IMAGE: A MAP OF THE STARS OF THE ORION CONSTELLATION

Print ISSN: 2514-863X Online ISSN: 2514-8648

JournalPreview

London Journal of Research in Computer Science and Technology
Volume 17 | Issue 1 | Compilation 1.0

London Journals Press

WWW.JOURNALSPRESS.COM

JournalPreview

LONDON JOURANALS OF RESEARCH IN COMPUTER SCIENCE AND TECHNOLOGY

This document is a pre-published view of London Journal of Research in Management and Business Volume 18, Issue 1 and Compilation 1.0. For any minor changes and updations kindly follow your paper\'s live editing URL given in sent email or get in touch with our support team at support@journalspress . com or visit our website to use live chat support . This is a beta document thus order, content or existence of papers may alter in the published eJournal . You are requested to kindly acknowledge and approve your research paper in this JounalPreview within three days.

Journal Content
In this Issue

London Journals Press

Statistical Evaluation of the Performance of the Neural Network

Sargsyan Siranush, Hovakimyan Anna

Yerevan State University

ABSTRACT

In this paper the problem of evaluating the performance of the neural network, based on a study of the probabilistic behavior of the network is considered. Direct propagation network consisted of layer of input nodes, hidden layer and output layer is examined. To evaluate the network performance the mathematical expectation and dispersion of weight at the input of the output layer are considered. For such networks the estimates for some of the statistical characteristics of the neural network in the case of two recognized classes were obtained.

Keywords : neural network, the weight of the neuron, recognition of the stimulus, mathematical expectation, dispersion.

Classification : C.2.6

Language : English

London Journals Press

LJP Copyright ID: 953198
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 17 | Issue 1 | Compilation 1.0

Statistical Evaluation of the Performance of the Neural Network

Sargsyan Siranush[α] & Hovakimyan Anna[σ]

I. ABSTRACT

In this paper the problem of evaluating the performance of the neural network, based on a study of the probabilistic behavior of the network is considered. Direct propagation network consisted of layer of input nodes, hidden layer and output layer is examined. To evaluate the network performance the mathematical expectation and dispersion of weight at the input of the output layer are considered. For such networks the estimates for some of the statistical characteristics of the neural network in the case of two recognized classes were obtained.

Keywords: neural network, the weight of the neuron, recognition of the stimulus, mathematical expectation, dispersion.

Author α σ: Department of Programming and Information Technologies Yerevan State University, Yerevan, Armenia.

II. INTRODUCTION

Artificial neural networks are often used for a variety of applications. For the successful application of artificial neural networks it is necessary to choose the right network architecture, to pick up its parameters, thresholds elements, activation function, and others [1, 2, 4]. Currently neural computation of different level of implementation, from specialized hardware to the neural network software packages, are becoming more widely used.

They are successfully used to solve a number of tasks such as forecasting of economic and financial indicators, the prediction of complications in patients in the postoperative period, biometric identification based on various characteristics, image processing, and others.

All this is possible for the neural networks because of their ability to learn and to establish of associative connections in the input data. Typically, depending on the task, it is required to resort to different methods to transform input data, allowing a correct judgment about the laws and special features in the data that reflect their quality characteristics [4].

III. PROBLEM DESCRIPTION

Artificial neural network is a mathematical model of biological neuron. It consists of relating to each other neurons. A neural network is exposed to learning by providing it with the input information in the form of numerical sequences. The training set up internal connections between neurons through which the network is endowed ability to recognize unfamiliar images. There are different types of neural networks that differ in both the topology and the learning algorithms.

In the neural network technologies an important role particularly play the selection of the network architecture and the values of its parameters that affect its efficiency.

A particular interest the study of the probabilistic behavior of the neural networks is presented [3, 4]. The neural network of direct distribution, which consists of a layer of input nodes, hidden layer and output layer is investigated. Neurons have a one-way communication, do not contain links between elements within the layer and backward linkages between the layers. The neurons of the input layer are connected to the

London Journal of Research in Computer Science and Technology

hidden layer neurons with excitatory and inhibitory connections randomly. The outputs of all neurons in the hidden layer neurons are connected to the output layer. Neurons in each layer are referred to as the input, hidden and output elements, respectively [1, 2, 4]. Neurons have unidirectional links and do not contain links between elements within a layer and feedbacks between the layers.

Signal (causative agent, incentive, stimulus) supplied to the input layer of neural network corresponds to external stimuli and is modeled by the vector whose coordinates take the values 0 and 1, depending on whether or not the corresponding neuron of the input layer is excited. The output signal from each neuron of the input layer is set to 1 if the neuron is excited, and 0 otherwise. The output signal generated by the input layer, is transferred to the hidden layer. Each neuron of the hidden layer implements a threshold function η as it inputs [4]:

$$\eta_k(i) = \begin{cases} 1, & \text{if } F_k(i) \geq \theta \\ 0, & \text{if } F_k(i) < \theta \end{cases}$$

÷where σ_n and σ_m are numbers of excited and inhibited bonds from excitatory and m inhibitory connections respectively, θ is threshold of hidden element (integer number). Excitatory and inhibitory connections between the input and hidden layers are assumed to be randomly and uniformly distributed. The outputs of all elements of the hidden layer are passed as an input to the output layer that implements the function R:

$$R = \begin{cases} 1, & \text{if } \sum_{k=1}^{N_A} \eta_k V_k > \theta_R \\ 0, & \text{if } \sum_{k=1}^{N_A} \eta_k V_k < \theta_R \\ \text{not determined}, & \text{if } \sum_{k=1}^{N_A} \eta_k V_k = \theta_R \end{cases}$$

where η_k is the activity the k-th hidden element, v_k is the weight of the k-th hidden element, N_A is the number of elements in the hidden layer, and θ_R is the threshold of the output element.

The structure of the neural network allows to correct errors during incorrect responses by means of gradual complication of decision rules. This is done by changing the weight vectors.

Let us determine the weight of the hidden element after training.

The activity $\eta_k(i)$ of k-th hidden element when the stimulus ξ_i is submitted to the network is determined by the formula

$$\eta_k(i) = \begin{cases} 1, & \text{if } F_k(i) \geq \theta \\ 0, & \text{if } F_k(i) < \theta \end{cases}$$

where $F_k(i)$ is the value of signal of the k-th hidden element, and θ is the threshold of hidden element.

At the stage of neural network training, a training sequence of stimuli is presented to network. For each stimulus a hidden element is tested for activity, and all the active elements of the hidden layer are encouraged with values δ_i (i = 1, 2) for the i-th class of stimuli.

$$\delta_i = \begin{cases} 1, & \text{for the first class of stimuli} \\ -1, & \text{for the second class of stimuli} \end{cases}$$

If each class has, respectively, l_1 and l_2 representatives, then after training via the sequence of two classes activators of the length L = l_1+ l_2, in the k-th hidden item the weight V_k is accumulated:

$$V_k = \sum_{i=1}^{2} \delta_i \sum_{j=1}^{l_i} \eta_k(j) + V_0 \qquad (1)$$

where V_o- is a initial weight of k –th hidden element.

London Journal of Research in Computer Science and Technology

When at the input to a network is the stimulus ξ_t, so the input of the output layer is fed weight (k=1, ...N_A)

$$U_t = \sum_{k=1}^{N_A} V_k \eta_k(t) = \sum_{k=1}^{N_A} (\sum_{i=1}^{2} \delta_i \sum_{j=1}^{l_i} \eta_k(j) + V_0)\eta_k(t) \quad (2)$$

Let $V_o = 0, l_1 = l_2 = l$. Then

$$U_t = \sum_{k=1}^{N_A} \sum_{i=1}^{2} \delta_i \sum_{j=1}^{l} \eta_k(j,t) \quad (3)$$

where

$$\eta_k(j,t) = \begin{cases} 1, & \text{if the } k^{th} \text{ hidden element is active for } \xi_j \text{ and } \xi_t \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The belonging of pathogen ξ_t to one of two classes is determined by comparing of the weight U_t with a threshold output element R. When $U_t > \theta_R$ so we have first class. When $U < \theta_R$ - second class. When $Ut = \theta_R$ - refusal of recognition.

For the given network, a predetermined training sequence and a predetermined reference stimulus ξ_t, the weight U_t has a certain value. However, for the class of neural networks the U_t is a random variable. To determine the probability, that the network selected from a class, correctly classifies stimulus ξ_x, is required to calculate the probability characteristics of the random variable of weight U_x, supplied to the input of the output layer.

According to the modified Chebyshev inequality [6], for any random variable z with mathematical expectation Mz=μ with any dispersion Dz= σ^2 we have following relations:

$$P(z>0) >= 1 - 1/(\mu^2/\sigma^2)) = 1 - \sigma^2/\mu^2, \text{ when } \mu>0 \quad (5)$$

$$P(z<0) >= 1 - 1/(\mu^2/\sigma^2)) = 1 - \sigma^2/\mu^2, \text{ when } \mu<0 \quad (6)$$

where P – is the probability of the corresponding event, σ- average quadratic deviation. Equations (5,6) can be used to estimate the probability of a correct response of network on ξ_x with θ_R=0 in the case of two recognized classes.

From the (5, 6) it follows that the probability of correct recognition increases, if attitude σ^2/μ^2 tends to zero.

With appropriate probabilistic characteristics of the network, we can estimate the probability of a correct response of network on ξ_x. If the relation $\sigma^2(U_x)/\mu^2(U_x)$ can be made arbitrarily small, then for a selected network with θ_R=0 the probability that a stimulus ξ_x is classified correctly seeks to 1. Let N_A is number of hidden elements of the network, $L=l_i+l_j$, $l_i(l_j)$-is the length of the training sequence of i-th(j- th) class, $\delta_i(\delta_j)$- is the increment of weight of hidden element when the stimulus from the i-th(j -th) class is presented. P_i is probability of excitation of hidden element when a stimulus from the i –th class is presented, P_{ij}- is probability of excitation of hidden element when incentives from the both i-th and j-th classes are presented. To find the expectation of the weight at the input of the output layer the following theorem is proved [3].`

Theorem. Let the a set of pathogens $\Omega=\{\xi\}$, $\Omega=\Omega_1 \cup \Omega_2$, $\Omega_1 \cap \Omega_2=\emptyset$, and training sequence $\xi_1, \xi_2, \ldots \xi_L$ are given. Then the expectation of the weight input in the output layer when the stimulus is from the i –th class is equal to

$$\mu_i = N_A (\delta_i P_i l_i + \delta_j P_{ij} l_j); \quad i,j = 1,2 \; i \neq j \quad (7)$$

Having μ_i, let's estimate dispersion of random variable U_x with appearance of the pathogen ξ_x at the entrance of the network.`

For j = 1,..L, and r= 1,..L we receive:

$$\sigma^2(U_x) = N_A L^2 \sum\sum \nu_j \nu_r \delta_j \delta_r (P_{jrx} - P_{jx} * P_{rx}) \quad (8)$$

where P_{jx}- is probability of excitation of hidden element when displaying incentives ξ_j, ξ_x.

P_{rx}- is excitation probability of hidden element when displaying incentives ξ_r, ξ_x,

P_{jrx}- excitation probability of hidden element when displaying incentives ξ_j, ξ_r, ξ_x.

Since the ratio $\sigma^2(U_x)/\mu^2(U_x)$ is not depend on the length of the training sequence, so any number of repetitions of the same training sequence does not change the characteristics of the system.

IV. RESULTS

Let's consider the relation $\sigma^2(U_x)/\mu^2(U_x)$ for the control stimulus ξ_x in cases, if $\xi_x \in \Omega_1$ and $\xi_x \in \Omega_2$.

Assessing the probability P_{jrx},P_{jx},P_{rx} the expressions for the dispersion will be recieved. Selected the following cases:

First case: assume that the control stimulus $\xi_x \in \Omega_1$. For the stimuli ξ_j and ξ_r in this case we obtain the following probabilities:

a) if $\xi_j \in \Omega_1$ and $\xi_r \in \Omega_1$, so $P_{jrx} = P_{jx} = P_{rx} = P_1$;
b) if $\xi_j \in \Omega_1$ and $\xi_r \in \Omega_2$, so $P_{jrx} = P_{12}$, $P_{jx} = P_1$, $P_{rx} = P_{12}$;
c) if $\xi_j \in \Omega_2$ and $\xi_r \in \Omega_1$, so $P_{jrx} = P_{12}$, $P_{jx} = P_{12}$, $P_{rx} = P_1$;
d) if $\xi_j \in \Omega_2$ and $\xi_r \in \Omega_2$, so $P_{jrx} = P_{12}$, $P_{jx} = P_{12}$, $P_{rx} = P_{12}$.

Let's calculate follow relationship $\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)}$ for these case (a,b,c,d):

$$\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)} = \frac{\sum_{j=1}^{L}\sum_{r=1}^{L} \nu_j\nu_r\delta_j\delta_r(P_1 - P_1^2)}{N_A \sum_{j=1}^{L}\sum_{r=1}^{L} \nu_j\nu_r\delta_j\delta_r P_1^2} = \frac{1-P_1}{N_A P_1} \quad (9)$$

$$\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)} = \frac{\sum_{j=1}^{L}\sum_{r=1}^{L} \nu_j\nu_r\delta_j\delta_r(P_{12} - P_1 * P_{12})}{N_A \sum_{j=1}^{L}\sum_{r=1}^{L} \nu_j\nu_r\delta_j\delta_r P_1 * P_{12}} = \frac{1-P_1}{N_A P_1} \quad (10)$$

$$\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)} = \frac{(P_{12} - P_{12} * P_1)}{N_A P_1 P_{12}} = \frac{1-P_1}{N_A P_1} \quad (11)$$

$$\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)} = \frac{(P_{12} - P_{12}^2)}{N_A P_{12}^2} = \frac{1-P_{12}}{N_A P_{12}} \quad (12)$$

The probability of detection for the first class is increased by the maximum value of P_1- P_{12}, called the characteristic function of the perceptron (CFP) [4]. Obviously, the function CFP seeks to a maximum value at $P_1 \to 1$.

In assessing the value of $\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)}$ again see that $P_1 \to 1$ when $\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)} \to 0$ (in most of the discussed cases). For the first class of stimuli the condition $\frac{\sigma_1^2(Ux)}{\mu_1^2(Ux)} \to 0$ is received at which execution CFP $\to$max, i.e. the probability of correct recognition for the first class stimulus increases.

Second case: assume that the control stimulus $\xi_x \in \Omega_2$.

For the stimulus ξ_j and ξ_r in discussed case the following probability will be obtained:

a). if $\xi_j \in \Omega_2$ and $\xi_r \in \Omega_2$, so $P_{jrx} = P_{jx} = P_{rx} = P_2$;
b). if $\xi_j \in \Omega_1$ and $\xi_r \in \Omega_2$, so $P_{jrx} = P_{12}$,$P_{jx} = P_{12}$,$P_{rx} = P_2$;
c). if $\xi_j \in \Omega_2$ and $\xi_r \in \Omega_1$, so $P_{jrx} = P_{12}$,$P_{jx} = P_2$,$P_{rx} = P_{12}$;
d). if $\xi_j \in \Omega_1$ and $\xi_r \in \Omega_1$, so $P_{jrx} = P_{12}$,$P_{jx} = P_{12}$,$P_{rx} = P_{12}$.

Similarly, calculating $\frac{\sigma_2^2(Ux)}{\mu_2^2(Ux)}$, for the second class will be received the condition for correct recognition:

$$\frac{\sigma_2^2(Ux)}{\mu_2^2(Ux)} \to 0$$

Therefore we received new conditions to increase of the right recognition for the stimulus of the i-th class:

$$\frac{\sigma_i^2(Ux)}{\mu_i^2(Ux)} \to 0 \qquad (i = 1; 2) \quad (13)$$

V. CONCLUSIONS

The resulting estimates for certain statistical characteristics of a neural network in the case of two recognized classes have shown efficiency in the training of neural networks. Moreover, a new condition for improving accuracy of recognition for the stimulus i-th class is received.

REFERENCES

1. Panteleev S.V. Development, research the use of neural network algorithms, M: 2001, 496 p. (in Russian).
2. Barsky A.B. Neural networks: the recognition, management, decision-making. M .: Finance and Statistics, 2004, 176 p. (in Russian).
3. Sargsyan S.G. Determination of the probability characteristics of adaptive recognition system, Trans. of Intern. Conf. Adaptable software, Kishinev, 1990.pp. 46-51. (in Russian).

4. Ivakhnenko A.G. Perceptron pattern recognition system, Naukova Dumka, Kiev, 1975.p. 426. (in Russian).
5. Hovakimyan A.,Sargsyan S.,Nazaryan A. Self-Organizing Map Application for Iris Recognition . Journal of Commun & Comput. Eng.ISSN 2090-623, www.m-sciences.com, Volume 3,Issue 2. 2013. PP.10-13.
6. Feller V., Introduction to probability theory and its applications, M., Mir, 1984, (in Russian).

