \( S \),

\[
((x, y) \mu, z) \mu = (x, (y, z) \mu) \mu.
\]

Let \( X_n = \{1, 2, 3, \ldots, n\} \). Then a transformation \( \alpha : \text{Dom} \alpha \subseteq X_n \rightarrow \text{Im} \alpha \) is said to be full or total if \( \text{Dom} \alpha = X_n \). The full transformation semigroup is also known as the symmetric semigroup. It is also a well known fact that, if \( S = T_n \), then \(|S| = n^n [1]\). Enumerative problems of an essentially combinatorial in nature arise naturally in the study of semigroups of transformations \[2\]. An element \( \alpha \in T_n \) is an idempotent, if \( \alpha^2 = \alpha \). If we denote by \( S_n \) the symmetric group, which is one – one in nature, that is \( S_n \subseteq T_n \), then Howie \[3\] showed that \( T_n \setminus S_n = S_n \) is idempotent generated and Tainiter \[6\] showed that the total number of its idempotents is

\[
\sum_{r=0}^{n-1} \binom{n}{r} r^{n-r}
\]

Another study on the full transformation semigroup which is of interest in this paper is knowing the total number of collapsible elements, \(|c(\alpha)|\). Howie \[3\] defined a collapsible element as:

\[
c(\alpha) = \bigcup_{r \in \text{Im} \alpha} t\alpha^{-1} : |t\alpha^{-1}| \geq 2
\]

In his paper, Howie \[5\] still defined collapsible element as

\[
c(\alpha) = \sum_{r \in \text{Im} \alpha} t\alpha^{-1} \bigg| t\alpha^{-1} \bigg| - 1
\]

The former is of interest in this paper.

We shall denote the number of collapsible elements by

\[
\eta = |c(\alpha)| = \bigg| \bigcup_{r \in \text{Im} \alpha} t\alpha^{-1} : |t\alpha^{-1}| \geq 2 \bigg|
\]

For the non-collapsible elements of \( T_n \), which is the symmetric group \( S_n \), we have the expression, \( n! \) for \( c(\alpha) = 0 \). That is, a situation when no image appears more than once. The number of collapsible elements in \( T_n \) does not seem to have been investigated before and we set out to investigate these numbers. In this paper we obtain formula for the total number of collapsible elements for \( |t\alpha^{-1}| = 2 \) and \( |t\alpha^{-1}| = 3 \) for \( n \geq 2 \) in \( T_n \), the full transformation semigroup. Also \( q \) is defined for some height \( h(\alpha) = |\text{Im} \alpha| \).

II COMBINATORIAL RESULTS
Let \( S \) be the semigroup \( T_n \). If \( \alpha \in S \), \( h(\alpha) = |\text{Im} \alpha| \), the length of the image of \( \alpha \). Then the principal result of this paper is:

**Theorem 1**:  
Let \( X_n = \{1, 2, 3 \ldots, n\}, \alpha \in S \) and  
\[
\eta = \bigg| \bigcup_{r \in \text{Im} \alpha} t\alpha^{-1} : |t\alpha^{-1}| \geq 2 \bigg|
\]

Then with the initial value of \( n \) as the value of \( |t\alpha - 1| \),

\[
\eta = n! \binom{n}{q} \quad \text{for } q = |t\alpha - 1| = 2
\]

\[
\eta = \frac{n}{2} \binom{n}{q} \quad \text{for } q = 3.
\]
As a step towards proving the theorem above, we have the following:

**Lemma 1**: An element $\alpha$ with $h(\alpha) = n$ is not collapsible.

**Proof**: Let $\alpha \in S$. Let $Im\alpha = t_\alpha$ for each $t \in X_n$ and domain of $\alpha$ is $n$ i.e., $dom\alpha = |t_\alpha-1| = n$. Assume that $|Im\alpha| \neq 1$, it means that $|t_\alpha| \geq 2$ and it implies that the domain $|t_\alpha-1| \geq 2$.

Taking $|t_\alpha| = 1$ does not imply $|t_\alpha-1| = 1$. In fact $|t_\alpha-1|$ cannot be $1$ inorder to satisfy collapsible property. Thus $|t_\alpha-1| \geq 2$ and $|t_\alpha| \geq 1$.

