Covenant Journal of Informatics & Communication Technology

Vol.2, No. 2, December, 2014.

A Bi-annual Publication of the Departments of Computer Information Science, and Electrical & Information Engineering, Covenant University, Canaan Land, Km 10, Idiroko Road, Ota, Ogun State, Nigeria.

Website: http//Journal.covenantuniversity.edu.ng

ISSN - Print: 2354 – 3566
- Electronics: 2354 – 3507

Published by Covenant University Journals,
Covenant University, Canaanland, Km 10, Idiroko Road,
P.M.B. 1023, Ota, Ogun State, Nigeria.

Printed by Covenant University Press.

ARTICLES

Using Statistical Analysis of FLOSS Systems Complexity to Understand Software Inactivity

Francielly Grigorio[1],

Daniel Miranda de Brito[2],

Eudisley Anjos[3],

Mario Zenha-Rela[4]

[1]Centre for Informatics, Federal University of Paraíba, João Pessoa – PB, Brazil
[1]franciellygrigorio@gmail.com

[2]Centre for Informatics, Federal University of Paraíba, João Pessoa – PB, Brazil
[2]britmb@gmail.com

[3]CISUC, University of Coimbra, Coimbra, Portugal
[3]eudisley@ci.ufpb.br

[4]CISUC, University of Coimbra, Coimbra, Portugal
[4] mzrela@dei.uc.pt

Abstract: Understanding how systems evolves can reveal important pieces of information that can help open source stakeholders to identify what can be improved in the software system's internal organization. Once software complexity is one of the most important attributes to determine software maintainability, controlling its level in the system evolution process makes the software easier to maintain, reducing the maintainability costs. Otherwise, uncontrolled complexity makes the maintenance and enhancement process lengthy, more costly and sometimes it can contribute to the system abandonment. This work investigates the evolution of complexity in discontinued FLOSS projects, through statistical analysis with data obtained from analisis of SonarQube Software. SonarQube is an open-source software quality tool that analyzes the project's source code and give the developers a feedback about the internal status of what is being developed. After several analyses, the outcome showed interesting results. A substantial portion of inactive FLOSS projects do not seem to be able to keep up with the extra work required to control the systems complexity, presenting a different behaviour of the successful active FLOSS projects. Though, some inactive FLOSS projects do have a complexity evolution that resembles with the curves belonging to active projects.

Keywords: Software Complexity, FLOSS, software inactivity, open source success.

I. Introduction

Although Free/Libre Open Source Software (FLOSS) projects has increasingly gained visibility among programmers in these past years, it is still a challenge for these kind of projects to manage to reach success and high quality systems

(Michlmayr, Hunt, & Probert, 2005). The lack of a rigid structure more present in proprietary software, sometimes may prejudice the prosecution of the system. The amount of discontinued projects is undoubtedly higher compared to the number of active projects. To illustrate this scenario, consider the data extracted in November, 2012, from SourceForge.net, one of the most popular Free Software repositories. Out of over 174,000 hosted projects, 65% of them were classified as "dormant", (no development within the last two years) while 28% were classified as "active", and the 7% remainder were classified as "inactive" (its development was allegedly ceased by the lead developer) (Khondhu, Capiluppi, & Stol, 2013).

Predict model projects in order to correctly foresee success or failure is still a significant challenge in software engineering (Beaver, Cui, Charles, & Potok, 2009). The evolution of these projects may occurs very quickly depending on programming effort from collaborators.

According to Lehman's second law of software evolution, system's complexity increases unless work is done to maintain or reduce it. So, as the FLOSS systems grow in size and, consequently, features and capabilities, they will also grow in complexity if the level of required work is not reached (Capiluppi & Beecher, 2009). The deterioration of the integrity of the software, result of its continuous evolution, may manifest itself as growing complexity (Lehman, 1978). This phenomenon, called code decay in the literature, is likely to make it progressively more difficult to understand the inner workings of the software and, hence, to implement functional additions and changes (Eick, Graves, Karr, Marron, & Mockus, 2001). Thereby, it will be required increasing levels of work done to control complexity, in order to avoid system regression and to maintain their evolution (Capiluppi & Beecher, 2009).