This page is intentionally left blank

Survey on Current Trends and Techniques of Data Mining Research

Sadiq Hussain

Dibrugarh University

ABSTRACT

The paper surveys different aspects of data mining research. Data mining is helpful in acquiring knowledge from large domains of databases, datawarehouses and data marts. Different and current areas of data mining are also discussed. Issues and challenges of data mining along with different open source tools are addressed as well. Data mining is an important and evolving research area and used by the biologists to statisticians and computer scientists as well.

Keywords : data mining, knowledge discovery in databases, areas and tools in data mining, challenges of data mining.

Classification : H.2.6 H.2

Language : English

London Journals Press

LJP Copyright ID: 789758
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 17 | Issue 1 | Compilation 1.0

Survey on Current Trends and Techniques of Data Mining Research

Sadiq Hussain

I. ABSTRACT

The paper surveys different aspects of data mining research. Data mining is helpful in acquiring knowledge from large domains of databases, data warehouses and data marts. Different and current areas of data mining also discussed. Issues and challenges of data mining along with various open source tools are addressed as well. Data mining is an important and evolving research area and used by the biologists to statisticians and computer scientists as well.

Keywords: data mining, knowledge discovery in databases, areas and tools in data mining, challenges of data mining.

Author: System Administrator, Dibrugarh University, Dibrugarh, India.

II. INTRODUCTION

Data mining is extracting information and knowledge from huge amount of data. Data mining is an essential step in discovering knowledge from databases. There are numbers of databases, data marts, data warehouses all over the world. If the data are not analyzed to find out the interesting patterns, then the data would become data tombs. Data miners seek for the pearl in the sea of data. A data mining system may generate lots of patterns. Typically a small fraction of the patterns are interesting. Here the interesting means useable, valid and novel. Moreover, it is almost impossible to extract the interesting hidden patterns in the sea of data without the help of data mining tools. There are seven steps in data mining. They are data cleaning, data integration, data selection, data transformation, data mining, knowledge presentation and pattern evolution[7]. Database technology had evolved from primitive file processing to the development of data mining tools and applications. The data may be collected from various applications including science and engineering, management, business houses, government administration and environmental control. Interesting data patterns may be mined from spatial, time-related, text, biological, multimedia, web and legacy databases. Data mining facilitate management in decision making. The data mining job includes the discovery of concept descriptions, association, classification, prediction, clustering, trend analysis, deviation analysis and similarity analysis. Data mining in large databases poses various requirements and challenges for the researchers and developers. A multidimensional data model is used for the design of data warehouses and data marts. The core of such model is data cube [7]. Data cube consists of large set of facts and number of dimensions. Dimensions are the entities on which an organization keeps records. By nature, they are hierarchical.

III. DIFFERENT AREAS OF DATA MINING

3.1 Web Mining

As there is huge amount of data and information available in the World Wide Web, the data miners have a fertile area for web mining. Web mining is data mining techniques for extraction of information from web documents and services. The contents of the web are very dynamic. It is growing at a rapid pace, and the information is

London Journal of Research in Computer Science and Technology

continuously updated. Web mining may be divided into the following subtasks [2].

1. *Resource finding:* finding documents intended for the Web.
2. *Information selection and preprocessing:* Selection and preprocessing of the information retrieved from the Web.
3. *Generalization:* To discover the general patterns from the individual as well as multiple sites.
4. *Analysis:* Discovered patterns are interpreted for meaningful knowledge.

Web mining may be divided into Web Structure, Web Contents, and Web Access Patterns.

3.2 Text Mining

The term text mining or KDT (Knowledge Discovery in Text) was first proposed by Feldman and Dagan in 1996 [2]. The unstructured text may be mined using information retrieval, text categorization, or applying NLP techniques as a preprocessing step. Text Mining involves many applications such that text categorization, clustering, finding patterns and sequential patterns in texts, computational linguistics, and association discovery.

3.3 Spatial Data Mining

The spatial data mining deals with data related to location. The explosion of geographically related data for rapid development of IT, digital mapping, remote sensing, GIS demands for developing databases for spatial analysis and modeling. Spatial data description, classification, association, clustering, trend, and outlier analysis are the main components for spatial data mining.

3.4 Multimedia data mining

Multimedia data mining explores the interesting patterns from databases related to multimedia that manages a large collection of multimedia objects. Multimedia objects include audio, video, image, sequence data and hypertext data containing text, text markups, and linkages. Multimedia data research focuses on content-based retrieval, similarity search, association, classification and prediction analysis.

3.5 Time series data mining

A time series database changes its values and events with respect to time. Some of the examples of time series data are stock market data, business transaction data, dynamic production data, medical treatment data, web page access sequence and so on. The time series research involves issues related to similarity search, trend analysis, mining sequential and periodic patterns in time-related data.

3.6 Biological data mining

There is a large storage of clinical and biological data from DNA microarray data, genomic sequences, protein interactions as well as sequences, electronic health records, disease pathways, biomedical images and the list goes on. In the clinical context, biologists are trying to find the biological processes that are the cause of a disease. There are some issues related to these high-dimensional biological data. These matters include noisy and incomplete data, integrating various sources of data and processing computer intensive tasks. Biologists as well as clinical scientists used a variety of data mining tools to discover interesting and meaningful observations from a large number of heterogeneous data from different biological domains.

3.7 Educational data mining

Educational Data Mining (EDM) is an emerging research area concerned with the unique types of data that come from educational settings, and using those methods to better understand students. Educational Data Mining focuses on developing new tools and algorithms for discovering data patterns. EDM develops methods and applies techniques from statistics, machine learning, and data mining to analyze data collected during teaching and learning. New computer-supported interactive learning methods and tools have opened up opportunities to collect

and analyze student data, to discover patterns and trends in those data, and to make new discoveries and test hypotheses about how students learn. Data collected from online learning systems can be aggregated over large numbers of students and can contain many variables that data mining algorithms can explore for model building. Different student models are used for prediction of future learning behavior of the students. Computational models are used based on the student domain and pedagogy.

3.8 Ubiquitous data mining (UDM)

The data miners have a new challenge in the form of the ubiquitous access by using wearable computers, palmtops, cell phones, laptops. To extract hidden information from these devices requires advanced analysis. In the world of UDM, communication, computation, security, etc. are some of the factors. The one of the objectives of the UDM is to extract interesting patterns while minimizing the additional cost of the computing due to the above-cited factors. To implement data mining tasks like classification, clustering, associations, etc. are difficult for ubiquitous devices. Small display areas, data management in mobile are some of the challenges in this regards. The key issues are the advanced algorithm for mobile and distributed computing, data management issues, data representation techniques, integration of these devices with database applications, UDM architecture, software agents, agent interaction and applications of UDM [5].

3.9 Constraint-based data mining

Constraint-based data mining is one of the developing areas where the data miners use the constraint for better data mining. One of the applications of constraint-based data mining is Online Analytical Mining Architecture (OALM) developed by [6] and is designed for multi-dimensional as well as constraint based mining based on databases and data warehouses. Usually, data mining techniques lack user control. One form of data mining is where the human involvement is there in the form of constraints. There are various types of constraints with their own characteristics and purpose. They are knowledge type, data, dimension/level, interestingness, rule constraints.

IV. DATA MINING TOOLS

The following are the popular data mining open source tools.

4.1 RapidMiner

This tool is written in Java programming language, and it offers analytics of advanced level through its template-based framework. Users hardly have to do any coding. RapidMiner is capable of handling various tasks like statistical modeling, predictive analytics and visualization apart from data mining tasks. Rapid- Miner provides learning schemes, models and algorithms from WEKA and R scripts that make it more powerful. This open source is distributed under the AGPL open source license and it can be downloaded from SourceForge. It is one of the best business analytics software. All the data mining tasks are bundled in one single suite [http://rapid-i.com/content/view/181/ 190/].

4.2 WEKA

Weka was originally developed in a non-Java version for analyzing agricultural data. Later, the Java version was developed, and it became a powerful tool for different data mining applications like predictive modeling and data analysis. This software is free under the GNU General Public License, which is a big advantage compared to RapidMiner. As it is free under the GNU General Public License which is a big advantage of it as compared to its counterparts like RapidMiner. It can be customized by the users. Most of the data mining jobs are supported by Weka. They are classification, clustering, regression, feature extraction, visualization, etc. Its graphical user interface makes it a better-sophisticated tool for data mining process. So, Weka has become one of the most powerful open source data mining software. [http://en.

wikipedia.org/wiki/Weka_(machine_learning)] [http://www.cs.waikato.ac.nz/ml/ weka/].

4.3 R-Programming

Project R, which is a GNU project, is written in C, FORTRAN and R Language. R language is used for writing lots of modules of the software itself. R programming software is free, and it is also used for statistical computing and graphics. Data miners used R for developing statistical packages and analyzing the data. In recent years the popularity of R had increased because of its ease of use and extensibility. R provides different statistical techniques that include linear and nonlinear modeling; data mining processes i.e. classification, clustering, time series analysis and others. [http://www.r-project.org/][14].

4.4 Orange

Orange, a Python-based, powerful and open source tool for data mining users for the purpose of knowledge extraction. It has powerful visual programming and Python scripting attached to it. It can be used for machine learning as well as bioinformatics and text mining by adding add-ons. It's packed with features for data analytics. Orange has specialized add-ons like Bioorange for bio-informatics [http://orange.biolab.si/ features/].

4.5 KNIME

KNIME is capable of performing three main tasks in data preprocessing. They are extraction, transformation, and loading. The data processing is done by allowing the assembly of nodes. It is an integration platform with strong data analytics and reporting. KNIME used modular data pipelining concept for machine learning and data mining. It is used for business intelligence as well as financial data mining. KNIME is easily extendible and can be added a plug-in for specific jobs. This open source is also written in Java and based on Eclipse. The core version consists of various data integration modules. Its research area not only includes pharmaceutical research but also business data, financial intelligence and CRM customer data. [https://en.wikipedia.org/wiki/KNIME].

4.6 NLTK

When it comes to language processing tasks, NLTK is one of the major players. NLTK is used for machine learning, data mining, sentiment analysis and data scraping. It is also extensively used for language processing. Because it's written in Python, one can build applications on top of it, customizing it for small tasks. NLTK played a major role as a teaching tool, study tool, prototyping and can be used as a platform for high-quality research. [https://en.wikipedia.org/wiki/Natural_Language_Toolkit]

V. LITERATURE REVIEW

There are lots of data mining studies around the globe.

Students Mood recognition [3] was proposed by Christos N. Moridis et. al. for online self-assessment test. Exponential logic and formulas were used in this regards. The inputs were student's previous answers and slide bar status. The exponential logic variables were a total number of questions for the online self-assessment test, student's goal, and slide bar value. Appropriate feedbacks are recorded based on current status of moods of the students. Student's manual selection of their mood using slide bar without any automation is the limitation of the system.

A novel weakly supervised cyber criminal network mining method [18] was proposed by Raymond Y.K. Lau et. al. The technique was based on relationships both explicit and implicit among the cyber criminals. The messages posted by these criminals on the social media were the basis of this method. The algorithm used in this context was context-sensitive Gibbs sampling algorithm. The algorithm mined both transactional and collaborative semantics to find the relationship among such criminals. The model used was a probabilistic generative model for extracting multi-word expressions. Two types of cyber

criminal relationships were established in unlabeled messages. The approach used here is concept level for the implicit semantics associated with the text.

Shenghua Bao et. al. [16] proposed for discovering and connecting with social emotions based on the online documents with emotions to help the users to select related documents by their emotional preferences. This is a problem of document categorization. For such social affective text mining, a joint emotion-topic model was proposed by introducing an additional layer for such kind of emotion modeling into Latent Dirichlet Allocation (LDA). Associate emotions with specific emotional context were used instead of a single term. The authors developed an approximate inference model by using Gibbs Sampling Algorithm. The model categorized text based on different emotions such as touch, surprise, and empathy, etc. by using social affective text as input.

Luigi Lancieri et. al. [10] proposed a classification method for Internet users based on their behavior at net to offer enhanced services. For this purpose, IP Address, timestamp, keywords from proxy cache, URL, categorized user behaviour were collected. Two different kinds of categorization algorithms were used. One is called "hard clustering" for partition and another is "soft clustering" for finding overlapping clusters to group users. Hierarchical agglomerative clustering (HAC) was used for hard clustering.

Li-Der Chou et. al. [9] proposed the use of social media with the help of mobile devices to create social network group for the children with developmental disabilities (CDD). Families with CDD, university, hospital and foundation came hand to hand to share significant information based on online social network related to childcare of such children. The users can access the application with the help of PDA, personal computer or mobile devices by installing the application on such devices.

In [17], the authors used distributional features of text categorization that took into account the compactness and the position of the first appearance of the word. Previous researchers had used 'bag of words' representation and assigned a word with values and concerned with whether the word appeared in the document or not or the frequency of the word. The authors in their research work explored other types of values which express distribution of word in a document. The distributional features are used by a tf idf style equation and features of different categories are combined using ensemble learning techniques. The authors proved experimentally that distributional features are useful for text categorisation. The categorisation performance improves significantly by using these features with little additional cost in contrast to traditional methods. The distribution features performances are enhanced the case of long documents and when the writing style is casual.

In [15], the authors designed web service recommendation systems. While designing web service recommendation systems, the focused research problem was to avoid recommending unfair or poor services to the users. The system should help users to choose right service from the huge number of available web services. The widely recommended metric in this regards is the reputation of web services. The feedback ratings by the users are used for providing service reputation score. Malicious and subjective user feedback often leads to bias that affects the reputation measurement of web services. In their research work, they proposed a novel system for the same. Cumulative Sum Control Chart and Pearson Correlation Coefficient were used to find malicious user feedback ratings. The system performed better by using Bloom filtering and proposed malicious feedback rating prevention scheme. Extensive experiments were conducted by using 1.5 million web service invocation records. The experimental results showed that success ratio of the web service recommendations may be enhanced and the system might reduce the deviation of reputation measurement.

In [11], the researchers proposed a novel intelligent system which would be able to detect the road accidents automatically, notify them by using vehicular networks and estimate the severity of the accident based on data mining tools and knowledge interference. Various variables such as the vehicle speed, the type of vehicles involved, the impact speed, and the status of the airbag, etc. are used for measuring the severity of the accident. A prototype based on off-the-shelf devices was developed and validated it at the Applus + IDIADA Automotive Research Corporation facilities, showing that this system can reduce the time needed to alert and deploy emergency services notably after an accident takes place. Three classification algorithms were used such as Decision Trees, Support Vector Machines, and Bayesian networks and were compared for best results. It was found that Bayesian model for classification is the best-suited model.

In [4], the authors proposed a novel system called Mobile Commerce Explorer (MCE). The system was for mining as well as prediction of mobile users' movements. It can also be used for purchasing transactions under the context of mobile commerce. The framework (MCE) contains three major components - 1)Similarity Inference Model (SIM) for measuring the similarities among stores and items, which are two basic mobile commerce entities considered in the paper; 2) Personal Mobile Commerce Pattern Mine (PMCP-Mine) algorithm for efficient discovery of mobile users' Personal Mobile Commerce Patterns (PMCPs) and 3) Mobile Commerce Behavior Predictor (MCBP) for prediction of possible mobile user behaviors. The study predicted mobile users' commerce behaviors to recommend stores and items previously not known to a user.

In [8], the researchers proposed a technique for the prediction of what else the customer likely to buy based on partial information about the contents of a shopping cart. The data structure used in this context was itemset trees (ITtrees), they obtained all the rules whose antecedents contain at least one item that is missing from the shopping cart in a computationally efficient manner. The classical Bayesian decision theory and a new algorithm based on Dempster-Shafer (DS) theory of evidence combination were combined for finding out rules based uncertainty processing technique. The proposed algorithm enhanced the performance. As the input, the algorithm takes an incoming item set and returns a graph based on association rules entailed by the incoming item set. The proposed algorithm used depth-first search technique and also updated the rule graph.

VI. DATA MINING TECHNIQUES

Several data mining techniques are used in data mining tasks. Association, classification, clustering, prediction, sequential pattern mining, etc. are data mining techniques.

6.1 Classification

Classification finds rules that partition data into some groups. The input for the classification is the training set. The training set's class labels are already known. Classification assigns class labels to unlabelled records based on a model that acquires knowledge from the training datasets. Such classification is known as supervised learning as the class labels are known. There are several classification models. Some of the common classification models are decision trees, neural networks, genetic algorithms, support vector machines, Bayesian classifiers. The application includes credit risk analysis, fraud detection, banking and medical application, etc. [2].