Now let $|t_\alpha-1| = n$ then we have what is called a symmetric group, $S_n$ which is one-one and not collapsible. Hence the result $\eta = n!$ for $q = 0^\ast$.

**Proof of Theorem 1**

Let $t_k \in Im\alpha$, $S = T_n$ and $dom\alpha = |t_\alpha-1| = n$. Let $\alpha_y \in S$ be such that $\alpha_y = \begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ t_1 & t_2 & t_3 & \ldots & t_k \end{pmatrix}$

For $\alpha_y$ to be collapsible, $t_k \leq n$ and $|t_k\alpha| < n$ where $y$ represents each element in the order of $T_n$ which is $n$ in number. That is, we have $\alpha_y = \alpha_1, \alpha_2, \alpha_3, \ldots \alpha_n$. If $dom\alpha = n$ and $|t_k\alpha| = 2$ for a particular $\alpha_y \in S$,

$\Rightarrow h(\alpha) = |Im\alpha| = 2$ and in that case $|t_k\alpha-1| > 2$. From lemma 1, $h(\alpha) \neq n$ for a collapsible element with domain $n$. Thus for $n = 2$, $|t_\alpha-1| = 2 \Rightarrow h(\alpha) = 1$. If $|t_k\alpha-1| > 2$ then $|t_k\alpha| \geq 1$. Therefore for $|t_k\alpha-1| = 2 = q$,

, we have $\eta = n! \binom{n}{q}$ and for $|t_k\alpha-1| = 3$, $\eta = \frac{n!}{2} \binom{n}{q}$

with the condition that $n$ starts from the value of $q^\ast$.

### III CONCLUSION

It can be deduced from theorem 1 and lemma 1 that if $q = n$, then $h(\alpha) = 1$; $q = 2$, then $h(\alpha) = n-1$ and when $q = 3$ then $h(\alpha) = n-2 \ \forall n$. This is to establish the relationship between collapse and height in this paper. Also, values of $\eta$ is known for each of $q = 2$ and $q = 3$.

### REFERENCES

ADVERTIZE YOUR PRODUCTS AND SERVICES IN THE

AFRICAN JOURNAL OF COMPUTING AND ICT

(ISSN 2006 -1781)

FOR DETAILED INFORMATION ON SUBSCRIPTION AND ADVERTIZEMENT RATES, PLEASE CONTACT THE PRINCIPAL CONSULTANT

(dewadeconsult@yahoo.com)
Biodata of Authors in This Issue

• **Adedeji Badiru** is a Professor and Department Head of the Department of Systems & Engineering Management at the Air Force Institute of Technology, Dayton, Ohio. He was previously professor and department head of Industrial & Information Engineering at the University of Tennessee in Knoxville. Prior to that, he was professor of industrial engineering and Dean of University College at the University of Oklahoma. He is a registered professional engineer, a certified Project Management Professional (PMP), a Fellow of the Institute of Industrial Engineers, and a Fellow of the Nigerian Academy of Engineering. He holds BS in Industrial Engineering, MS in Mathematics, and MS in Industrial Engineering from Tennessee Technological University, and Ph.D. in Industrial Engineering from the University of Central Florida. His areas of interest include mathematical modeling, project modeling and analysis, economic analysis, and productivity analysis and improvement. He is the author of several books and technical journal articles. He is the editor of the Handbook of Industrial & Systems Engineering. He is a member of several professional associations including Institute of Industrial Engineers (IIE), Society of Manufacturing Engineers (SME), Institute for Operations Research and Management Science (INFORMS), American Society for Engineering Education (ASEE), New York Academy of Science, and Project Management Institute (PMI). He has served as a consultant to several organizations around the world including Russia, Mexico, Taiwan, Nigeria, and Ghana. He is on the editorial and review boards of several technical journals and book publishers.