Given the great commitment necessary to keep a working and evolving FLOSS project, it is not unusual to find projects in which the lead developer(s) lost the interest on it. This was proven by the numbers of "discontinued" and "dormant" projects hosted by SourceForge.net repository shown previously. Occasionally, there are contributors willing to continue the development, but its complexity makes it more practical for them to start a new project as a replacement to the original (Terceiro & Chavez, 2009). Sometimes even the lead developer(s) realize that the code became so complex that it is more cost-effective to rewrite large parts of the software, or even to rewrite it entirely from scratch, than investing time in enhancing existing code (Terceiro & Chavez, 2009).

During the initial research on the theme, we notice that some discontinued projects hosted by web-based repositories, even when declared inactive by its developers (The inactive tag of SourceForge.net only can be setted by the development team) still hold a high download rate per week. That means that projects that no longer have updates in their systems and no maintenance whatsoever still manage to be successful. For instance, we can mention the ffdshow project (Ivanov et al., n.d.), that currently counts with about eighteen thousands of weekly downloads. Below in Table 1, you can see some discontinued projects hosted by SourceForge.net repository that have a high download rate per week. If these projects continue being that popular, what is reason why they are discontinued then? To better understand this question, it is important to consider not only the interest of the public that utilizes the software, but also the internal operation of the system. If the system reaches a point where it is difficult to make necessary changes that contribute to the system's evolution, this can lead to its inactivation. In this case, the software development is ceased, but it is not necessarily discontinued. discontinued. The difference is that though the system cannot suffer further modifications, it is still running correctly so it can have a large number of page visits and high weekly downloads rate, as well as a wide range of users. Therefore, regardless external discriminators of success, what determines whether a project will continue to evolve or cease its development can be also related to the state of its internal structure. It is in the light of these questions, we propose to explore the source code complexity in order to acquire an overview about the internal behaviour of the system.

TABLE I: DISCONTINUED PROJECTS HOSTED ON SOURCEFORGE.NET WITH A HIGH WEEKLY DOWNLOAD RATE'

Project	**Weekly Downloads**
ffdshow	18,881
VirtualDubMod	8,461
Guliverkli2	4,763
butt (broadcast using this tool)	1,088
Python/XML	1,240

In this paper, we approach the relation between discontinued FLOSS and complexity metrics. We think that complexity is one of the

most important attributes to be evaluated attributes to be evaluated in way to ensure an acceptable level of system maintenance. A project with an uncontrolled complexity can become difficult to maintain, and it can contribute to its abandonment. We think that understanding the complexity evolution for these software and the relation with its failure, it can guide us on how to avoid similar failures in software that are still under development. This paper is organised as follows: in Section II we show significant related work about FLOSS projects in order to situate the reader in the subject and relevance of this work. In Section III we state the research strategy containing the methodology we adopted to guide our work. In Section IV we presented the theoretical grounding to our statistical analysis. The results are presented in Section V, in Section VI we discuss future work, and we finalize the paper with the conclusions in Section VI.

II. Related Work

The categorization of OSS projects has been the target of an extensive number of studies, such in (Capiluppi, Lago, & Morisio, 2003), (English & Schweik, 2007), where is established that a considerable number of OSS projects tend to be discontinued, suffering the "abandonment tragedy" (English & Schweik, 2007).

A study conducted by (Beecher et al., n.d.), investigates whether the inclusion of a specific project in the same forge and distribution of a successful FLOSS project has an influence on its evolutionary characteristics. For the analysis, they sampled 50 projects from both forges: Debian and Source-Forge, and studied their evolution. They concluded that Debian projects do indeed show different characteristics than projects from SourceForge. Debian projects were shown to have a longer period of evolution, were larger in size, attracted more developers and experienced greater activity than SourceForge projects. Their second research question was based on the Debian sample only and assessed the presence of two phases of evolution, i.e. before and after the inclusion into the Debian forge. However, their results were ineficient to conclude that there was a statistically significant difference before and after the inclusion.

Using a sample set of 83 projects hosted on Source-Forge.net that had at least 7 developers that have ever contributed code and had at least 100 bugs reported, (Wang, 2005) concluded that the success of an OSS project can be predicted by just considering its first 9 month development data with his K-Means clustering predictor at relatively high confidence. k-means clustering is a machine learning algorithm used to cluster observations into groups of related observations without any

prior knowledge of those relationships.