6.2 Clustering

Clustering is a method of grouping data so that data within the cluster have high similarity and dissimilar to data in other groups. Clustering algorithms may be used for organizing data, categorize data for model construction and data compression, outlier detection, etc. Many clustering algorithms were developed and are categorized as partitioning methods, hierarchical methods, density based and grid based methods.

The datasets may be numerical or categorical. K-Means, hierarchical, DBSCAN, OPTICS, STING are some of the well-known data clustering algorithms [13].

6.3 Association Rule Mining

Association rule mining is a well-researched method for discovering interesting relations between variables in large databases. In association rule, the expression is of the form X=>Y, where X and Y are set of items [2]. The main objective is to discover all the rules that have support and confidence greater than or equal to minimum support or confidence in a database. Support means that how often X and Y occurs together as a percentage of total transactions. Confidence means that how much a particular item is dependent on another. There is no significance for the patterns with low confidence and support. The users can extract useful and interesting information from the patterns with intermediate values of confidence and support. The association rule mining algorithms include Apriori, AprioriTid, Apriori hybrid and Tertius algorithms [13].

6.4 Neural Networks

Neural networks are new computing paradigm that is inspired by the biological nervous system, such as the brain, to process information [13]. It involves developing mathematical structures with ability to learn [2]. The Neural networks have the ability to extract meaningful and useful patterns and trends from the complex data. It is applicable to real world problems especially in case of industry. As the neural networks are good at identifying patterns or trends, they may be applicable for prediction or forecasting needs. The system is composed of highly interconnected processing elements (neurons) working together to solve a specific problem. Artificial neural network (ANN) learns by example [15]. ANN is configured for specific application as classification, pattern recognition etc. through a learning process. It may also be used for three-dimensional object recognition, hand-written word recognition, face recognition, etc. Neural networks have the drawback of not explaining the derived results. Another problem is that it suffers from long learning times. As the data grows, the situation becomes worse for that problem.

6.5 Support Vector Machines

Support vector machines (SVM) belong to a new class of machine learning algorithms and are based on statistical learning theory [2]. The main concept is to non-linearly map the data set into a high dimensional feature space and use a linear discriminator for classification of data. It is basically used for regression, classification and decision tree construction. SVMs select the plane which maximizes the margin separating the two classes. The margin is defined as the distance between the separating hyperplane to the nearest point of A, plus the distance from the hyperplane to the nearest point in B, where A and B are two linearly separable sets. SVM has been used in many applications including face detection, handwritten character and digits recognition, speech recognition, image and information retrieval [12].

6.6 Genetic Algorithms

Genetic algorithms are a new paradigm in computing inspired by Darwin's theory of evolution [2]. A population of the individual with possible solution to a problem is created initially at random. Then the crossover is done by combining pairs of individuals to produce offspring of next generation. A mutation process is used to modify the genetic structure of some members of new generation randomly. The algorithm searches for a solution in the successive generation. When an optimum solution is found or some fixed time is elapsed, the process comes to an end. Genetic algorithms are widely used in problems where optimization is required.

VII. ACKNOWLEDGEMENT

The author expressed his gratefulness to Prof. Alak Kr. Buragohain, Vice-Chancellor,

Dibrugarh University for his inspiring words. The author also acknowledged Prof. Gopal Chandra Hazarika, Professor, Department of Mathematics for his valuable suggestions.

REFERENCES

1. Adam Baba, Gouse Pasha, Shaik Althaf Ahammed, S. Nasira Tabassum, "Introduction to Neural Networks Design Architecture", International Journal of Scientific & Engineering Research Volume 4, Issue 2, Februry 2013, ISSN 2229-5518.
2. Arun K Pujari, Data Mining Techniques, University Press, 2013.
3. Christos N. Moridis and Anastasios A. Economides "Mood Recognition during Online Self-Assessment Tests" IEEE Transactions on Learning Technologies, Vol. 2, NO. 1, January March 2009
4. Eric Hsueh-Chan Lu, Wang-Chien Lee, Member, IEEE, and Vincent S. Tseng, Member, IEEE," A Framework for Personal Mobile Commerce Pattern Mining and Prediction", IEEE Transactions on Knowledge And Data Engineering, Vol. 24, No. 5, May 2012.
5. H. Kargupta and A. Joshi, "Data Mining to Go: Ubiquitous KDD for Mobile and Distributed Environments", KDD-2001, San Francisco, August 2001.
6. J. Han, V.S. Lakshmanan and R T Ng, "Constraint-based, Multidimensional Data Mining", COMPUTER (Special issue on Data Mining), 32(8): 45-50, 1999.
7. Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2003.
8. Kasun Wickramaratna, Student Member, IEEE, Miroslav Kubat, Senior Member, IEEE, and Kamal Premaratne, Senior Member, IEEE," Predicting Missing Items in Shopping Carts", IEEE Transactions on Knowledge And Data Engineering, Vol. 21, No. 7, July 2009.
9. Li-Der Chou, Member, IEEE, Nien-Hwa Lai, Yen-Wen Chen, Member, IEEE, Yao-Jen Chang, Jyun-Yan Yang, Lien-Fu Huang, Wen-Ling Chiang, Hung-Yi Chiu, and Haw-Yun Shin "Mobile Social Network Services for Families With Children With Developmental Disabilities" IEEE Transactions on Information Technology.
10. Luigi Lancieri, Member, IEEE, and Nicolas Durand "Internet User Behavior: Compared Study of the Access Traces and Application to the Discovery of Communities" IEEE Transactions On Systems, Man, and Cybernetics—part A: Systems And Humans, Vol. 36, No. 1, January 2006.
11. Manuel Fogue, Piedad Garrido, Member, IEEE, Francisco J. Martinez, Member, IEEE, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni, Member, IEEE," A System for Automatic Notification and Severity Estimation of Automotive Accidents", IEEE Transactions on Mobile Computing, Vol. 13, No. 5, May 2014.
12. Maya Nayak and Jnana Ranjan Tripathy: "Pattern Classification Using Neuro Fuzzy and Support Vector Machine (SVM) – A Comparative Study", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 5, May 2013.
13. N. Mlambo, "Data Mining: Techniques, Key Challenges and Approaches for Improvement", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 3, March 2016.
14. Paško Konjevoda and Nikola Štambuk, "Open-Source Tools for Data Mining in Social Science," Theoretical and Methodological Approaches to Social Sciences and Knowledge Management, pp. 163-176 .
15. Shangguang Wang, Member, IEEE, Zibin Zheng, Member, IEEE, Zhengping Wu, Member, IEEE, Fangchun Yang, Member, IEEE, Michael R. Lyu, Fellow, IEEE, "Reputation Measurement and Malicious Feedback Rating Prevention in Web Service
16. Recommendation Systems", IEEE Transactions on Services Computing, Vol. , No. , March 2014.

17. Shenghua Bao, Shengliang Xu, Li Zhang, Rong Yan, Zhong Su, Dingyi Han, and Yong Yu "Mining Social Emotions from Affective Text "IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 9, September 2012.
18. Xiao-Bing Xue and Zhi-Hua Zhou, Senior Member, IEEE, "Distributional Features for Text Categorization", IEEE Transactions On Knowledge And Data Engineering, Vol. 21, No. 3, March 2009.
19. Y.K. Raymond, Lau SAR Yunqing Xia, Yunming Ye Shenzhen "A Probabilistic Generative Model for Mining Cybercriminal Networks from Online Social Media", China.

This page is intentionally left blank

Music/Multimedia Technology: Creating Variations from Rhythm Generation and Melody Synthesis Processes of Hybridized Interactive Algorithmic Composition Model

Garba Joshua, Wajiga, Gregory Maksha

Federal University of Technology

ABSTRACT

The Hybridized Interactive Algorithmic Composition (HIAC) Model was developed by Dr. Etemi Joshua Garba under the supervision of Professor Gregory Maksha Wajiga (Garba and Wajiga, 2014a). The HIAC model is an aggregate of various algorithmic composition models in order to minimize the weaknesses experienced when such models are used singlehandedly in music composition.The hybridization of the models (at different stages of composition) capitalized on the advantages of such models. The HIAC model was presented as software framework. In this paper, however, we shall take a review of how variations are created from the rhythm generation and melody synthesis processes of the HIAC model. The variation aspect of the HIAC model is expected to be useful to software designers and application developers in the field of Music and Multimedia Technology.

Keywords : music and multimedia technology; variations; rhythm generation; melody synthesis; hybridized interactive algorithmic composition model.

Classification : H.5.5, H.5.1

Language : English

London Journals Press

LJP Copyright ID: 787716
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 17 | Issue 1 | Compilation 1.0

Music/Multimedia Technology: Creating Variations from Rhythm Generation and Melody Synthesis Processes of Hybridized Interactive Algorithmic Composition Model

Garba, Etemi Joshua[α] & Wajiga, Gregory Maksha[σ]

I. ABSTRACT

The Hybridized Interactive Algorithmic Composition (HIAC) Model was developed by Dr. Etemi Joshua Garba under the supervision of Professor Gregory Maksha Wajiga (Garba and Wajiga, 2014a). The HIAC model is an aggregate of various algorithmic composition models in order to minimize the weaknesses experienced when such models are used singlehandedly in music composition. The hybridization of the models (at different stages of composition) capitalized on the advantages of such models. The HIAC model was presented as software framework. In this paper, however, we shall take a review of how variations are created from the rhythm generation and melody synthesis processes of the HIAC model. The variation aspect of the HIAC model is expected to be useful to software designers and application developers in the field of Music and Multimedia Technology.

Keywords: music and multimedia technology; variations; rhythm generation; melody synthesis; hybridized interactive algorithmic composition model.

Author α σ: Department of Computer Science, Federal University of Technology (Modibbo Adama University of Technology) Yola.

II. INTRODUCTION

The development of the HIAC model became necessary since the existing algorithmic composition models do not single-handedly take care of all the stages of music composition. Therefore, the HIAC model harnesses the strength of existing algorithmic composition models in order to overcome their weaknesses – when they are used individually.

The models for algorithmic composition used included: Mathematical (Gilkerson and Owen, 2005), Grammar (Rule-Based) (Towsey, Brown, Wright and Diederich, 2009), stochastic (Microsoft Encarta Encyclopedia, 2009), Knowledge-based (Järveläinen, 2000), and Evolutionary (Genetic Algorithm) (Holland, 1975). The HIAC model is based on Interactive Algorithmic Composition approach where human creativity in music improvisation and composition is complemented by the leverage of speed and accuracy of the computer.

III. ALGORITHMIC COMPOSITION

Algorithmic composition is the technique of using algorithms to create music. According to Fernández and Vico (2013), algorithmic composition is the partial or total automation of the process of music composition by using computers programs. There have been many studies on automatic music composition using computer since the conception of the computer, and some automatic music composition models have been proposed (Unehara and Onisawa, 2009), (Todd and Werner, 1999) and (Espi, Ponce

de Leon, Perez-Sancho, Rizo, Inesta, Moreno-Seco, and Pertusa, 2009).

Music composition is perceived a complex and challenging activity for those not having musical knowledge or skill (Unehara and Onisawa, 2009). According to Garba (2003), music is the alternation of sound and silence. From the computational (logical) point of view, when sound is produced (or a note/tone is played) the binary value is 1, otherwise it is 0. Therefore, the binary musical concept perceives music as a stream of 1s and 0s. This means that musical ideas and concepts are logical – hence computable.

IV. RHYTHM GENERATION PROCESS OF THE HIAC MODEL

Rhythm Generation Process of the HIAC Model is based on both mathematical and grammar models. The Beat Binary code is the stream of bits required to represent a beat. See Table 1.

Table 1: Beat Binary Code

Note Type	Note symbol	Rest symbol	Beat Binary Code		
			Bits	Note Binary Value	Rest Binary Value
Whole			16	1000 0000 0000 0000	0000 0000 0000 0000
Half			8	1000 0000	0000 0000
Quarter			4	1000	0000
Eight			2	10	00
Sixteenth			1	1	0

$$B_{BC}=Random(Abs(2^{n}-1)) \quad (1)$$

Where: B stands for Beat, BC stands for Binary Code, Abs stands for absolute value and n specifies the type of note/rest assigned to one beat in a meter. The Rhythm Binary Code, which is made up of a measure, is a function of the Beat Binary code.

$$R_{BC}=\{B_{BC1},B_{BC2}, \ldots,B_{BCm}\} \quad (2)$$

Where: R stands for Rhythm, BC stands for Binary Code and m is the numerator that indicates the number of beats in a measure of a given meter. Note: R_{BC} logically represents a musical Rhythmic Pattern (RP). For example, an R_{BC} = {1010, 1000, 1110}, could be musically represented as shown in Figure 1.

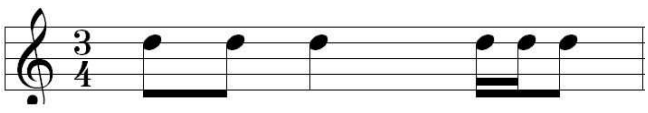

Figure 1: Musical notation of Rhythmic Pattern.

The rhythm generation process of the Hybridized Interactive Algorithmic Composition Model was developed from the concept formulation of the research by Garba(2012). The result of the rhythm generation process is a rhythm pattern devoid of melody. See Figure 2.

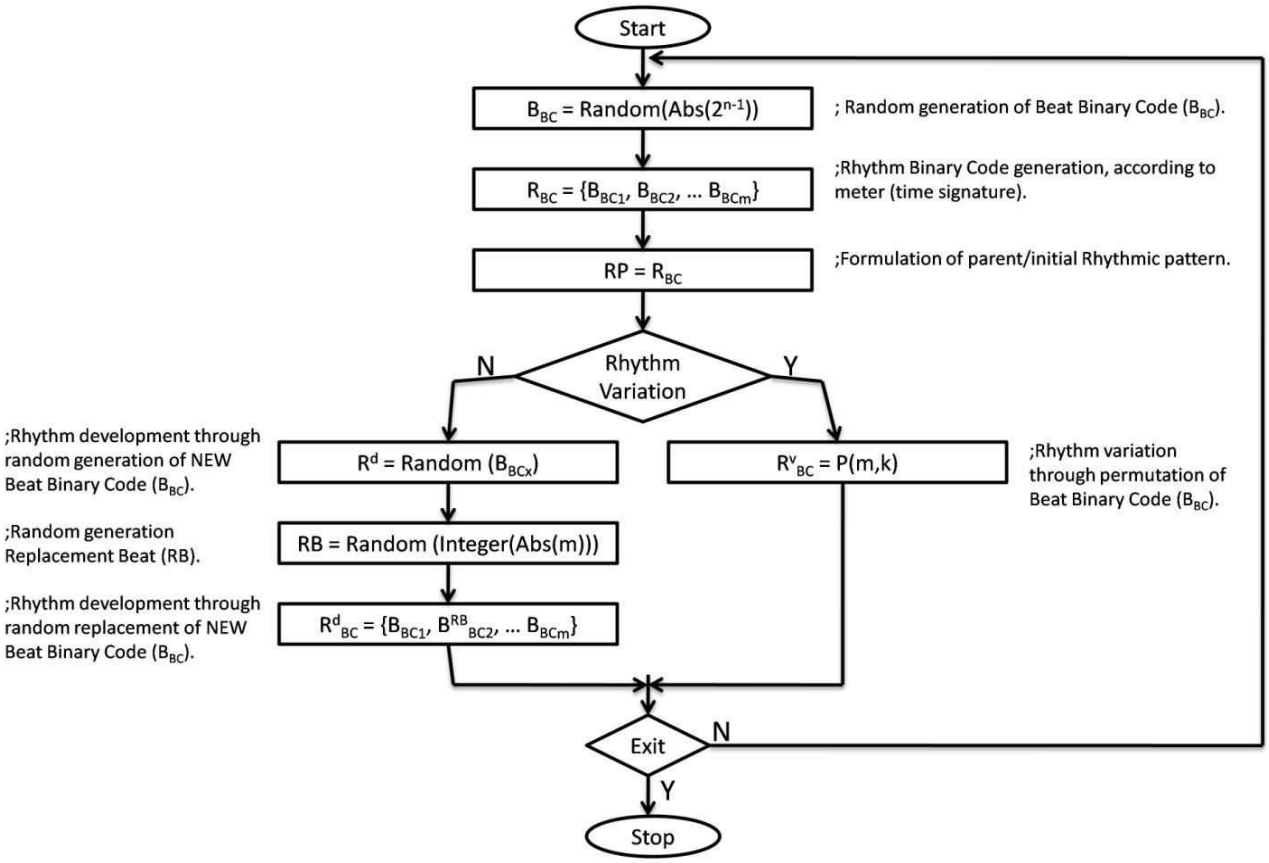

Figure 2: Rhythm Creation Process(Garba, 2012).