• **0. Ibidapo-Obe** was awarded a Bachelor of Science [B.Sc. (Hons)] degree in Mathematics in the 1st Class Division by the University of Lagos, Nigeria in 1971; a Master of Mathematics [M. Maths] degree in Applied Mathematics with a minor in Computer Science in 1973 and a Doctor of Philosophy [Ph.D] in Civil Engineering with specialisation in Applied Mechanics/Systems in 1976 both from the University of Waterloo, Ontario, Canada. Professor Ibidapo-Obe was a Western State Government Scholar at the University of Lagos from 1968-1971 and was the overall best graduating student of the University in 1971. Professor Ibidapo-Obe was a Commonwealth Scholar (Canada) (1972-1976); a recipient of the University of Waterloo Post Doctoral Distinction Award Fellow (1976) and a NSERC/CIDA (Natural Sciences and Engineering Research Council/Canadian International Development Agency) Research Fellow (1977-1979) both at the University of Waterloo. He was subsequently a Senior Fulbright Research Scholar (1980-1981) at the State University of New York at Buffalo, N.Y. U.S.A. where he served as a Visiting Research Associate Professor in the Department of Civil Engineering (Under Professor T.T. Soong). He has had several international teaching/research exposures as a Visiting Professor at the University of Tennessee in Knoxville, USA (2001) and the University of Oklahoma, Norman, USA (1998); Academic Guest at the Otto-von-Guericke Universitat in Magdeburg, Germany (1995,1998); Visiting Professor at the University of Toronto, Canada (1990); Visiting Associate Professor at the Louisiana State Southern University, Baton Rouge, USA (1989) and Visiting Professor at the University of Zimbabwe (1989) and University of Yaounde I (1995). Professor Ibidapo-Obe’s major academic contributions are in the areas of Control and
Information Systems including specialist interests in Stochastic/Optimization Problems in Engineering, Reliability Studies; Simulation and Animation Studies (with application to Urban Transportation, Water Resources, Biomedics, Expert Systems, etc.). He pioneered studies on Stochastic Methods in Mechanics and the Development of Computer Algorithms applying Martingale Concepts to the Control of Nonlinear Dynamical Systems (1975). He was also part of the pioneering efforts on Robotics developments and applications especially in the area of sensors and actuators placement (1981 ----). Professor Ibidapo-Obe has served as Head of the Engineering Analysis Unit (1991-1995) and the Dean of the Faculty of Engineering at the University of Lagos, (1995- 1999). He was subsequently the Deputy Vice-Chancellor in April 2000 and acted as Vice-Chancellor between September 2000 and April 2002. He was appointed substantive Vice-Chancellor on 1st May, 2002. He was the pioneering Managing Director of UNILAG CONSULT - the University of Lagos Consultancy Services (1984-1989). Professor Ibidapo-Obe has served on the Editorial Board and Reviewer of several journals and publications. In 2004; Professor Ibidapo-Obe was conferred with the prestigious Fellowship of the Academy of Science, and Academy of Engineering, Nigerian Computer Society and Mathematical Association of Nigeria as well as the Officer of the Federal Republic (OFR) by the Federal Republic of Nigeria.

- Oluwakemi Christiana Abikoye obtained a Bachelor of Science degree in Computer Science from the University of Ilorin and a Master of Science degree in Computer Science from the University of Ibadan. Currently, she is pursuing a Doctor of Philosophy (PhD) degree. She is lecturing at the Department of Computer Science, University of Ilorin. Her research interests include computer/communication network security, cryptography, human computer interaction and biometrics.

- Dele Olopade-Oluwade studied both Building Technology and Mathematics at the University of Ife/Obafemi Awolowo University (O.A.U), Ile-Ife, Nigeria and received a Bachelor of Science (B.Sc) degree in Mathematics (1990). He also received a Postgraduate Diploma (PGD) in Computer Science from the University of Lagos, Nigeria (1998), a Master of Science (M.Sc) degree in Mathematics from the O.A.U.,Ile-Ife (1999), as well as a Master of Philosophy (M.Phil) degree (2002) and Doctor of Philosophy (PhD) degree (2004) in Computer Science from the University of Ibadan, Nigeria . Dr. Oluwade became a member (MIEEE) of the Institute of Electrical & Electronics Engineers (IEEE) Inc., USA and IEEE Computer Society in 2001, and a Senior Member of the association (SMIEEE) in 2007. Since 2006, he has been the Publicity Secretary of the IEEE Nigeria Section and Chair, IEEE Nigeria Computer Chapter. Apart from Nigeria, Dr. Oluwade has had teaching and/or research experience in the Kingdom of Saudi Arabia, United States of America (USA) and Republic of South Africa. In 1991, he was elected as an Associate Member of the Sigma Xi, The Scientific Research Society, Raleigh, North Carolina, U.S.A. He is now a full member of the society. He has published a sizeable number of research papers on Computer Science and Mathematics in national and international journals and conference proceedings. Currently, he is a Senior Lecturer & Head, Department of Library & Information Technology, Federal University of Technology, Minna, Nigeria, and Principal Consultant, Dewade Systems Consult. He is a chartered information technology professional registered with the Computer Professionals
Registration Council of Nigeria. He also belongs to other professional associations including the American Mathematical Society and the Nigeria Computer Society.