Contrary to the idea of most existing studies on the maintainability of OSS projects, which shows that evolving software tends to decrease its quality and maintainability (Bakota et al., 2012), through our research of related work we found that this is not necessarily true, as stated in (Khondhu et al., 2013).

In their paper, (Stamelos, Angelis, Oikonomou, & Bleris, 2002) show empirical results on the relationship between the size of application components and the delivered quality measured through user satisfaction. Quality characteristics of 100 applications written in C for GNU/Linux were compared to industrial standards. They limited their analysis to the component level – a component is any C function in the program. They have determined that, up to a certain extent, the average component size of an application is negatively related to the user satisfaction for this application. Their findings relating structural quality to user satisfaction reflects the connection between internal structure and the application acceptability by the user, what may contribute to determine whether an OSS project will become successful or not. In (Schweik, English, Paienjton, & Haire, 2010), the authors investigated what factors lead to success or abandonment of open source software (OSS) projects. As mentioned in their paper, the most important external discriminators of success and abandonment in the growth stage of development process are the quantity of page visits and the download rate. However, it is not clear which factors in the source code structure affects these number.

III. Research Design

A. Research Strategy

Our research strategy is based on the Goal-Question- Metric (GQM) approach (Basili, Caldiera, & Rombach, 1994). According to this strategy, we must first specify our goals, then we must delineate those goals to the data that are designated to define those goals operationally, and finally provide a underlying structure for interpreting the data with respect to the stated goals. The GQM approach is a mechanism for defining and interpreting operational and measurable software. It can be used in isolation or, better, within a context of a more general approach to software quality improvement. The following are the objectives and questions that make up the strategy in this paper.

– **Goal**. The overall goal of this paper consists in study the relation between source code complexity metrics and the success of FLOSS projects, as well as investigate how further this relation can affect the system's survival. Once it is clear that we cannot directly point out the cause of the software's

abandonment, we propose to explore the internal factors that may contribute to the project failure.

– **Questions**. This paper addresses the following research question:

1) How the internal properties of the software influence the abandonment of a project?

2) In which point of development process discontinued FLOSS projects tend to fail?

– **Metrics and Definitions**. In order to analyse the OSS, we used the following metrics and definitions:

– **Lines of Code (LoC)**: Number of physical lines that contain at least one character which is neither a whitespace or a tabulation or part of a comment.

– **Complexity:** It is the cyclomatic complexity, also known as McCabe's complexity metric. Whenever the control flow of a function splits, the complexity counter gets incremented by one. Each function has a minimum complexity of 1.

– **Complexity/class**: Average complexity by class.

– **Complexity/file:** Average complexity by file.

– **Complexity/function:** Average complexity by function.

– **Normalized Complexity:** Outcome value of dividing the total complexity of each system version by its size (source code number of lines) of that same version.

B. Data Collection and Analysis

In this subsection we specify how the data collection and the analysis of the data obtained was performed. *SonarQube Software*: Firstly, aiming to pick a source code quality measurement tool that matched with our needs to fulfil the goals proposed at the beginning of this study, we have chosen to use the SonarQube software. SonarQube is an open source software that aims to be the source code's quality management platform, allowing control over a large number of software metrics by the development team. The software main goal is to identify how the code is evolving, as well the behaviour of the multiple quality measures, while pointing possible software bugs.

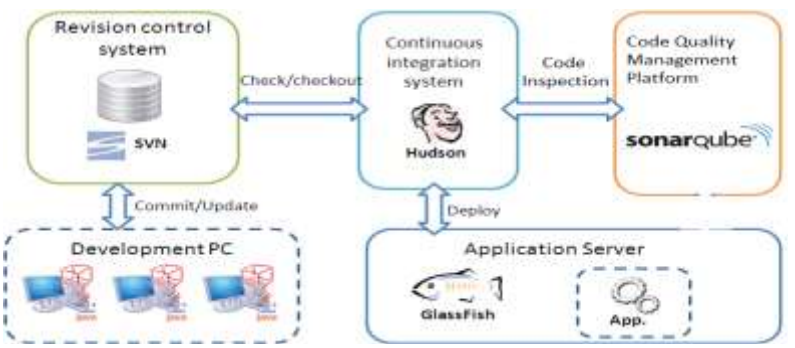

Figure. 1: Integration of SonarQube: How the software works.