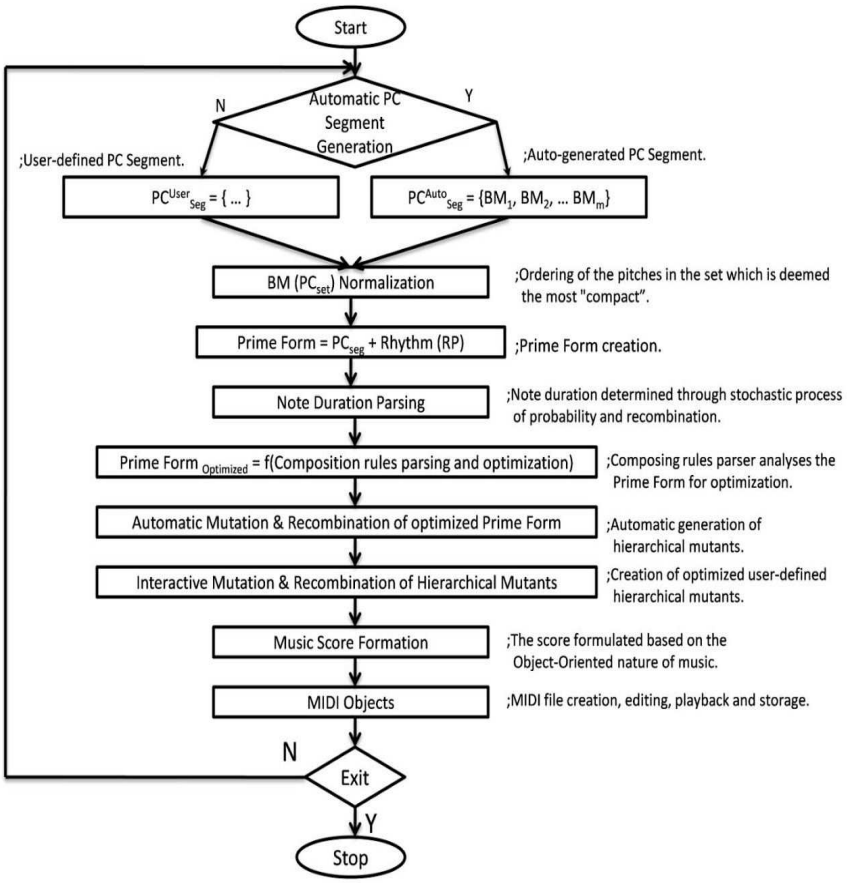

Figure 3: Melody Synthesis Process(Garba, 2012).

V. MELODY SYNTHESIS PROCESS OF THE HIAC MODEL

In the HIAC Model, the melody synthesis process involves several steps until the Prime Form of the melody is created (Garba and Wajiga, 2014b). See Figure 3 for full overview of the melody synthesis process. Note that the Prime Form is further analyzed by the Composition Rules Parser based on the basic prevalent contemporary music composition rules. Afterwards, the Prime Form is refined by the Optimizer. The optimization process could be done either manually by the composer or automatically. The final result of this stage is known as Optimized Prime Form. See Figure 4 for an example of an Optimized Prime Form.

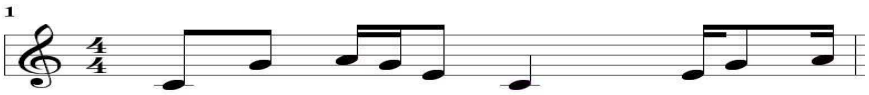

Figure 4: Optimized Prime Form.

VI. CREATING VARIATIONS IN HIAC MODEL

The repetition of musical patterns helps listeners to comprehend music without having because that leads to the experience of similarity and variations within the listening process. Musicians do use altered repeated musical patterns to enhance the communication and experience of their musical ideas ((Muller and Kurth, 2007), (Deliège, 2007) and (Meyer, 2000)).

Music variation is strongly supported in HIAC Model through hierarchical mutation and recombination of the Optimized Prime Form that gives birth to what is known as Hierarchical Mutant (which is a variant of the original Melodic Pattern). Hierarchical mutants could be produced from the Optimized Prime Form through these processes: retrogradation, transposition and inversion. See Figure 5.

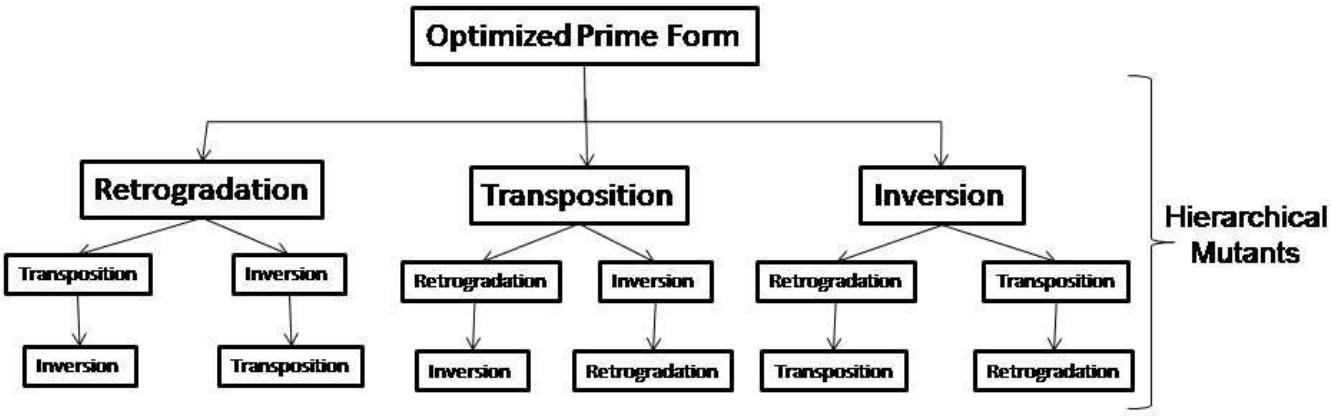

Figure 5: Hierarchical Mutation and Recombination of Optimized Prime Forms.

Transposition

Transposition mathematically captures the restatement of a melody at higher and lower pitch levels in a way that preserves intervals. Transpositional equivalence has also been in place in tonal theory. Any sonority (collection of pcs, or "pc set") can be transposed to another pitch level and retain its character (though the *function* may change in tonal music). For instance, Figure 6 shows a transposition of the Optimized Prime Form in Figure 4. Note that transposition neither changes the rhythm nor the intervallic structure of the melody.

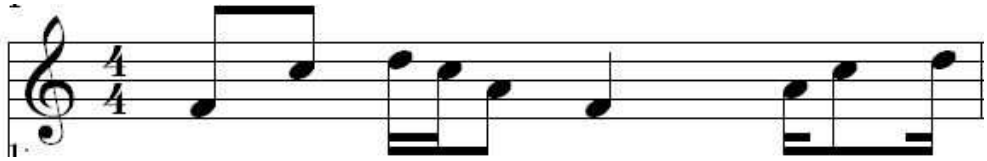

Figure 6: Transposition of the Optimized Prime Form (after transposing the musical piece by +5 cents, the tonic note is “F”).

Retrogradation

The retrograde form of the Optimized Prime Form is created by writing the notes in the original version in reverse order. The retrograde, or backward form of the Optimized Prime Form, is the possible manipulation that can be used in serial composition. Since it is simply the reverse of the prime form, it is most conveniently found in a matrix by reading the desired musical piece backwards (from right to left). Figure 7 shows the retrograde of the Optimized Prime Form in Figure 4. After retrogradation, though the rhythmic structure of the melody changes, the intervallic structure of melody remains unaltered.

London Journal of Research in Computer Science and Technology

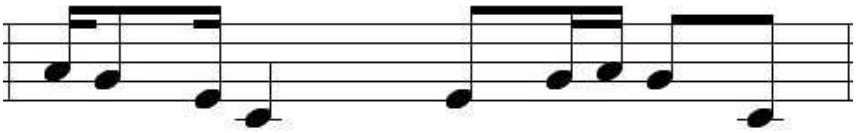

Figure 7: Retrogradation of the Optimized Prime Form (after retrogradation, "A" is now the tonic note of the musical piece).

Inversion

Inversion is another way to create the musical variation while preserving the intervallic sound of a melody (although it does not preserve the exact intervals). Inversion is the operation of turning the melody contour upside down across a horizontal line of symmetry. Figure 8 shows the graphical representation of the inversion operation, in which the Pitch Class sets (PC sets) are "flipped" around the 0 axis ((Nelson, 2007), (Huron, 1994), (Pendergrass, 2013), (Lewin, 2001), (Mead, 1988) and (Mead, 1989)).

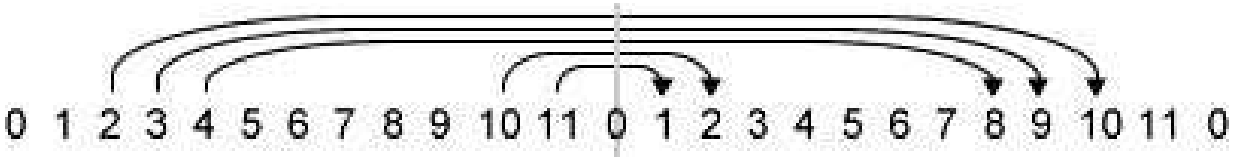

Figure 8: Horizontal line of symmetry used in the inversion operation of a Pitch Class sets.

Table 2 shows how an Inverted Prime Form is created from an original Prime Form using the horizontal line of symmetry in Figure 8.

Table 2: Creation of Inverted Prime Form

Prime Form	c	_	g	_	a	g	e	_	c	_	_	_	e	g	_	a
PC Number	0	_	7	_	9	7	4	_	0	_	_	_	4	7	_	9
Inverted PC Number	0	_	5	_	3	5	8	_	0	_	_	_	8	5	_	3
Inverted Prime Form	c	_	f	_	d#	f	g#	_	c	_	_	_	g#	f	_	d#

Melody Variation

Melody variation is realized by exchanging notes positions. That is, the permutations of the existing pitched notes, without altering the Rhythmic Pattern. This implies that a given Rhythmic Pattern could produce several Melodic Patterns through variation. Figure 8 shows the creation of a melody variation of the Optimized Prime Form in Figure 4.

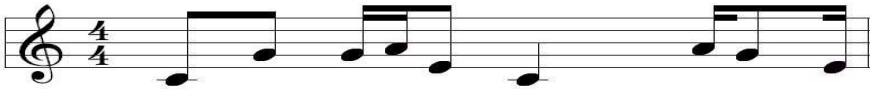

Figure 9: Melody Variation of the Optimized Prime Form.

Melody Repetition requires using the same Melodic Pattern over and over again without making changes to both the Melodic and Rhythmic Patterns. Melody derivation is achieved when the Melodic Pattern is kept constant, while the Rhythmic Pattern is changed. Melody contrast is created when both the Melodic and Rhythmic patterns practically remain unchanged, but long notes are replaced with shorter ones or vice versa.

VII. CONCLUSION

The composer can interactively re edit/ recompose parts of the hierarchical mutants to create more customized user-defined Melodic Patterns. This task is realizable through variation, repetition, derivation and contrast. At this stage the hierarchical mutants are analyzed and optimized by the Composition Rules Parser and Optimizer of the HIAC Model.

REFERENCES

1. Deliège, I. (2007). Similarity relations in listening to music: How do they come into play? Musicae Scientiae, Discussion Forum 4A, 9-37. Retrieved from https://jyx.jyu.fi/dspace/bitstream/handle/123456789/19315/Deliege_MS2007D4A.pdf?sequence=1
2. Espi, D., P. J. Ponce de Leon, C. Perez-Sancho, D. Rizo, J. M. Inesta, F. Moreno-Seco, and A. Pertusa. (2009). A Cooperative Approach to Style-Oriented Music Composition. Departamento de Lenguajes y Sistemas Informaticos University of Alicante, Spain. Retrieved from http://193.145.231.49/repositori/grfia/pubs/186/ wijcai07.pdf.
3. Fernández, J. D. and Vico, F. (2013). AI Methods in Algorithmic Composition: A Comprehensive Survey. *Journal of Artificial Intelligence Research*, 48 513-582.
4. Garba, E. J. (2003). *Computer Music – Rhythm Programming, Processing and Mastering*. Bloomington, Canada: Trafford Publishing Canada.
5. Garba, E. J. (2012). *Multimedia Technology: A Software Framework for Interactive Music Composition*. (PhD Thesis in Music/ Multimedia Technology), School of Pure and Applied Sciences Federal University of Technology, Yola, Nigeria).
6. Garba, E. J. and Wajiga, G. M. (2014a). Music/Multimedia Technology: Modeling and Simulation of the Hybridized Interactive Algorithmic Composition Model. *Software Engineering*. 2(2), 15-21. doi: 10.11648/j.se.20140202.11.
7. Garba, E. J., Wajiga, G. M. (2014b). Music/ Multimedia Technology: Melody Synthesis and Rhythm Creation Processes of the Hybridized Interactive Algorithmic Composition Model. *American Journal of Software Engineering and Applications*. 3(6), 106-111. doi: 10.11648/j.ajsea.20140306.17
8. Gilkerson, J., Li, W. and Owen, D. (2005). *An Introduction to Random Number Generators and Monte Carlo Methods*. Retrieved June 1, 2007, from http://www.mgnet.org/~douglas/Classes/cs521/rng-mc/RandomMonteCarlo2005.ppt.
9. Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. USA: Ann Arbor, University of Michigan Press (Second edition: MIT Press, 1992).
10. Huron, D. (1994). Interval-Class Content in Equally Tempered Pitch-Class Sets: Common Scales Exhibit Optimum Tonal Consonance. *Music Perception: An Interdisciplinary Journal*, 11(3) 289-305. University of California Press. Retrieved from http://www.jstor.org/stable/40285624. Accessed: 08/04/2016 15:28
11. Järveläinen, H. (2000). *Algorithmic Musical Composition*. Helsinki University of Technology, Telecommunications software and multimedia laboratory. Retrieved December 5, 2007, from www.tml.tkk.fi/Studies/Tik-111.080/2000/papers/hanna/alco.pdf.
12. Lewin, D. (2001). Special Cases of the Interval Function Between Pitch-Class Sets X and Y. *Journal of Music Theory*, 45(1), 1–30.

13. Mead, A. (1988). Some Implications of the Pitch-Class/Order-Number Isomorphism Inherent in the Twelve-Tone System: Part One. *Perspectives of New Music*, 26(2), 96–163.
14. Mead, A. (1989). Some Implications of the Pitch-Class/Order-Number Isomorphism Inherent in the Twelve-Tone System: Part Two. *Perspectives of New Music*, 27(1), 180–233.
15. Meyer, L.B. (2000). *The spheres of music*, Chicago, IL, USA: University of Chicago Press.
16. Microsoft Encarta Encyclopedia. (2009). *Stochastic*. Microsoft Encarta 2009 [DVD]. Redmond, WA, USA: Microsoft Corporation.
17. Muller, M. and Kurth, F. (2007). Towards structural analysis of audio recordings in the presence of musical variations. *EURASIP Journal on Applied Signal Processing*, 2007(1), 163–163.
18. Nelson, P. (2007). *Pitch Class Sets*. Retrieved from http://composertools.com/Theory/PC-Sets/PCSets1.htm#_Toc72662199. Accessed: 08/04/2016 13:46
19. Pendergrass, M. (2013). *Two Musical Orderings*. Department of Mathematics and Computer Science, Hampden-Sydney College, Hampden- Sydney, Virginia, May 24, 2013. Retrieved from http://www.hsc.edu/Documents/academics/MathCS/Pendergrass/twoMusicalOrders_rev1_arxiv.pdf. Accessed: 08/04/2016 16:45.
20. Todd, P. M. and G. M. Werner. (1999). Frankensteinian Methods for Evolutionary Music Composition. In Griffith and Todd, P. M. (Eds.), *Musical networks: Parallel perception and performance*, 313-339. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary? doi=10.1.1.71.943
21. Towsey, M., Brown, A., Wright, S. and Diederich, J. (2009). *Towards Melodic Extension Using Genetic Algorithms*. Queensland University of Technology Kelvin Grove, QLD 4059, Australia. Retrieved September 23, 2009, from http://eprints.qut.edu.au/169/1/towsey.pdf.
22. Unehara, M. and T. Onisawa. (2009). Construction of Music Composition System with Interactive Genetic Algorithm. University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573, JAPAN. Retrieved from http://www.Idemployee.id.tue.nl/g.w.m.rauterberg/conferences/CD_doNotOpen/ADC/final_paper/549.pdf.