- **Ezekiel Uzor Okike** obtained his Bachelor of Science degree in computer science (1992), a Master of Information Science degree (1995) and a Doctor of Philosophy degree in Computer Science (1997), all from the University of Ibadan. Presently, he is on leave of absence from the Department of Computer Science, University of Ibadan to Kampala International University, Kampala, Uganda where he is a Senior Lecturer & Chair of Computer Science, School of Postgraduate Studies.

- **Adenike O. Osofisan** received a Bachelor of Science degree in Computer Science with Economics from the University of Ife (now Obafemi Awolowo University (O.A.U)), Ile-Ife, a Master of Science in Computer Science from Georgia Institute of Technology, USA, a Doctor of Philosophy degree in Computer Science from O.A.U.,Ile-Ife and a Master in Business Administration (MBA) from the University of Ibadan. She taught for several years at the Department of Computer Science, The Polytechnic Ibadan before joining the services of the University of Ibadan as a Senior Lecturer and Head of the Department of Computer Science. Presently, she is a Professor and Head, Department of Computer Science, Lagos State University, Ojoo, Nigeria.

- **Oluyemisi Ajibose** graduated with a Bachelor of Science degree in Computer Science from the University of Ado-Ekiti, a Master of Science degree in Computer Science from the University of Ibadan. Presently, she is working with the West African Bioethics Training Programme, Ibadan, Nigeria.

- **Adenike Olusola Adeniji** is lecturing at the Department of Mathematics, Statistics and Computer Science, University of Abuja, Gwagwalada, Abuja, Nigeria

- **Samuel Olorunsanya Makanjuola** is senior lecturer at the Department of Mathematics, University of Ilorin, Nigeria
ADVERTIZE YOUR PRODUCTS AND SERVICES IN THE

AFRICAN JOURNAL OF COMPUTING AND ICT

(ISSN 2006 -1781)

FOR DETAILED INFORMATION ON SUBSCRIPTION AND ADVERTIZE-
MENT RATES, PLEASE CONTACT THE PRINCIPAL CONSULTANT
(dewadeconsult@yahoo.com)
GENERAL INTEREST NEWS

I  IEEE COMPUTER SOCIETY CERTIFICATION PROGRAMME

The IEEE Computer Society organizes a Certified Software Development Professional (CSDP) programme for software developers and software engineers. The goal is to increase the global awareness of the programme and achieve dramatic growth in the number of CSDP certification holders worldwide. All Senior Members of the IEEE and all licensed engineers are eligible to take the certification examination. Others who have an advanced software engineering degree and 2 years of experience or alternatively, have 4 years of software development/engineering experience, are also eligible. The testing fee for IEEE members is $450 (consisting of $100 application fee and $350 examination fee) and $550 for non-members (consisting of $100 application fee and $450 examination fee); these fees are subject to tax charges. The application deadline for the 01 January 2009 – 11 July 2009 testing window is 27 June 2009. Detailed information about the programme can be obtained from:

http://www.computer.org/certification

Enquiries on the certification programme may be directed to:

csdp@computer.org

II  IEEE COMMUNICATIONS SOCIETY CERTIFICATION PROGRAMME

The IEEE Communications Society (COMSOC) has launched her Wireless Communication Engineering Technologies Certification (WCET) programme in September 2008. Candidates who intend to sit for the spring certification examination (16 March 2009 – 04 April 2009) are expected to complete the application latest 02 February 2009, by 23:59 p.m. UTC. Detailed information about the ComSoc WCET certification may be obtained from:

http://www.comsoc.org/cert

or

http://www.ieee-wcet.org/application
MODE OF SUBMISSION

Papers intended for consideration, and other editorial correspondence, should be sent to the Editor – in – Chief preferably as email attachment in MS WORD (or in exceptional case, in PDF with LATEX source file included). The email should preferably have in its SUBJECT line ‘PAPER FOR AFRICAN J. COMP. & ICT’. A prospective author may suggest to the Editor-in-Chief names of reviewers suitable for his/her paper. The editor is however not obliged to use any of the suggested reviewers. Every submission should be prepared in a form that approximates the general IEEE style guide. Each paper should be accompanied by an informative Abstract and a list of keywords. The title of the paper and the names, institutional (and email) addresses of the author should be clearly indicated. The body of the paper should begin with an Introduction and end with a Discussion and/or Conclusion section. References should follow the following format for journals, conference proceedings and books


A reference should be cited in the body in square brackets, e.g. [4], and all citations should be included in the list of references at the end of the paper. Accepted papers would be published in a double column format. Every author (or group of authors) of an accepted paper shall be required to pay a publication charge of US 80.00 (or the Nigerian currency equivalent, which is presently approximately NGN 9,500), shortly before the paper is published, to defray administrative and processing cost. A statement of commitment to pay this charge would be required from all prospective authors. This charge shall be paid to the designated bank account of the journal. Every author of a published paper is entitled to a reprint of his/her paper.

SCOPE OF ARTICLES

Submissions to the journal must be relevant to computer scientists, computer engineers, information technology professionals, engineering scientists and allied practitioners in the industry and the academia. Papers are welcome from pure / applied computing and ICT, and allied areas including, software engineering, computer communication/networks, human computer interaction, data mining, discrete/data structures, intelligent library buildings, differential equations of computing, computer algebra, information systems, bioinformatics, information management etc. Papers are welcome from Nigeria, Africa and the international community. The journal considers original research papers, case reports and review papers. An author must clearly justify the importance of his submission to the readership of the journal. A prospective author need not be a member of the (Computer Chapter) IEEE Nigeria Section before his/her paper is published. The journal currently publishes only a print edition, twice in a year, one in the first half of the year and the other in the second half. Online edition is expected to follow later. Papers should be easily readable and be written in lucid language comprehensible to readers who are not working within the immediate subject area.

INFORMATION TO SUBSCRIBERS AND ADVERTIZERS

The subscription fee for one single copy of the journal is US 8.00, excluding postage (or equivalent Nigerian currency, which is presently approximately NGN1,000). A complete volume of the journal comprising of two editions cost US 15.00 (NGN 1,800), excluding postage. Discount shall be offered to individuals, institutions and organizations making bulk purchases. All correspondence on subscription and advertisement should be directed to the Principal Consultant.

---

1 http://www.ieee.org/organizations/pubs/transactions/stylesheets.htm
AIM AND OBJECTIVE OF JOURNAL

*African Journal of Computing and ICT* was established by the Nigeria Computer Chapter of the Institute of Electrical and Electronics Engineers (IEEE) Inc, New York, USA to publish concise technical papers on Computer Science, Information Technology, Information Management and related IEEE designated fields, in a relatively short time. First time review of a submitted paper is expected to be completed within 3 - 6 months. The goal of the journal is to contribute to rapid industrial development of Nigeria in particular, and Africa in general, via the publication of topical and relevant papers from a theoretic/academic and practical/industrial points of view. Submitted papers should contain substantial ideas which have the potential of aiding an invention or a technological breakthrough for African use. All correspondence pertaining to the Computer Chapter of the IEEE Nigeria Section should be sent to the Chair of the Chapter, Dr. ‘Dele Oluwade (*bamideleoluwade@computer.org*) . General information on the aim and objective as well as membership of IEEE, IEEE Computer Society and the IEEE Nigeria Section can be found on the following websites respectively: [http://www.ieee.org](http://www.ieee.org); [http://computer.org](http://computer.org); [http://www.ieee.org/nigeria](http://www.ieee.org/nigeria). Correspondence to the IEEE Nigeria Section should be sent to the Chair of the Section, Engr. Tunde Salihu (*tunde_salihu@yahoo.com*) .

*African Journal of Computing and ICT is published for the IEEE Nigeria Computer Chapter by Dewade Systems Consult, Nigeria (Reg.No. BN 0000524) . Email:dewadeconsult@yahoo.com*