When a SonarQube analysis is performed, the outcome is generated automatically through a complete inspection on the code base, after which the results are displayed through the web interface displayed in graphics and dashboards.

Figure. 2: SonarQube Web Interface.

Bellow, we detail the steps taken to perform this empirical study.

1) Sampling of projects. In way to find the software to fit in our strategy, we performed a research through the SourceForge.net repository. The results were filtered to show projects only written in Java, so they were ordered using two tags: inactive and rating. We chose the Java language because it is a widely used object-oriented programming language. Moreover, the rating was important to check how relevant the software among the discontinued ones was. Once that projects had a high value of rating, most of them had a high download rate per week. Then, we manually selected ten projects with more than five released versions to proceed with analysis. We checked the last update date to include only projects that were inactive for more than two years. Afterwards, we selected three successful and active FLOSS projects. They were used to compare with the unsuccessful ones. Two out of them, are well-known systems (Tomcat and Hibernate), and the third one, JabRef, is a smaller software, but still very popular. We chose them based on the availability and good organization of the source code (what simplifies the analysis).

2) Extracting data. The source code of the selected projects was manually extracted from the respective projects' web-pages on SourceForge.net repository. All released versions available were downloaded.

3) Analysing the source code. As stated before, we used the SonarQube tool to execute the analysis. A Python script was written

to automatize the analysis via terminal. Once the analysis is performed, the results are available in the web interface, and count with the extraction of almost twenty metrics analysed. Within the range of metrics generated by the analysis, we adopted six metrics to proceed with our research. The metrics we chose to proceed within this work are presented in the previous subsection.

4) Evaluating the results. The final step is the evaluation process. The results are obtained through the execution of SonarQube and data analysis.

The following table (Table 2) shows information from the ten discontinued FLOSS projects analysed. As we can observe, all of the samples has at least two years of inactivity, what assures that the projects are discontinued in fact. They also present a significant number of downloads per project, which shows on the other hand that all of them had some relevance in the Open Source Software scenario.

TABLE II: DATA OF SELECTED DISCONTINUED PROJECTS.

Project ID	**First Version** *(dd-mm-yyyy)*	**Last Version** *(dd-mm-yyy)*	**Released versions**	**Number of downloads**
Saxpath	24-01-2001	26-04-2002	7	21,759
Jaxen	25-07-2001	26-04-2002	10	43,682
Jo1	18-03-2002	07-07-2005	9	23,502
IdeaVIM	17-04-2003	30-03-2010	53	9,568
RemoteTea	14-08-2003	02-01-2008	5	15,474
JRFD	20-12-2003	09-02-2011	26	18,172
G4J	16-09-2004	19-09-2005	11	45,158
Gilead	10-11-2008	22-05-2010	7	28,791
LslPlus	28-07-2008	06-03-2009	14	20,220
MorarChat	21-12-2009	14-01-2012	15	19,535

IV. Statistical Analysis

A. Correlation

In order to determine how the complexity metrics are correlated, we started to seek for some correlation coefficient that could measure this connexion. Correlation coefficient is a measure of association between two variables, and it ranges between 1 and 1. If the two variables are in perfect linear relationship, the correlation coefficient will be either 1 (total positive correlation) or 1 (total negative correlation). The correlation coefficient is 0 if there is no linear relationship between the variables. Two different types of correlation

coefficients are in use. One is called the Pearson product-moment correlation coefficient, and the other is called the Spearman rank correlation coefficient, which is based on the rank relationship between variables ("Encyclopaedia of Measurement and Statistics", 2007).

The Pearson product-moment correlation coefficient (sometimes referred to as the **PPMCC** or **PCC** or **Pearson's r**) is more widely used in measuring the association between two variables. Given paired measurements (X_1; Y_1); (X_2; Y_2); (X_n; Y_n), the Pearson product-moment correlation coefficient is a measure of association given by

$$r_P = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

Where $\overline{X}$ and $\overline{Y}$ are the simple mean of X_1; X_2; ... X_n and Y_1; Y_2; ... Y_n, respectively.

Bellow you can find examples of scatter diagrams with different values of Pearson's correlation coefficient.