The Characteristics of Good Systems

Chandra Amaravadi, Zachary L. Lessard

ABSTRACT

Software engineering attempts to produce systems that are "good systems" in terms of reliability, ease of maintenance etc. We take a broader definition of a good system as any general system that produces benefits that exceed initial expectations or intended scope or initial investment. There appear to be common characteristics that tie together such systems. These are hypothesized to include functional "goodness", good infrastructure, reliability, connect-ability, versatility and benefits that overflow/overwhelm the system's scope or initial investment. A case study approach involving four examples of what are regarded as "good systems" and four examples of what are regarded as "bad systems" fully supports this hypothesis. But support for the converse hypothesis, a bad system not having these characteristics was only 68.7%. The implications of these findings are discussed.

Keywords : good systems, system characteristics, strategic systems, systems theory, software engineering philosophy, information systems philosophy, complex systems analysis.

Classification : D.2

Language : English

London Journals Press

LJP Copyright ID: 646817
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 17 | Issue 1 | Compilation 1.0

The Characteristics of Good Systems

Chandra S. Amaravadi[α] & Zachary L. Lessard[σ]

I. ABSTRACT

Software engineering attempts to produce systems that are "good systems" in terms of reliability, ease of maintenance etc. We take a broader definition of a good system as any general system that produces benefits that exceed initial expectations or intended scope or initial investment. There appear to be common characteristics that tie together such systems. These are hypothesized to include functional "goodness", good infrastructure, reliability, connect-ability, versatility and benefits that overflow/overwhelm the system's scope or initial investment. A case study approach involving four examples of what are regarded as "good systems" and four examples of what are regarded as "bad systems" fully supports this hypothesis. But support for the converse hypothesis, a bad system not having these characteristics was only 68.7%. The implications of these findings are discussed.

Keywords: good systems, system characteristics, strategic systems, systems theory, software engineering philosophy, information systems philosophy, complex systems analysis.

Author α: School of Computer Sciences, College of Business and Technology, Western Illinois University.
σ: School of Law, Southern Illinois University.

II. INTRODUCTION

Systems theory is a very fundamental tenet of the Information systems field. The whole of software engineering revolves around the concept of systems. Bertalanffy (1950) developed systems theory based on observations from biology and physics. A biological organism consists of many sub-organisms that work together to process inputs and keep the organism alive. So he defined a *system* as anything composed of subsystems that work together towards the common goal of transforming inputs into outputs. A complex system can be understood by analyzing its sub-systems.

Systems theory led to systems engineering which is concerned with engineering complex systems. The goal of systems engineering is to manage complex systems so that they are reliable (Wikipedia '16a). It encompasses a number of ideas such as user requirements, systems architecture and reliability analysis that ultimately paved the way for software engineering.

In the 1980's Yourdon and others pioneered structured systems methodology to develop systems that fulfill their requirements and minimize maintenance (Page-Jones '82, Yourdon '80). They developed a number of tools and techniques for developing such systems based on the software development life cycle (SDLC). According to their methodology, during the *analysis* stage, systems were specified using Data-flow-diagramming, data dictionary and Structured English. This was carried out in a 'top-down' fashion starting with the system and 'decomposing' the subsystems. Thus DFDs could be drawn for several 'levels' of the system. In the *design* stage, DFDs were 'transformed' into a structure chart that exhibited certain characteristics. The structure chart, which is part of the high level design, consists of modules that called each other in a hierarchy. Design was 'top-down' in the sense that modules at the top co-ordinated the modules below them i.e "boss modules" call the modules below them. Lower level tasks such as input and output were carried out by modules at the lower levels of the structure chart.

Modules were treated as 'black boxes' considering only the 'inputs' and 'outputs' of each module. When carrying out the low level design, care was taken not to 'couple' modules tightly to other modules. *Coupling* is the degree of dependence one module has on another. A high degree of coupling results in problems in one module having ripple effects on other. Coupling is minimized by: removing unnecessary relationships, minimizing the number of necessary relationships, and decreasing the "tightness" of the relationships that are required. It allows for a module to be changed without affecting other modules, making systems easier to comprehend and maintain. (Page-Jones '82: 101-103). Minimizing coupling between modules was thus a design objective (ibid:101). Another key characteristic derived from structured programming is the concept of cohesion. *Cohesion* is the degree to which the activities performed by a module are related to other activities within the module. The more closely related the activities, the better the cohesion. Thus the notion of a good system from the software engineering point of view is to design complex systems as modules, using a top down design strategy, and as black boxes, to minimize coupling and maximize cohesion. These concepts are further solidified in the object oriented methodologies. In object oriented approaches, packaging of data and methods into objects ensures high cohesion. Restricting access to methods through declaring an object as public or private controls coupling. These practices reduce errors and result in more reliable and maintainable systems.

In a different vein, successful system implement -ations of the 1980's led Wiseman (1985) to postulate the concept of a strategic system. A *strategic system* is defined as a system that supports or shapes the competitive strategy of an organization (ibid). Following the generic strategies introduced by Porter, a firm can make several strategic thrusts including cost leadership, differentiation, innovation, growth and strategic alliances. Wiseman (ibid) provided several examples of strategic information systems including McKesson's distribution system for drugstores and supermarkets, Banc One's credit card processing network, American Airlines' Sabre system, Benetton's apparel system and Walmart's merchandise information system to illustrate these different strategies. They also shared a number of characteristics in common – they are mainly infrastructural, supported a critical process/ process group or supported the value chain of the company. Infrastructural systems provided communications (Banc One's credit card processing network) or linked the company with its customers or suppliers (McKesson's drug distribution system) or provided access to other informational resources. As an example of the latter, PaineWebber (now part of Swiss Bank) negotiated with State Street Bank to enable its customers to use the MasterTeller network. Thus it exploited the information resources of other customers. Benetton's system supported its value chain - order entry, manufacturing as well as dynamic order management (ibid). All of these systems are reported to have produced great benefits for the host organization. For McKesson, benefits included reduction of sales force from 700 to 15 with sales increasing from $922m to $4.8b (Clemons and Row '88). There were also indirect impacts on the industry such as reduction in number of participants from 180 to 90 and increase in market share for the top four participants (ibid). An added bonus was that McKesson was able to leverage its distribution expertise into other business areas such as veterinary supplies, beverages and general merchandise.

These cases lead one to suspect if there is something more to a good system beyond good software design i.e. characteristics such as good infrastructure, core process support etc. If good systems had generalizable characteristics, these must be true regardless of whether they are information systems or other general systems fulfilling the system definition. To test this concept, we developed a set of criteria for a 'good

system' -- it is hypothesized that these criteria hold for good systems and are absent in bad systems. We selected four cases of what are regarded as "good systems" and four cases of what are regarded as "bad systems" to see if the hypotheses are valid. If validated, such criteria can be used to evaluate systems at an early stage to correct problematic systems and ensure that system investments yield desired results.

2.1 A description of the selected criteria

As discussed in the previous section, two streams of thought serve as inspiration for the present study. The first is the notion of a good system from the software engineering area and second the notion of a good information system as a strategic system. From the first we have characteristics such as modularity, black boxedness, low coupling, high cohesion which would result in systems that fulfill requirements, are readily maintainable and reliable. From the second, we have characteristics such as infrastructural ability, connectability and the notion of secondary and tertiary benefits. We discuss these and additional criteria in the following. We need to emphasize again that the discussion is not restricted to information systems but extends to any type of system fulfilling the definition of a system.

2.2 Fulfills its functionality

The philosophy of structured systems is to develop systems that fulfill their intended need. The entire SDLC process is geared towards this end (Page-Jones '80). Accuracy of the analysis stage is often considered key to developing systems that fulfill their functional requirements. So the first test of a good system is whether or not it fulfills its functionality in the present as well as the future, and for the purpose for which it is designed. Further since systems function under a variety of conditions, a good system should fulfill its functionality under all conditions.

2.3 Is Infrastructural

An infrastructure is defined as the collection of basic physical, organizational structures and facilities needed for the operation of a system (Oxford dictionary '15). For an airplane, the combination of airframe, fuselage, wings, rudder and engine forms the infrastructure. The significance of infrastructure arises from the nature of systems -- open systems are characterized by exchange of energies (Bertalanffy '50). In biological systems, fluids and chemicals are exchanged among components. Impediments to such flows will adversely affect the functioning of the system. Thus a good infrastructure is an indispensable requirement for a good system. A good system facilitates the flow of information/ goods within the system's scope. If the system's scope is an organization, then a good system should facilitate the flow of materials within an organization.

2.4 Is easily Connectable

It is hypothesized that a good system possesses a high degree and ease of connectability. This idea is related to coupling and modularity. If a system is modular, it enables it to be part of a larger system or enables it to be the host system for another system. Coupling is the other side of the coin. Low coupling and high cohesion lead to modularity and ease with which a system can be connected to another system. A simple example is a railway carriage (smaller system) being connected to a train (a larger system). If the coupling is high (electricity, vacuum, hydraulics) the process will be difficult. Low coupling on the hand, is essential to use ability. Connectivity to systems outside a system's environ -ment allows for goods or information to move freely into and out from the system just as carriages *connected* inside the train allow passengers to move back and forth. It enables a system to take advantage of assets in other systems and thus enhances their versatility. As discussed earlier, PaineWebber using the information assets of State Street Bank exemplifies this situation (Wiseman '85).

2.5 Is Adaptable/Versatile

A good system it is hypothesized, must be adaptable and versatile. It should be capable of being modified easily to serve a wide variety of functions. This has been illustrated by the case of Sabre system for American Airlines (Sabre Holdings 2015). Sabre was used not only for reservations but for crew scheduling and flight forecasting (ibid). To be considered adaptable and versatile, the system must readily lend itself to being utilized in a large number of potential scenarios and must readily shift to meet changes in its operating environment. This readiness for utilization appears to be a hallmark of a good system.

2.6 Is Reliable

Reliability is one of the main characteristics of a system as identified in the introduction. It is defined as the percentage of time the system is operational or can be measured in terms of % of failures. It will be operationalized differently depending on the type of system being dealt with. For airlines, reliability is defined as percentage of on-time arrivals (Watson et al. '06) whereas for aircraft it is given as the number of plane crashes per million departures (Boeing '14). Needless to say, reliability affects confidence in the system and therefore its usability, It is a required quality of a good system.

2.7 Produces Additional Benefits (or problems)

It is hypothesized that a good system provides benefits that are disproportionately high when compared to the initial investment (Amaravadi '05). The $40 million initial investment in SABRE has resulted in a spinoff worth $6.2b dollars (McCartney 1999). This may be evidence of the multiplicative benefits of good systems. Sabre revolutionized the airline industry by enabling computerized flight listings (ibid). Judging by this and other strategic systems, the impacts of a good system extend to different parts of the organization and in some cases to the industry and ultimately to society. Conversely a bad system should result in problems that exceed the scope of the system. For example, a badly designed building could cause problems in roofing, wiring, heating, ventilation and occupant movement.

III. METHODOLOGY

The discussion above leads to the following Hypotheses:

Hypothesis H1: A good system has all or most of the following characteristics: fulfills its functionality, has good infrastructure, ready connect-ability with other systems, good versatility/adaptability, good reliability and produces benefits far exceeding the initial investment (or scope of the system).

Hypothesis H2: A bad system has all or most of the following characteristics: does not fulfill its functionality, has poor infrastructure, poor connect-ability with other systems, poor adaptability, poor reliability and creates problems that exceed the scope of the system.

To test these hypotheses, the authors selected four cases of what are popularly regarded as good systems and four cases of what are popularly regarded as bad systems. An attempt was made to select systems that are from very diverse domains. The criteria for selection was: 1) does it satisfy the definition of a system?, 2) Is it sufficiently complex to be interesting? 3) Is it a dynamic system rather than a static system such as a road network? This requirement is imposed since system behavior is part of the hypotheses. 4) Is the system widely regarded as a good (bad) system? 5) Is sufficient literature available to verify the hypotheses? This requirement entailed selecting systems that are in the U.S. Then using available sources, authors evaluated the systems against the criteria. A score of '1' was given if a good system fulfilled a particular criteria, '0.5' if it did not completely fulfill it. If a characteristic did not apply, it is labelled 'NA' and given a score of '0'. These were added across each system and across all four good systems to give the confidence level. The reverse was done for a bad system. If a bad system did not fulfill a criteria, a score of '1' was assigned. Then these were also tallied. This

process was necessarily subjective and is one of the limitations of this study. System descriptions are given to enable readers to verify author's perceptions.

IV. CASES OF GOOD SYSTEMS

Cases of good systems include an Ecommerce system (Amazon.com), a Supply chain management system (Walmart), a mobile device (Apple iPhone) and a search engine (Google). Although all of these are related to information technology this was not intentional.

4.1 Amazon.com Ecommerce site

The First system the authors have selected is the highly regarded e-commerce site of Amazon.com (Post '12). The company initially started selling books, but soon branched into music, movies, software, household goods, home improvement and video games among others. The site has a well developed infrastructure that allows for 24/7 product display and shopping. Customers can shop for any item from any division in any part of the web site. In 2000 Amazon.com overhauled its systems using DBMS from Oracle, logistics from Manugistics, Analytics from SAS and Excelon to support business to business integration (ibid). Amazon's merchant and marketplace systems allow merchants to sell their products on the Amazon.com site. This is enabled by Excelon which allows partners with limited IT to connect into its systems in real time. The site also supports the formation of communities and allows for contributions from affiliates. The website has additional benefits exceeding the initial scope of their system -- their e-commerce infrastructure is so effective that they have turned it into a product and the company now sells web services (Amazon web services) to companies such as Sears, Bebe, Marks and Spencer among others (Wikipedia 16b). Additionally, the price check application they developed allows a customer to check the price of goods in a store to see what the prices would be on amazon.com (Hane '12).

4.2 Walmart's Information System

The second system the authors selected is the much regarded supply-chain management system of Walmart (Lu '15). Walmart's homegrown systems support its strategy of cost minimization and rock bottom prices. Their systems have been behind its phenomenal growth (Gallaugher '12). The company founded in 1962 now has over two million employees and 11,000 stores and is the largest retailer in the world (Wikipedia 16c). Their philosophy of utilizing centralized and common systems/platforms ensures connect-ability and good infrastructure (Post '12). All inventory items in store are bar coded and purchases are scanned and recorded by item and date of sale at the cash registers. Other items purchased by the customer are also recorded this way. This makes it possible to know how the items are selling as well as how they are selling relative to other stores and other times of the year. The data is stored in a large 423 Terabyte data warehouse (Gallaugher '12). This basic system provides a good foundation for its other activities. The sales information is shared with Walmart's more than 5,000 suppliers through the Retail Link system and drives all inventory decisions (ibid, Holstein et al. '98). Walmart uses the data to stock up on fast selling items and reduce inventory on slow selling items. The sales information together with demo -graphic data allows Walmart to customize its stores to individual regions (Holstein et al. '98). Walmart's various systems function together to fulfill the company's strategy. Employees are provided with VOF (Voice Based Order Filling) to direct them to item locations and place inventory orders, thus saving time and reducing costs (Purpura '97). Its web site, although not as highly regarded, is linked with its homegrown system (Post '12). In addition Walmart is allowing its customers to check out using 'Walmart App' on their iPhones (Wohl '15).

4.3 Apple iPhone

The Apple iPhone is the most successful smart phone to date with more than a billion units sold as of 2016 (Statista 16a) or 23% of all smartphone sales (Statista '16b). The main features of the iPhone include its high resolution touch interface,

London Journal of Research in Computer Science and Technology

one click access to applications, built-in wi-fi connectivity, ease of texting and messaging, camera, video conferencing, internet browsing and digital music capability (Want '10). These features are in addition to its voice communication capability. Thus the basic infrastructure consists of the communications, interface, networking and multimedia capability. Apple devices are known for their versatility. 'Apple Apps' make the iPhone very versatile. These apps include: navigation, bar code scanning, dictation, credit card processing, invoicing and remote control functionality for home appliances among millions of others (Appstore '15). The combination of powerful technologies and an open application model make business applications such as payment processing and workflow possible. The 'apps' have been used for everything from identifying fonts (Turner '09) to properly tilting a patient during a C-section (Ramamoorthy and Bailey '12). Medical students use the iPhone as a substitute to referring to books (Chatterly and Chojecki '10). The usefulness, versatility and benefits of the device defy description.

4.4 Google Search Engine

According to Wikipedia, the Google Search Engine is the internet's most popular search engine, handling more than 3.5 billion searches a day and accounting for 64.5% of the market share (Wikipedia '16d). The powerful infrastructure that Google has developed connects users to content they desire through the use of a giant index of ranked searchable links (Barroso et al. 2003). It has used this infrastructure effectively to diversify into a number of related products and services that combine to make it a technology juggernaut. The search engine can be easily embedded into other web sites such as 'cnn.com' for localized searches. Also the same technology can be used to search blogs, journal articles, images, and RSS feeds. Their "adsense" technology is tied to the searches such that a search for a product triggers advertisements it. Thus a search for "flowers" brings up advertisements of florists. This technology resulted in 2015 revenues of around $67.3 billion (Statista '16c). Also when certain keywords are entered unit, currency and time conversions, weather, stock quotes and airline schedules are triggered. The "Universal Search" feature combines search information from multiple pages and presents it as a summary (ibid). These features amply exhibit its versatility, connectability and its ability to generate additional benefits.

V. CASES OF "BAD" SYSTEMS

Cases of bad systems include an aircraft (DC-10), an operating system (DOS), a healthcare system (U.S. Healthcare) and database management systems (hierarchical systems).

5.1 McDonnell Douglas DC-10

The DC-10 was a wide bodied aircraft that was introduced by the McDonnell Douglas corporation in 1967 and certified airworthy in 1971 by the FAA (Kull '14). There was also a tanker version of the DC-10, the KC-10 which exhibits by its existence, some degree of adaptability. Although the DC-10 ultimately proved to be a reliable aircraft, it has been maligned due to a number of spectacular crashes (ibid). It is due to this reputation that we included the plane in this study. On 12th June, 1972, a cargo door blew out from the aircraft during a flight between Detroit and Buffalo. The resulting decompression caused the floor over the cargo compartment to cave in, damaging flight control cables. A similar problem caused the deaths of 346 passengers and crew in 1974. In another incident, the engine separated from the left wing and flipped up over the top of the aircraft resulting in the deaths of 271 passengers (ibid). In all there were 56 aviation occurrences including 32 hull-loss accidents with a total of 1,262 occupant fatalities (Wikipedia '16e). Clearly there were problems with the aircraft body which led to these fatalities, including the cargo door design, cargo roof design and engine mounting. Failure in one subsystem has also resulted in problems with other subsystems further reflecting a bad design. For example cargo area decompression led to floor collapse, which led to loss of aircraft control.

Despite occupant deaths, reliability of the aircraft (2.94 accidents per million departures) was comparable to Boeing 747 (2.85) and the Airbus A300 (2.29) (Boeing '14).

5.2 DOS and MS-DOS

DOS was introduced in 1981 as the operating system for the IBM PC (Wikipedia 16f). Although there were three different brands, PC-DOS, MS-DOS, DR-DOS the most popular variant was MS-DOS and we refer to this here between 1981-1985. It was the dominant operating system for PCs until replaced by Windows in the mid 1990's. DOS was designed as a single tasking, single user system based on the Intel 8086 microprocessor (Paterson '15). Its core modules include file/directory management, memory management, command interpreter and kernel programs (Verma '09, Paterson '83). Since the main purpose of operating systems is to control the hardware, they are necessarily coupled to a particular hardware. This makes adaptability a non-issue. DOS was hastily developed and it had a number of limitations that continued into later versions. The first version was disk based and had to be physically loaded into RAM from floppy disks. Later when hard disks were introduced, DOS could be configured to run from hard disk. The initial versions also did not have device drivers, and these had to be separately installed until MS-DOS version 5 was introduced in 1991 (Ferelli '92). The major limitation of DOS was the 640K limit on memory imposed by the underlying 8086 architecture (Gookin '91). Programmers could address only 640K memory in 64K RAM segments causing them to write program chunks that utilized only 64K segments. This issue was rectified with the introduction of the 80386 processor even though maintaining backwards compatibility required extra programming on the part of the programmers (McDugall 2013). This was a nightmare by all accounts (ibid). There was also a file size limitation of 65K owing to the 16 bit architecture which dictates the address size (Disc 2015). This was corrected in later versions, but due to the "Fat16" volume size, file sizes were limited to 2 GB (ibid). To round out the list of limitations, the command line interface was user unfriendly and required users to remember commands like "dir *.*, copy *.*" etc. The command line interface was replaced in the mid-nineties with the introduction of Windows.

5.3 The U.S. healthcare system

According to Wikipedia the health care system is the organization of people, institutions, and resources that deliver *health care* services to meet the *health* needs of target populations. The U.S. Health care system has received criticism from many quarters, which justifies its inclusion here (see for example Brodwin '14). On a number of measures (80 in all) lumped into the dimensions of quality, efficiency of care, equitable-ness and health indicators, the U.S. ranked last when compared to a list of 11 advanced countries that includes Australia, Canada and European countries (Davis et al. 2014). The per-capita healthcare spending costs are also highest when compared to these countries. Other than costs, there are problems of uninsureds, inefficiencies in co-ordination among different agencies, patient deaths due to hospital errors, inefficient information exchange, billing inefficiencies among others (ibid, Jost '06). These problems amount to a poorly functioning system. The basic infrastructure of hospitals, physicians, nurses, staff and facilities are present in the U.S. although to a lesser extent than some European countries (Anderson and Squires '10). Co-ordination of information about patient has generally been described as poor (Jost '06). According to one study, the U.S. ranks 6th out of 11 in this respect (Davis et al. 2014). According to another study between 220,000-440,000 patient deaths are preventable with effective healthcare (Npr '13). Adaptability of the system cannot be judged as it has not been adapted to different systems of medicine, but it has certainly proved versatile in the face of new technologies. There are many additional problems caused by the healthcare system. Patient mobility between physicians is restricted due to the in-network limitation imposed by many HMOs (Garson 2000). Patients also do not know their financial liability until after their treatment.

5.4 Hierarchical Database Management Systems

Hierarchical database management systems are exemplified by IMS (Information Management System), a database management system developed by IBM in 1966 (Wikipedia 16g). In hierarchical database management systems, records are defined using "segments" i.e. what would correspond with tables in relational databases are defined as segments in hierarchical systems (Panneerselvam 2004). Each segment has a number of fields that correspond to attributes in relational databases. Segments can have child segments such as a department having employees. The structure of the database is thus encoded in the data definition in the form of data definition language (DDL) statements. This hierarchical structure embedded in the DDL in the form of segments and "records" forms the infrastructure of hierarchical dbms. This is awkward at best since it is suited for 1:M relationships in the data (for example the relationship Department: Employees) rather than M:N relationships which tend to be more frequent (for example the relationship Companies: Suppliers) and for which the relational model is eminently suited (ibid). Retrieval also presented problems. To retrieve data it is necessary to "navigate" the structure. For example, a segment can be retrieved directly with 'GU' (Get Unique). 'GN' (Get Next) gets the next occurrence of the segment. 'GNP' gets the next occurrence of a child segment under a particular parent. After navigating the structure, data manipulation operations such as 'ISRT' and 'DELT' could be performed (ibid). Thus there are two sets of operators, one for navigation and one for data manipulation, whereas, relational systems required only one set of operators for data manipulation and retrieval (Gibbs '85). Another great disadvantage of the hierarchical scheme was for the need to know database structure for retrieval since it is necessary to navigate this structure which quickly becomes complex when there are more than a dozen segments with interrelationships. Hierarchical systems fell by the wayside as a result of these disadvantages (Prescott et al. 2010). However, because of the "tree" organization of data and indexing, retrieval was very fast. The DBMS can be connected to a transaction manager for use in transaction processing environments. Speeds of up to 100,000 TPS (transactions per second) have been recorded (Wikipedia '16g). Not surprisingly hierarchical systems have been used reliably in the banking industry (ibid).

VI. RESULTS AND DISCUSSION

Results of the study are shown in summary form in *Table 1* below and in detail in *Appendix 1*. It is seen that there is 100% support for Hypothesis 1; a good system has the characteristics of fulfilling its primary function well, of having a good infrastructure, being readily connectable, being versatile/adaptable, being reliable and producing a number of secondary benefits. Unfortunately, the case is not so clear-cut in the case of bad systems. As seen from table 2 there is only a 68.75% support for Hypothesis 2 -- not fulfilling its primary function, not having a good infrastructure, not being connectable, not being adaptable, reliable and producing problems instead of benefits. One reason for the asymmetry in results may be that only a few of the characteristics when unfulfilled are sufficient to move a system from the "good" category to the "bad" category i.e. only a few of the characteristics may be enough to prevent a system from being a good system. Bad systems can arise from problems in the subsystems which in turn leads to some or most of the criteria to be unfulfilled. This is seen in the cases of DOS and DC-10. In DOS the problem was with the microprocessor used for the PC whereas for the DC-10 it was the cargo door, engine mount and floor design. Component problems could result from sub-optimal design decisions that are forced by design constraints. These are evidenced in the case of DOS. Here memory limitations clearly led to several problems that affected application programs for years. Bad systems could also result if the system is so complex that the inter-relationships cannot be controlled. We see this in the case of the U.S.

healthcare system. Here there is a complex relationship between providers, physicians, pharmacies and insurance companies. The Apple iPhone presents the opposite end of the spectrum of systems. Here the individual subsystems have been endowed with powerful capabilities that applications (iApps) could exploit resulting in a very versatile device.

In this research there was no attempt to control for system complexity, although an attempt was made to ensure that the systems are roughly comparable to one another. It would seem that simple systems such as pumps or internal combustion engines evolve fairly rapidly in the milieu of modern business and technological forces. The same does not appear true for complex systems. These perhaps present a degree of "wickedness" that defies good organization/ design (Selfridge et al. '84). Secondly if complex systems have human elements as in healthcare or aircraft maintenance, important system characteristics such as functionality and reliability are affected. The situation is aggravated if different parts of the system are under control of different entities. Therefore it is safe to say that the more complex the system the more the likelihood of it not satisfying all the criteria and therefore being a "bad system." Obviously such systems are undesirable since like DOS and the DC-10, these cause problems to all. The other side of the coin is whether or not the satisfaction of these characteristics at the design stage would ensure that system would be a good system. Empirical verification is needed to support such a conclusion.

Table 1: Summary Evaluation of Good System Characteristics

Type of system	*∑ Score 'Yes' on system characteristics*	*∑ Score 'No' on system characteristics*	*Confidence*
Good system	24	0	H1: 100%
Bad system	6.5	16.5	H2: 68.7%

VII. CONCLUSIONS AND IMPLICATIONS

Based on the cases presented, there is strong support for the hypothesis that good systems have the characteristics as identified in this research, viz. fulfills its primary function, good infrastructure, is easily connectable, is adaptable/ versatile, reliable and finally produces benefits that exceed the system's scope. A primary limitation of the study is its qualitative nature. Evaluation of whether or not the criteria was fulfilled was subjective. There was also no attempt to control for the size or complexity of the system although all systems presented have their own levels of complexity. There is no assurance that the criteria are complete. Some criteria such as modularity, configurability, flexibility, evolvability, architecture, feedback and self organization have not been considered. These, it was felt would not be applicable to all types of systems. For example, an airplane is a self-contained system and modularity would not be relevant to whether or not it is a good system from the point of view of transporting passengers. Similarly the notion of an architecture is not appropriate to systems such as the healthcare system. There is obviously opportunity for further development.

The findings have to be further verified and if possible extended. If empirically proved these factors have important implications for designing systems, especially those satisfying the broader definition of systems such as dams and buildings. An extended list of characteristics could be utilized as a checklist for developing good systems. In addition, the research suggests that high- level technological or business constraints or complex interrelationships can result in a poor design that causes innumerable problems. This is also an idea worth exploring because the

economic and other costs of a badly designed system are staggering.

REFERENCES

1. Amaravadi, C. S., (2004). "The Laws of Information Systems," *Journal of Management Research*, 4(3), pp 129-137, December.
2. Anonymous (2010). "Walmart makes fashion statement with item-level RFID". Material Handling Management July 26th (no volume, issue or page information).
3. Anderson, G. F., and Squires, D. A. (2010). Measuring the US health care system: a cross-national comparison. Issue Brief (Commonwealth Fund), 90, 1-10.
4. App Store (2015). Apple Inc., https://itunes.apple.com/us/genre/ios/id36?mt=8.
5. Barroso, L. A., Dean, J., and Hölzle, U. (2003). Web search for a planet: The google cluster architecture. *IEEE Micro*, 23(2), 22-28.
6. Boeing (2014). Statistical Summary of Commercial Jet Airplane Accidents Worldwide Operations|1959–2013 (August, 2014). Retrieved from http://www.boeing.com/news/techissues/pdf/statsum.pdf current July 2015.
7. Bertalanffy, L. V. (1950). An Outline of General Systems Theory," *The British Journal for the Philosophy of Science,* 1:2 August, pp. 134-165.
8. Brin, S., and Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18), 3825-3833.
9. Brodwin. E., 2014. 11 Charts that show exactly what's wrong with the US Healthcare System. *Business Insider*, September 23. http://finance.yahoo.com/news/11-charts-show-exactly-whats-133000491.html.
10. Chatterley, T., & Chojecki, D. (2010). Personal digital assistant usage among undergraduate medical students: exploring trends, barriers, and the advent of smartphones. *Journal of the Medical Library Association* : JMLA, 98(2), 157-160.
11. Clemons, E. K., & Row, M. (1988). McKesson Drug Company: a case study of Economost—a strategic information system. *Journal of Management Information Systems*, 5(1), 36-50.
12. Davis, K., Stremikis, K., Squires, D., Schoen, C. (2014). Mirror, Mirror on the Wall, How the performance of the U.S. Healthcare system compares internationally, *Commonwealth* Fund, June. http://www.Commonwealthfund.org/~/media/files/publications/fund-report/2014/jun/1755_davis_mirror_mirror_2014.pdf
13. Disc, (2015), PC Large-File Limitations, Data Interchange Service Company, http://www.3480-3590-data-conversion.com/article-large-files.html, Current July 2015.
14. Ferelli, M. (1992). Device Drivers Seek Universality, *Computer Technology Review*, 12(3), March, p34.
15. Garson, A. (2000). The US Healthcare System 2010 Problems, Principles, and Potential Solutions. *Circulation*, 101(16), 2015-2016.
16. Gallaugher, J. (2012). "Data asset in action: Technology and the rise of Walmart." Getting the most out of information systems, Creative Commons, pp 467-470.
17. German, K., (2007). Apple iPhone review, CNET reviews, June 30, http://www.cnet.com/products/apple-iphone/
18. Gibbs, S.J. (1985). Conceptual modelling and office information systems. In Tsichritzis, D. (ed.), Office Automation, Springer-Verlag. pp. 193-225.
19. Gookin, D. (1991). Memory managers: Taming DOS' RAM. InfoWorld, 13(49), 69-69, 72+
20. Hane, J. P. (2012), Amazon's Ever-Expanding Empire, *Information Today* February 29(2),
21. Holstein, W., Sieder, J., Svetcov, D. (1998). Data-crunching Santa. *U.S. News & World Report*, 125 (24), p44.