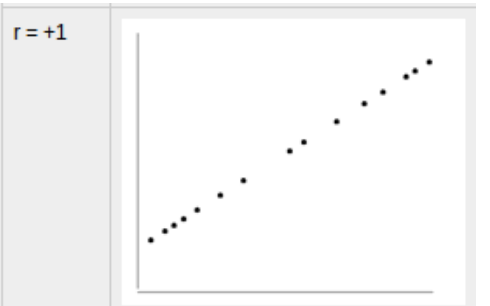

Fig. 3: Data lie on a perfect straight line with a negative slope

Fig. 4: No linear relationship between the variables.

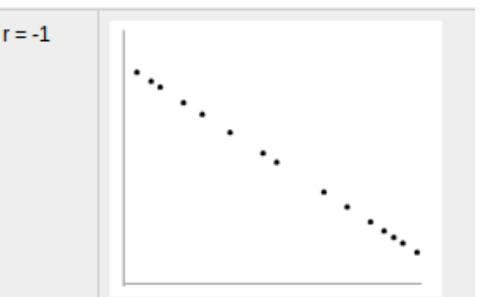

Fig. 5: Data lie on a perfect straight line with a positive slope.

B. R Programming To Calculate Correlation

R is an open-source (GPL) statistical environment, started by Robert Gentleman and Ross Ihaka (hence the name, R) of the Statistics Department of the University of Auckland in 1995. R is a powerful statistical program but it is first and

foremost a programming language. R provides a wide variety of statistical and graphical techniques, including linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, and others. R is easily extensible through functions and extensions, and, besides, routines have been written for R by people all over the world and mad freely available from the R project website as "packages".

R can perform correlation with the cor() function. Built-in to the base distribution of the program are three routines: for Pearson, Kendal and Spearman Rank correlations. The detailed steps undertaken are the following:

1) The first step is to arrange your data in a .CSV file. Use a column for each variable and give it a meaningful name. Don't forget that variable names in R can contain letters and numbers but the only punctuation allowed is a period. In our case, we arranged the data in a readable formatation, so we could copy the data into a .CSV file.

2) The second step is to read your data file into memory and give it a sensible name.

3) The next step is to attach your data set so that the individual variables are read into memory.

4) Finally, to get the correlation coefficient you type: cor(var1, var2, method = "method")

The default method is "pearson" so you may omit this if that is what you want. If you type "kendall" or "spearman" then you will get the appropriate correlation coefficient.

V. Results

Subsection IV.A shows the evaluation of the outcome obtained by performed analyse detailed in Subsection III.B. As described in Section III, we defined our goal that is addressed by the research questions. Subsections IV.B discuss questions 1 and 2. Subsection IV.C explain the exception in the results pattern.

A. Evaluating analysis outcome

Given the results of the analysis made with SonarQube, we began to interpret some of the data obtained. The data collected allows us to affirm that the projects are consistently growing since their first release. Delimiting the number of released versions served to not select small projects that are locally developed for a certain public. Thus, the selected projects were being actively developed and receiving new features as an effect of new user requirements.

In order to make a more complete analysis of the systems' metrics and compare them to each other, we made some normalizations. Thereby, we calculated the complexity ratio by dividing the total complexity of each system version by its size (source code number of lines) of that same version. We called it *"normalized complexity"*.

As we can observe through the statistics, shown by Fig. 1, 3, 5, 6, 7 and 9, six out of ten projects analysed have a resembling behaviour. These six projects (Jaxen, IdeaVIM, JRFD, G4J, Gilead and MorarChat), show a gradual (Fig. 1, 3, 6 and 7) or uncontrolled (Fig. 5 and 9) rise in their normalized complexity, which tends to remain for the upcoming versions. Comparing them with the graphics that shows the normalized complexity of active applications (Fig. 11), we can point out the differences. While the active projects has a more controlled complexity, presenting almost a constant normalized complexity, with small decreases on its value, the discontinued projects do not seem to be able to control the system's complexity, since its complexity is gradually increasing or sometimes it is even widely varying. In the other hand, Fig. 2, 4, 8 and 10 shows that the normalized complexity metric is kept almost constant for these projects. This demonstrate that these inactive projects (Jo!, RemoteTea, Saxpath and LslPlus) have a very similar behaviour to the active ones.