22. Hull, K., (2014). *Airline Reporter*, A Historical Look at the DC-10 Before its Final Passenger Flight, February 19th. [http://www.airlinereporter.com/2014/02/historical-look-dc-10-final-passenger-flight/]
23. Jost, T. S. (2006). Our broken health care system and how to fix it: an essay on health law and policy. *Wake Forest Law Rev*iew, 41, 537.
24. Lu, C. (2015) "Incredibly successful supply chain management: how does Walmart do it?" TradeGecko. https://www.tradegecko.com/blog/incredibly-successful-supply-chain-management-walmart.
25. McCartney, S. (1999). AMR announces its plan to spin off Sabre Holdings. Wall Street Journal, Dec 14. http://www.wsj.com/articles/SB945130846522949523 accessed July 2015.
26. Myers, D., (1983). "Porting MS-DOS", *Systems International*, 11(9), September, pp. 106-107.
27. McDougall, S., (2013). Limitations of the IBM PC Architecture. The World, http://world.std.com/~swmcd/steven/rants/pc.html.
28. Npr.org (2013). How many die from medical mistakes in U. S. Hospitals? September 20 http://www.npr.org/blogs/health/2013/09/20/224507654/how-many-die-from-medical-mistakes-in-u-s-hospitals, Accessed July 2015.
29. Oliver, D. W., Kelliher, T. P., Keegan, J. G. (1997). *Engineering Complex Systems with Models and Objects*. McGraw-Hill.
30. Oxford dictionary (2015). http://www.oxforddictionaries.com/us/definition/american_english/infrastructure.
31. Page-Jones, M. (1980). *The Practical Guide to Structured Systems Design*. New York, NY: Yourdon Press.
32. Panneerselvam, R. (2004). Database Management Systems. PHI Learning Pvt. Ltd.
33. Paterson, T. (2015). "An inside look at MS-DOS", Byte Magazine, June 1983. http://www.patersontech.com/dos/byte%E2%80%93 inside-look.aspx
34. Post, G., (2012). Cases in MIS, https://www.jerrypost.com/MIS/MISCases2012.pdf.
35. Prescott, M. B., McFadden, F. R., & Hoffer, J. A.. *Modern Database Management*. Pearson Higher Education, 2010
36. Purpura, Linda. (1997). At Wal-mart, voice based order filling speaks up, *Supermarket news*, June 30, http://supermarketnews.com.
37. Ramamoorthy, K. G., and Bailey, K (2012). iPhone for measuring tilt during 150 caesarean section, *Anaesthesia*, 67(5), 551-552, May.
38. Rogowski, M. (2014). Without much fanfare Apple has sold its 500 millionth iPhone, Forbes, March 25, http://www.forbes.com/sites/markro-gowsky/2014/03/25/without-much-fanfare-apple-has-sold-its-500-millionth-iphone/.
39. Selfridge, O., Rissland, E., and Arbib, M. (1984). (editors), Adaptive control of ill-defined systems, Plenum Press.
40. Sabre Holdings. (2015). Sabre History. *Sabre Holdings*. http://www.sabre.com/home/about/sabre_history Accessed November 2016.
41. Statista (2016a). Global Apple iPhone sales from 3rd quarter 2007 to 4th quarter 2016 (in million units). Current August 2016, http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/
42. Statista (2016b). Number of smartphones sold to end users worldwide from 2007 to 2015. Current August 2016, http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/
43. Statista (2016c). Google's ad revenue from 2001 to 2015 (in billion U.S. dollars) Current November 2016, https://www.statista.com/statistics/266249/advertising-revenue-of-google/
44. Turner, K. (2009). 7 Surprising uses for the iPhone's camera, Macworld, May 7. http://

www.macworld.com/article/1140435/iPhone_photo_tips.html.
45. Verma, S. (2009). Operating System, Krishna Prakashan Media, Meerut, India.
46. Want, R. (2010). IPhone: Smarter than the average phone. *IEEE Pervasive Computing*, 9(3), 6-9.
47. Watson, H. J., Wixom, B. H., Hoffer, J. A., Anderson-Lehman, R., & Reynolds, A. M. (2006). Real-Time Business Intelligence: Best Practices at Continental Airlines. *Information Systems Management*, 23(1), 7-18.
48. Wikipedia (2016a). Systems Engineering. Current November 2016 http://en.wikipedia.org /wiki/Systems_engineering.
49. Wikipedia (2016b). Amazon.Com Current November 2016. http://en.wikipedia.org/wiki/Amazon.com.
50. Wikipedia (2016c). Current August 2016. Walmart, https://en.wikipedia.org/wiki/Walmart.
51. Wikipedia (2016d). Google Search Engine, http://en.wikipedia.org/wiki/Google_search_engine. Current November 2016.
52. Wikipedia (2016e). McDonnell Douglas DC-10, Current November 2016. http://en.Wikipedia.org/wiki/McDonnell_Douglas_DC-10
53. Wikipedia (2016f). DOS, https://en.wikipedia.org/wiki/DOS. Current November 2016.
54. Wikipedia (2016g). Information Management System, https://en.wikipedia.org/wiki/IBM_Information_Management_System. Current November 2016.
55. Wiseman, C. (1985). *Strategy and Computers*. Homewood, IL: Dow Jones-Irwin
56. Wohl, J. (2015) Walmart Adds iPhone Scan-and-Checkout Feature to 12 More Markets, March 20 Reuters. http://www.reuters.com/article/2013/03/20/us-walmart-checkout-expansion-idUSBRE92J0P020130320.
57. Yourdon, E. (1988). *Modern Structured Analysis*. Prentice Hall.

Appendix 1: Evaluation of Systems along Criteria

Criteria → System	Fulfills primary function well?	Has good Infrastructure?	Connectability?	Adaptable/Versatile ?	Reliable?	Additional Benefits/ (Problems)?
Amazon.com e-commerce system (Post '12, Wikipedia '16b)	Yes, the system is the only outlet for Amazon, #1 online retailer.	Yes! In 2000 Amazon revamped its infrastructure to include DBMS, ERP, mining and analysis. Ordering and fulfillment added in 2004.	Yes! Their new IT system allows easy connections with supplier's systems as well as their partners.	Very versatile. Initially started selling books then branched into music, toys, electronics, apparel and more..	Reliable. Web site failures are rare.	Yes. They branched into cloud services in 2002.
Wal-mart information system (Post '12, Gallaugher '12, Wikipedia '16c).	Yes, system enabled Walmart to be #1 retailer.	Focuses on using a centralized system with common platforms. Items RFID'd and scanned. sales transactions stored in a warehouse.	Yes. Systems integrated with Voice Order Filling and iPhone Apps.	Yes, easily adapted to several store formats.	Yes, apparently very robust.	Yes. Ability to spot high volume products through use of warehousing and RFID technologies.
Apple iphone (German '07, Want '10)	Yes, millions of users have used iPhone – many for critical tasks.	Built around a home screen and a graphical menu of available apps. Also provides 3G support.	Can provide connections through local WiFi networks, EDGE networks, or GSM Networks.	'Apple Apps' make the iPhone very versatile. Including: navigation, bar code scanning, dictation, credit card processing, invoicing	Yes, also dependent on battery.	Yes. It is being used as a payment processing terminal by some mobile vendors.

				and remote control functionality for home appliances.		Millions of other apps.
Google Search Engine (Barroso et al '03; Wikipedia '16d)	Yes. Extremely well. More than 3.5 billion searches/ day.	Yes. Has a giant index of websites and cached web pages that coupled with a page ranking algorithm, is able to respond quickly to user queries.	Search engine can be easily embedded into other web sites for local searches.	Very Adaptable. Initially started for word search, but expanded into synonyms, stock quotes etc. The same technology has been used to search blogs, journal articles, images, and RSS feeds.	Yes, due to sophisticated algorithms.	Yes. Triggers special features for some keywords, such as: unit, currency, and time conversions, weather, stock quotes, and discrete math functions.
DOS (McDugall '13, Paterson '15)	No. Does not have device drivers, needed for writing any application.	No. Single user single tasking operating system.	No. DOS has always been plagued with inconsistencies and incompatibilities between different software programs.	No. DOS was written for the Intel 8086 family.	No. System crashes were frequent.	(Yes), developing applications was a nightmare.
DC-10/MD-10 (anonymous '14, Wikipedia '16e, Boeing '14)	Yes, it transported cargo and passengers as well as other aircraft.	Yes and No. Basic infrastructure sound, but engine mount, cargo door and hydraulic systems designs have led to crashes.	NA. An aircraft is meant to be an independent system.	Versatile. Had 13 versions, served a wide range of uses from short distance to long distance, cargo transport and tanker applications	Initially poor but later reliable - 2.94 accidents per million departures compared to 2.85 for Boeing 747 and 2.29 for Airbus A300.	(Yes). In three separate incidents, hydraulic systems were ruptured by decompression and engine separation.
Healthcare system (Davis et al. '14, Jost '06, Npr '13).	No, poorly functioning when compared to other advanced countries.	Yes but basic infrastructure is slightly poor when compared to other countries.	Poor. It is very difficult to share patient information.	No. Adaptability is not evident since the system has not been adapted to alternative systems of medicine.	Unacceptable because humans are involved. Between 220,000-440,000 deaths annually.	(Yes)! System makes it difficult for patients to move from one physician to another. Patients do not know their obligation prior to treatment.
Hierarchical DBMS (Panneerselvam '04, Wikipedia '16g)	Yes and No. Good for storage but not retrieval.	No. Data is stored in a hierarchy with links between them – made for poor infrastructure	Yes, system could be connected to other systems for banking etc.	No. It was difficult to write queries for different views than what was designed.	Yes. Very reliable, used in the banking industry.	(Yes). Users had to be aware of database structure to write queries.

This page is intentionally left blank

On enforcing relational constraints in MatBase

Christian Mancas, Valentina Dorobantu

Ovidius University

ABSTRACT

MatBase is a very powerful database management system based not only on the relational data model, but also on the elementary mathematical, entity-relationship, and Datalog logic ones. This paper briefly introduces MatBase and the elementary mathematical data model, after which is focused on the five relational constraint types and their enforcement in relational database management systems, as well as in MatBase.

Keywords : relational data model, domain constraint, not null constraint, unique key constraint, referential integrity constraint, tuple constraint, elementary mathematical data model, MatBase.

Classification : H.2.6 H.2

Language : English

London Journals Press

LJP Copyright ID: 250628
Print ISSN: 2514-863X
Online ISSN: 2514-8648

London Journal of Research in Computer Science and Technology

Volume 17 | Issue 1 | Compilation 1.0

On Enforcing Relational Constraints in *MatBase*

Valentina Dorobantu[α] & Christian Mancas[σ]

I. ABSTRACT

***MatBase** is a very powerful database management system based not only on the relational data model, but also on the elementary mathematical, entity-relationship, and Datalog logic ones. This paper briefly introduces MatBase and the elementary mathematical data model, after which is focused on the five relational constraint types and their enforcement in relational database management systems, as well as in MatBase.*

Keywords: relational data model, domain constraint, not null constraint, unique key constraint, referential integrity constraint, tuple constraint, elementary mathematical data model, MatBase.

Author α σ: Mathematics and Computer Science Dept., Ovidius University, Constanta, Romania.

II. INTRODUCTION

This first section briefly presents *MatBase*, the Elementary Mathematical Data Model (EMDM), the five relational constraint types ((co-) domain/ range, not null, unique keys, referential integrity/ foreign keys, and tuple/check) and their enforcement in the Relational Database Management Systems (RDBMSes), as well as related and further work.

The second section is the core of the paper, introducing the ways in which *MatBase* is enforcing and extending the five relational constraint types. The paper ends with conclusions and references.

2.1 MatBase

MatBase [1-5] is a powerful prototype multi-model, multi-user, and multi-language knowledge and data (KD) base management system (KDBMS) that currently has two versions: one in MS Access 2015 and the other in MS *SQL Server 2015* and C#. *MatBase* provides four data models: the (Elementary) Mathematical ((E) MDM) [1, 6-11], the Relational (RDM) [10, 12, 13], the Entity- Relationship (E-RDM) [10, 13, 14], and the *Datalog* ¬ [11, 13] based Logic (DL¬DM) ones.

Its main and most powerful interface is the (E) MDM one (which includes the DL¬DM one), where users manage sets, functions, and constraints that *MatBase* is automatically translating into corresponding tables, columns, and constraints (but users may also manage Datalog ¬inference rules and programs). For the relational constraints (see section 2.3), *MatBase* mainly uses the corresponding ones provided by MS Access and SQL Server. For the non-relational ones, it automatically generates embedded and/or extended SQL code into forms automatically built upon the corresponding tables.

MatBase also provides an RDM interface, where, dually, users manage tables, columns, and (only relational) constraints and it automatically generates corresponding sets, functions, and constraints.

Finally, *MatBase* also provides an E-RDM interface, where users manage E-R diagrams (E-RDs) and it automatically generates corresponding sets, functions, and constraints, as well as corresponding tables, columns, and constraints. Dually, users may ask in both (E)MDM and RDM interfaces generation of E-RDs for any set and its related ones on *n* nodes

London Journal of Research in Computer Science and Technology

distance from it, *n* being a natural parameter, or the full db scheme.

2.2 The Elementary Mathematical Data Model

(E)MDM schemes are quadruples made out of a finite nonempty collection of sets *S* partially ordered by inclusion, a finite nonempty set of mappings *M* defined on and taking values from sets of *S*, a finite nonempty set of constraints *C*, and a finite set of Datalog¬ programs *P* associated to the sets of *S* and mappings of *M*. Conventional database (db) schemas are triples <*S*, *M*, *C* >); when *P* is non- empty, the corresponding db is a deterministic deductive one, so a knowledge base.

S is partitioned into the following four blocks: object, value, system, and computed sets. *Object* ones are partitioned into *entity* (i.e. atomic) and *relationship* (i.e. non-functional math relations immune to their domain permutations). *Value* ones are subsets of (programming) data types. *System sets* include at least the data types, the empty set, and a distinguished countable set NULLS of *null values*. *Computed sets* are obtained from all other types of sets by using semi-naïve sets, functions, and relations algebra operators (e.g. union, difference, intersection, etc.).

All mappings in *M* are defined either on object sets or on computed sets based only on object ones. *M* is partitioned into the following four blocks: attributes, structural functions, system, and computed mappings. *Attributes* are taking values from value sets, while *structural functions* from object ones, both possibly combined with NULLS. *System mappings* include canonical projections, injections, and unity mappings. *Computed mappings* are obtained from all other types of mappings by using semi-naïve sets, functions, and relations algebra operators (e.g. composition, (Cartesian) product, etc.).

C is partitioned into the following four blocks: set, dyadic relation, mapping, and object constraints. (E)MDM has a rich panoply of constraint types: currently, it has 56, out of which 30 fundamental and 26 derived ones (see everyday life examples for all of them in [10, 11]). The main reason behind its introduction is that neither RDM nor E-RDM constraint types are not at all enough to guarantee db instances plausibility. For example, even such a simple constraint as "the capital city of any country should belong to that country" is not expressible in either RDM or E-RDM, whereas in (E)MDM it is, either as *Country* ^{o} *CapitalCity reflexive or, equivalently, as Country* ^{o} *Capital City* = 1_{CITIES}.

2.3 The 5 Relational Constraint Types and Their Enforcement in RDBMSes

RDM provides five types of constraints (i.e. closed first order logic calculus (FOLC) with equality formulae) that are embedded in all Relational Database Management Systems (RDBMSes) for enforcing these business rules, namely: domain (range), not null (function total definition), keys (uniqueness), typed inclusion (foreign keys, referential integrity), and tuple (check).

Tables storing fundamental data should make heavy use of them. Those storing temporary data should not have constraints (except for debugging purposes), as their enforcement costs both disk space and, especially, processing time. In what follows we consider only fundamental data tables.

Domain constraints restrict for columns the corresponding data types to some plausible subset. For example, to a function *BirthDate* : *EMPLOYEES* → [1/1/1900, *SysDate*() – 18 years] corresponds a DATE column *BirthDate* to which you have to add the domain constraint "between '1/1/1900' and SysDate() – 18 years". Obviously, without it users might store even highly implausible data, as DATE starts, for example, in MS Access with 1/1/100, in MS *SQL* Server and IBM DB/2 with 1/1/1, in Oracle with 1/1/4712 BC, and they all end on 31/12/9999: why letting users (and it doesn't matter whether by mistake or on purpose, to test your db design skills) entering data on employees born some 2000 years ago or that will be born only some 7000 years from now?

Note that, mathematically, this is, in fact, a co-domain definition, while in RDBMSes, as the corresponding SQL syntax is "CHECK BirthDate between '1/1/1900' and SysDate() – 18 years", it is generally called a check constraint.

A fundamental best practice rule is that for any column of type numeric (including DATE, which is also stored numerically) you should add a corresponding domain constraint. String character columns might also need such constraints. For example, to column *Sex* of table *EMPLOYEES* you should add the constraint "CHECK Sex in ('F', 'M')".

Not null constraints are, in fact, a particularization of the *existence constraints:* given any two columns f and g of a same table T, such a constraint is denoted by $f \vdash g$ and has the meaning "whenever f is not null, g should be not null too". For example, in a library db, in table *ITEMS* you should enforce both *BorrowDate* $\vdash$ *Borrower* (whenever an item was borrowed, you should also know to whom) and *Borrower* $\vdash$ *BorrowDate* (whenever somebody borrowed an item, you should also know when).

In the particular case when f is void, according to FOLC, the meaning of such a constraint is that g should never accept null values and this is why they are called *not null constraints*. Unfortunately, no RDBMS, except for *MatBase*, is providing existence, but only not null constraints. In SQL, they are enforced by declaring the corresponding column as NOT NULL.

A fundamental best practice rule is that any table should have at least one column, except for the surrogate key one, not accepting nulls.

Beware of notational confusions: Oracle, for example, considers NOT NULL constraints as being of type CHECK too.

Mathematically, functions are totally defined, so, for example, the correct definition of *CapitalCity*, if knowing capitals of all countries should not be compulsory in a db, would be *CapitalCity* : *COUNTRIES* → *CITIES* ∪ NULLS. In fact, as in dbs the vast majority of columns accept nulls, (E)MDM uses the dual notation that never explicitly uses NULLS, considering totality an optional constraint. For example, *CapitalCity* : *COUNTRIES* → *CITIES* and *CountryName* : *COUNTRIES* → ASCII(255), *total* mean that *CapitalCity* accepts nulls, whereas *CountryName* does not.

Key constraints are rejecting any attempt to duplicate data on corresponding columns of a table. For example, as there may not be two countries with a same name in the world, *CountryName* should be declared as UNIQUE in table *COUNTRIES*; similarly, as there may not be two states of a same country having same names the pair (*StateName*, *Country*) should also be declared as UNIQUE in table *STATES*.