B. Addressing paper questions

Question 1. *"How the internal properties of the software influence the abandonment of a project?"*

The complexity data reveals interesting issues. As verified in the previous subsection, on six out of ten inactive projects analysed the complexity grows as the time passes, according as the discontinued projects evolve. Also, the curve drawn by the complexity sometimes may shows a lot of variations. Therefore, the increasing complexity of these projects enables us to assert that these projects can not completely control the complexity as faster as it grows. The level of work needed to keep the control over the complexity seems not to be reached by the developers, and this could also contribute to the abandonment of these projects.

In order to further investigate the relationship between the complexity of a project and its abandonment, we calculate de standard deviation of the complexity metrics. The standard deviation measures how close the set of data is to the mean value of the data set. If the data set have a high standard deviation then the values are spread out very much. If the data set have a small standard deviation then the data points are very close to the mean. The following graphs (figure 12 and 13) shows that discontinued projects have a larger standard deviation than the active ones, when it comes to its normalized complexity, indicating that active projects could possibly have a more effective control over its complexity. However, it is mandatory to notice that not all of the inactive projects presents the same behaviour. Fig. 12 ratifies what we observed previously about

similarities etween inactive and active projects. According to Fig. 12, Jo!, RemoteTea, Saxpath and LslPlus projects, shows indeed a low standard deviation, indicating that these projects kept the normalized complexity under control.

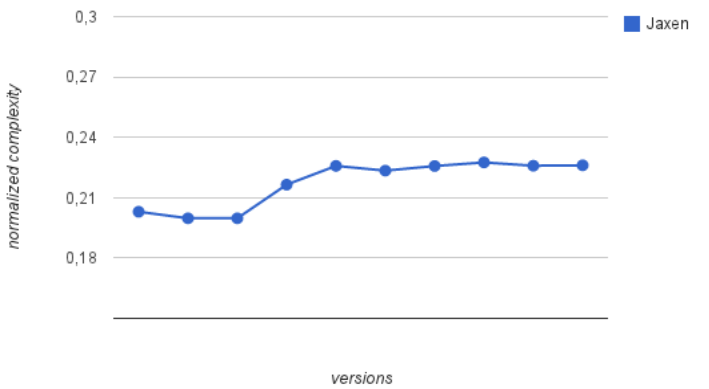

Fig. 6: Normalized Complexity of each version of Jaxen Application.

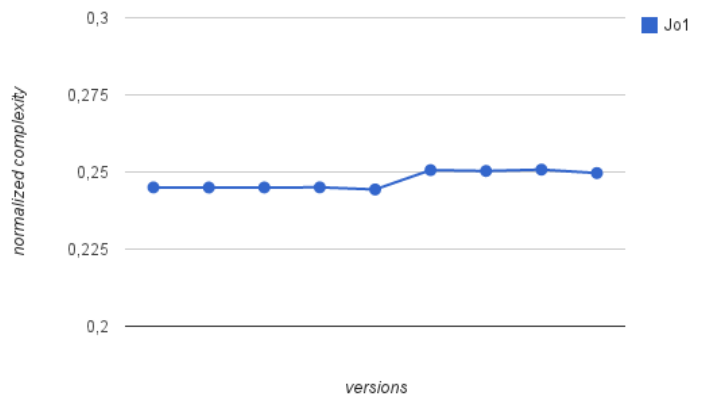

Fig. 7: Normalized Complexity of each version of Jo! Application

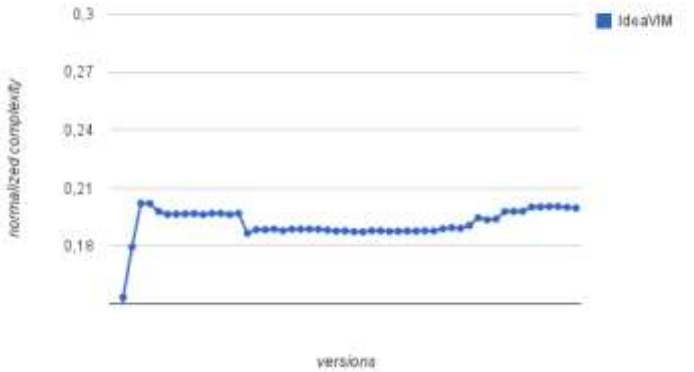

Fig. 8: Normalized Complexity of each version of Jaxen Application.

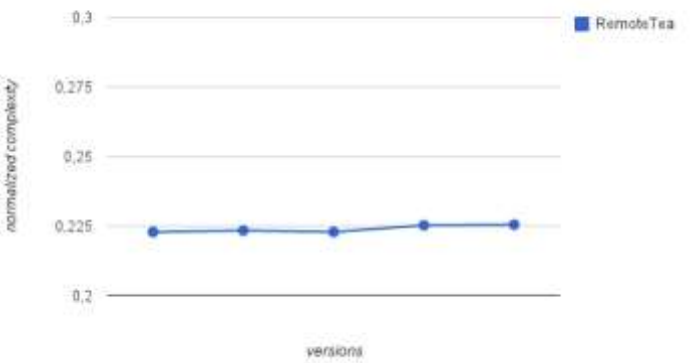

Fig. 9: Normalized Complexity of each version of Jaxen Application.

Moreover, observing the graphs we also identified that the complexity/function metric has a higher impact in the systems' evolution among the other complexity related metrics also analysed. We calculate the standard deviation as well as the variance of this metric. The variance measures how far a set of numbers is spread out. A variance of zero indicates that all the values are identical. Therefore, a small variance means that the variable has its numbers not so spread out, thus the similar values indicates constancy in the metric. We found out that both standard deviation and variance is larger for discontinued projects as well (Fig. 14, 15, 16 and 17), what lead us to conclude that in some projects the complexity may have a connection with its failure, especially when we focus in its complexity by function.

Question 2. *"In which point of development process discontinued FLOSS projects tend to fail?"*

As seen in Section I, there are a lot of projects hosted by web-based repositories that became so complex that it is more cost-effective to rewrite large parts of its source code. When an application reaches this apex of complexity, we can declare it a failed project. Therefore, this "failure point" – considering only the structural motivations – it is when the complexity makes the system no longer maintainable and then the project becomes discontinued.

However, as we can notice from the charts, the structural elements of software do not determine precisely when a project must be discontinued. Taking in account Fig. 5, which shows the behaviour of the normalized complexity of JRFD project, we can see that in the sixth released version of the software occurs a peak on the value of its normalized complexity, but this factor is clearly not sufficient to immediately cease its development. That lead us to conclude that, though the complexity may have a contribution on the discontinuation of a project, the point of development process that a project tend to fail is a complex question that involves way more external aspects rather than a limited subset of factors dictated by internal properties.

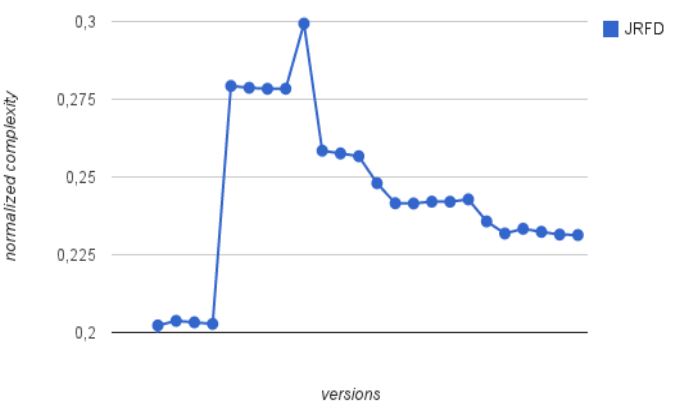

Fig. 10: Normalized Complexity of each version of Jaxen Application.