RDBMSes enforce key constraints with unique indexes. Unfortunately, besides arity limitations (e.g. in current versions of MS Access a key may contain at most 10 columns, 16 in MS SQL Server, 32 in Oracle, 64 in IBM DB/2), RDBMSes do not accept in keys columns of some data types (e.g. long texts, OLE etc.) and some of them (e.g. MS SQL Server) do not accept columns having more than one null value.

Mathematically, a key is either a one-to-one function or a minimally one-to-one function product. A not minimally one-to-one function product is called *superkey* both in RDM and (E)MDM. For example, (*Population*, *StateName*, *Country*) is a superkey in *STATES*. Unfortunately (as not only conceptually superkeys are not minimal, but they also waste unneeded both disk space and enforcement time), all RDBMSes, again except for *MatBase*, accept enforcing both keys and superkeys.

Tables may have a *primary key*, which is a key not accepting nulls. Most RDBMSes provide a surrogate key data type (called COUNTER, GENERATED, AUTONUMBER etc.), i.e. fixed point numeric, generally with auto-numbering,

having no other semantics but unique row identification.

Other fundamental best practice rules are:

- Every table should have a surrogate primary key, with auto-numbering whenever it is not also a foreign key (which is needed for tables corresponding to subsets, e.g. for *DRIVERS* ⊆ *EMPLOYEES*);
- Every table should have all keys existing in the corresponding actual subuniverse.

A *referential integrity constraint* restricts the values that may be taken by a column or set of columns to those taken by another column or set of columns, respectively. For example, column *Country* from table *STATES* should only take values from those stored by the surrogate primary key column *ID* from table *COUNTRIES*. Columns to which such constraints are associated are called *foreign keys*; for example, *Country* above is a foreign key.

The SQL syntax for such constraints, exemplified for *Country*, is "FOREIGN KEY (Country) REFERENCES COUNTRIES", where whenever the foreign key references the primary key the latter needs not be explicitly mentioned.
Mathematically, if f references g, then $Im(f) \subseteq Im(g)$, where the image of a function, denoted *Im*, is the set of the values it is taking.

Another fundamental best practice rule is that every foreign key references a surrogate primary key.

Finally, *tuple constraints* have no generally accepted definition; generalizing corresponding RDBMS implementations, they are constraints relating several columns of a same table, e.g., in *EMPLOYEES*, *BirthDate* ≤ *HireDate* ≤ *Passed Away Date* (written by notational abuse as formulas of a propositional calculus, but being in fact FOLC closed formulas with only one variable occurrence universally quantified; e.g. the one above is a shortcut for $(\forall x \in EMPLOYEES)$ $(BirthDate(x) \leq HireDate(x) \leq PassedAway\ Date(x))$).

2.4 Related and Further Work

Oracle constraint enforcement is presented in Chapter 6 of [15]. Corresponding data for the MS SQL Server can be found in [16]. For MS Access, [17] presents the SQL CONSTRAINT clause, while [18] the ADO API.

A planned improvement of *MatBase* is to allow definition of existence constraints for computed functions too, as well as modifying existence constraints.

Last, but not least, further work will be done for providing *MatBase* versions for Oracle, IBM DB2, *MySQL*, and *PostgreSQL* in a next step.

III. ENFORCING RELATIONAL CONSTRAINTS IN *MATBASE*

MatBase not only enforces all five relational constraint types, but it does so more elegantly and powerful than any other DBMS. As most of the systems, it enforces constraints, regardless of type, only if the current db instance satisfies them.

3.1 Domain constraints

Both *MatBase* versions establish data types based on the corresponding co-domains. For example, both NAT (the naturals) and INT (the integers) are implemented as integers (unsigned for NAT). Corresponding subtypes are chosen based on the *DomCnstr* (Domain Constraint) values, if any. For example, a co-domain NAT with *DomCnstr* = 2 (i.e. NAT(2)) is implemented as the subtype Byte, as it can only store naturals less than 100. When no such value is specified, the largest corresponding subtype is selected; for example, INT with a null value in *DomCnstr* is implemented in Access as Long Integer.

In its MS Access version, as there is no SQL CHECK clause, *MatBase* uses DAO, just like Access does when enforcing its Graphic User Interface (GUI) Validation Rule for columns. In its MS SQL Server version, *MatBase* uses the SQL CHECK clause.

In its GUI, namely in its *FUNCTIONS* form, you can specify domain constraints other than with *DomCnstr* by using either *MinValue* and/or *MaxValue* or user-defined value sets as codomains. Obviously, if, for example *MinValue* is *min* and *MaxValue* is *max*, it will enforce a BETWEEN *min* and *max* validation rule or CHECK constraint, respectively; if only *min* is not null whereas *max* is null a $\geq$ *min* one is enforced. You can define a value set by declaring it as a subset of a data type; for example, you can define *RAINBOW_COLORS* $\subseteq$ ASCII(6), which will create a corresponding table, and then fill it with data (e.g. 'red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet'). If the corresponding set is declared as static, then in its MS SQL Server version a corresponding CHECK IN (...) constraint is enforced, whereas in its MS Access one a combo-box filled with corresponding field values is declared; if not, a referential integrity constraint is enforced between the corresponding column and the surrogate primary key with auto-numbering of that value set.

3.2 Existence constraints

MatBase enforces both the particular NOT NULL constraints and the general existence ones. NOT NULL is enforced in both versions with the corresponding SQL clause. According to the math definition, the *FUNCTIONS* form contains a *Total* column for it: when checked, NOT NULL is enforced for the corresponding column. For object identifiers (to which table primary keys correspond), canonical Cartesian projections (the so-called *roles* of underlying sets in relationships, e.g. *Country* and *Neighbor* in *NEIGHBORS*), canonical surjections (e.g. *Represented By* : *PEOPLE* $\rightarrow$ *MPS*, where *MPS* $\subseteq$ *PEOPLE* is the set of the members of the Parliament) totality cannot be removed.

As the general existence constraints (EC) imply each two functions, *MatBase* GUI provides an *EXISTENCE CONSTRAINTS* form with three columns: the corresponding primary key (that is also a foreign key referencing the primary one of the *CONSTRAINTS* table), the left, and the right side functions. You can only add or remove ECs. When adding a new one, first of all, *MatBase* enforces the meta-constraints stating that both functions have to be defined on a same set, not being total (i.e. accepting nulls), and, for the moment, not being calculated. If they are met and the db instance satisfies it, then corresponding code (VBA or C#, respectively, with embedded SQL) is automatically generated into the standard update form associated to the common functions domain. When an EC is deleted, this code is automatically deleted too.

3.3 Uniqueness (key) constraints

The unique feature of *MatBase* when it comes to key constraints is its *Keys Discovery Assistant* (KDA) [19]. You can invoke this wizard set by set, for both relationship and entity type ones. For relationships, in a first step, only structural keys are considered, i.e. keys made out only of the corresponding canonical Cartesian projections. In a second step, just like for the entity type sets, all prime mappings defined on that set are considered. When it opens, the wizard presents you with all keys already declared for that set, if any, as well as with all other possible key candidates, if any. You can delete existing keys or/and declare new ones out of the existing candidates. After each such operation, the wizard re-computes the possible keys. Moreover, KDA does the following other tasks:

- It is never including superkeys among the candidates.
- It stops when the maximum possible number of keys is reached.
- It does not allow enforcement of keys whenever the current db instance violates them.
- Generates and apply corresponding ALTER TABLE DROP/ADD CONSTRAINT SQL DDL UNIQUE statements whenever possible (i.e. in MS Access, for example, when the key arity is maximum 10 and all involved column data types are accepted in keys, and there are at most 6 keys per table, whereas in MS SQL Server the maximum is 32, and, moreover,

none of the columns has more than one null value).

- For all other keys (i.e. of greater arity or/and containing not accepted column data types), in order to enforce them it automatically generates code (VBA or C#, respectively, with embedded SQL) in the corresponding set/ table standard update form.

3.4 Referential integrities (foreign key) constraints

For any set, *MatBase* adds a corresponding object identifier mapping (named *x*), i.e. a totally defined, one-to-one function defined on that set and taking integer values (for which, in the corresponding table, a surrogate primary key is added, with auto- numbering if the table does not correspond to a subset of another set). For any structural function, i.e. one defined on and taking values from object sets, it automatically adds a corresponding column in the table corresponding to the domain set and declares it as being a foreign key referencing the primary key of the table corresponding to the co-domain set (for auto-functions, obviously, this is the primary key of the same table). Enforcement/ dropping of referential integrities is done through generation and execution of corresponding SQL DDL CONSTRAINT clauses of type FOREIGN KEY.

Consequently, all foreign keys generated by *MatBase* are single-columned and integer type ones referencing primary keys. For example, for the function *State* : *CITIES* → *STATES*, *total* in table *CITIES* an integer column *State* not accepting nulls is added and declared as a foreign key referencing the primary one of table *STATES*.

3.5 Tuple (check) constraints

MatBase enforces/drops these constraints exactly like the RDBMSes on which is built upon, either by generating and executing corresponding ALTER TABLE ADD/DROP CONSTRAINT SQL DDL CHECK statements in its MS SQL Server version or by calling VBA + embedded SQL methods using ADO with corresponding parameters in its MS Access one (as there is no CHECK clause in Access' SQL).

For example, in its MS SQL Server version, for the constraint *StartDate* ≤ *EndDate* ≤ *StartDate* + 30 attached to a table *PROJECTS MatBase* generates and runs the SQL DDL statement ALTER TABLE PROJECTS ADD CONSTRAINT PROJECTS_C_1 CHECK EndDate BETWEEN StartDate AND StartDate + 30.

IV. CONCLUSION

MatBase enforces all five types of relational constraints in both its versions. Whenever this is possible, it does so by using the underlying MS engines (Access and SQL Server, respectively).
Moreover, *MatBase* has several unique features that no other DBMS has, out of which the main ones are the following:

- Also enforces general existence constraints, not only their particular NOT NULL case.
- Provides users with the facility to declare functions (and corresponding columns) as being nonprime (i.e. never able to take part in a unique key), so as to minimize the time needed to discover all existing keys.
- Provides a Keys Discovery Assistant, which guides users and enforces/drops keys and only keys (i.e. never superkeys) in discovering and enforcing all existing keys in the least time possible.
- Enforces keys that are not enforceable by using the underlying MS engines (i.e. in Access, for example, those of greater than accepted arity or/and containing columns not accepted in keys or/and all those that surpass the maximum of 6 keys per table) by automatic code generation.
- Automatically generates optimal primary keys (i.e. single-columned and integer type ones).
- Automatically generates optimal foreign keys (i.e. single-columned and integer type ones referencing primary keys).
- Enforces dozens of other constraint types, which are non-relational and exist only in the

Elementary Mathematical Data Model, through automatic code generation.

Further work will include allowing definition of existence constraints for computed functions too, as well as modifying existence constraints. Moreover, *MatBase* versions for Oracle, IBM DB2, MySQL, and PostgreSQL are planned too.

REFERENCES

1. Mancas, C. (1997) Conceptual data modeling. (in Romanian) Ph.D., Thesis: Politehnica University, Bucharest, Romania.
2. Mancas, C.; Dragomir S.; Crasovschi, L. (2003) On modeling First Order Predicate Calculus using the Elementary Mathematical Data Model in *MatBase* DBMS. In Proc. IASTED AI 2003 MIT Conf. on Applied Informatics, 1197-1202, Acta Press, Innsbruck, Austria.
3. Mancas, C.; Dragomir S. (2004) *MatBase* Datalog Subsystem Metacatalog Conceptual Design. In Proc. IASTED SEA 2004 MIT Conf. on Software Eng. and App., 34-41, Acta Press, Cambridge, MA.
4. Mancas, C.; Mancas, S. (2005) *MatBase* E-R Diagrams Subsystem Metacatalog Conceptual Design. In Proc. IASTED DBA 2005 Conf. on DB and App., 83-89, Acta Press, Innsbruck, Austria.
5. Mancas, C.; Mancas, S. (2006) *MatBase* Relational Import Subsystem. In Proc. IASTED DBA 2006 Conf. on DB and App., 123-128, Acta Press, Innsbruck, Austria.
6. Mancas, C. (1985) Introduction to a data model based on the elementary theory of sets, relations and functions (in Romanian). In Proc. of INFO IASI '85, 314-320, A.I.Cuza University, Iasi, Romania.
7. Mancas, C. (1990) A Deeper Insight into the Mathematical Data Model. Proc. 13th Intl. Seminar on DBMS, ISDBMS'90, 122-134, Mamaia, Romania.
8. Mancas, C. On Modeling Closed E-R Diagrams Using an Elementary Mathematical Data Model. Proc. 6th ADBIS 2002 Conf. on Adv. in DB and Inf. Syst., 165-174, Slovak Technology University Press, Bratislava, Slovakia, 2002.
9. Mancas, C. (2002) On Knowledge Representation Using an Elementary Mathematical Data Model. In Proc. IASTED IKS 2002 Conf. on Inf. and Knowledge Sharing, 206-211, Acta Press, St. Thomas, U.S. Virgin Islands, U.S.A.
10. Mancas, C. (2015) Conceptual Data Modeling and Database Design: A Completely Algorithmic Approach. Volume I: The Shortest Advisable Path. Waretown, NJ: Apple Academic Press / CRC Press / Francis & Taylor.
11. Mancas, C. (2017) Conceptual Data Modeling and Database Design: A Completely Algorithmic Approach. Volume II: Refinements for an Expert Path. Waretown, NJ: Apple Academic Press / CRC Press / Francis & Taylor (to be published).
12. Codd, E. F. (1970) A relational model for large shared data banks. CACM 13(6): 377-387.
13. Abiteboul, S.; Hull, R.; Vianu, V. (1995) Foundations of Databases; Addison-Wesley: Reading, MA.
14. Chen, P. P. (1976) The entity-relationship model: Toward a unified view of data. ACM TODS 1(1): 9-36.
15. Oracle Corp. (2005) Application Developer's Guide - Fundamentals. https://docs.oracle.com/cd/B19306_01/appdev.102/b14251.pdf
16. Microsoft Corp. (2016) SQL Server Constraints. https://technet.microsoft.com/en-us/library/ms1 89862(v=sql.105).aspx
17. Microsoft Corp. (2015) Access Constraint clause. https://msdn.microsoft.com/en-us/library/office/ff836971.aspx
18. Microsoft Corp. (2016) ADO API Reference. https://msdn.microsoft.com/en-us/library/ms678086(v=vs.85).aspx
19. Mancas, C. (2016) Algorithms for Database Keys Discovery Assistance. In Perspectives in Business Informatics Research, 322-338, LNCS 261, Springer International Publishing.

This page is intentionally left blank

London Journal Press Membership

For Authors, subscribers, Boards and organizations

London Journals Press membership is an elite community of scholars, researchers, scientists, professionals and institutions associated with all the major disciplines. London Journals Press memberships are for individuals, research institutions, and universities. Authors, subscribers, Editorial Board members, Advisory Board members, and organizations are all part of member network.

Read more and apply for membership here:
https://journalspress.com/journals/membership

For Authors

Author Membership provide access to scientific innovation, next generation tools, access to conferences/seminars /symposiums/webinars, networking opportunities, and privileged benefits.
Authors may submit research manuscript or paper without being an existing member of LJP. Once a non-member author submits a research paper he/she becomes a part of "Provisional Author Membership".

For Institutions

Society flourish when two institutions come together." Organizations, research institutes, and universities can join LJP Subscription membership or privileged "Fellow Membership" membership facilitating researchers to publish their work with us, become peer reviewers and join us on Advisory Board.

For Subscribers

Subscribe to distinguished STM (scientific, technical, and medical) publisher. Subscription membership is available for individuals universities and institutions (print & online). Subscribers can access journals from our libraries, published in different formats like Printed Hardcopy, Interactive PDFs, EPUBs, eBooks, indexable documents and the author managed dynamic live web page articles, LaTeX, PDFs etc.

GO GREEN AND HELP
SAVE THE ENVIRONMENT

JOURNAL AVAILABLE IN

PRINTED VERSION, INTERACTIVE PDFS, EPUBS, EBOOKS, INDEXABLE DOCUMENTS AND THE AUTHOR MANAGED DYNAMIC LIVE WEB PAGE ARTICLES, LATEX, PDFS, RESTRUCTURED TEXT, TEXTILE, HTML, DOCBOOK, MEDIAWIKI MARKUP, TWIKI MARKUP, OPML, EMACS ORG-MODE & OTHER

SCAN TO KNOW MORE

support@journalspress.com
www.journalspress.com

*THIS JOURNAL SUPPORT AUGMENTED REALITY APPS AND SOFTWARES