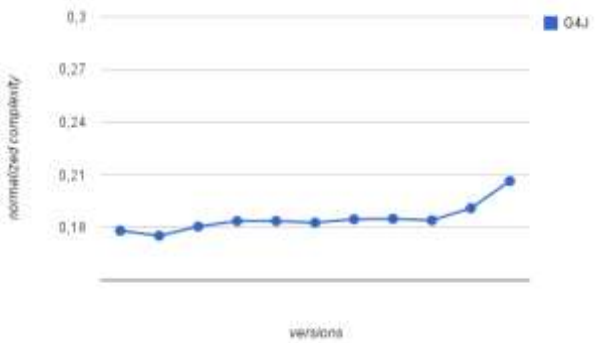

Fig. 11: Normalized Complexity of each version of G4J Application

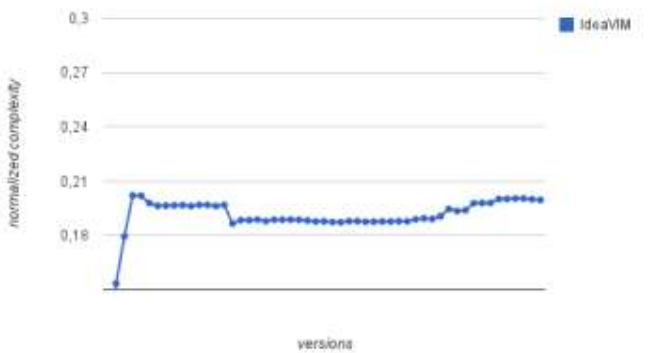

Fig. 12: Normalized Complexity of each version of IdeaVIM Application.

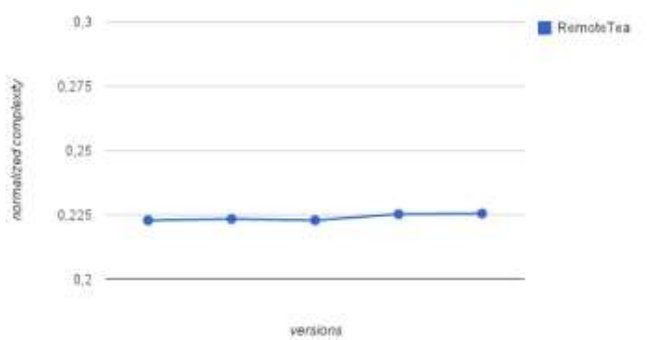

Fig. 13: Normalized Complexity of each version of RemoteTea Application

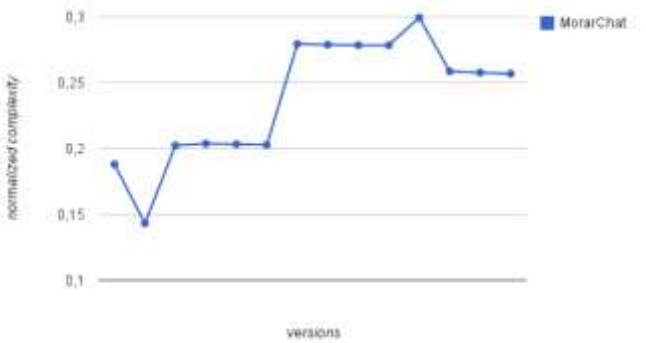

Fig. 14: Normalized Complexity of each version of MorarChat Application.

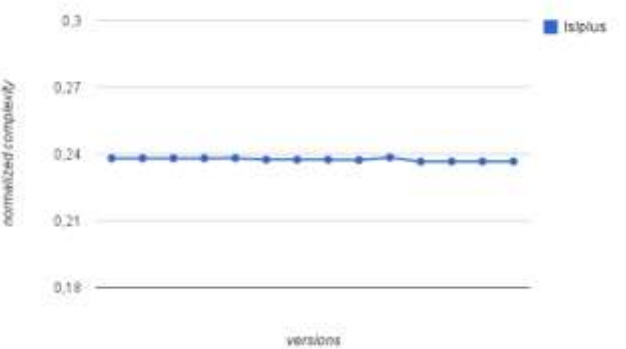

Fig. 15: Normalized Complexity of each version of LslPlus Application

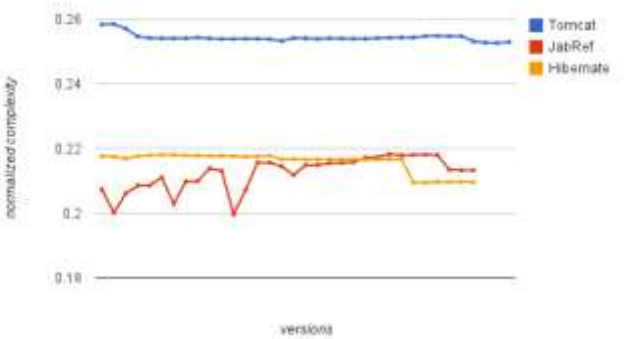

Fig. 16: Released versions Normalized Complexity of Actives FLOSS projects: Tomcat, JabRef and Hibernate Applications.

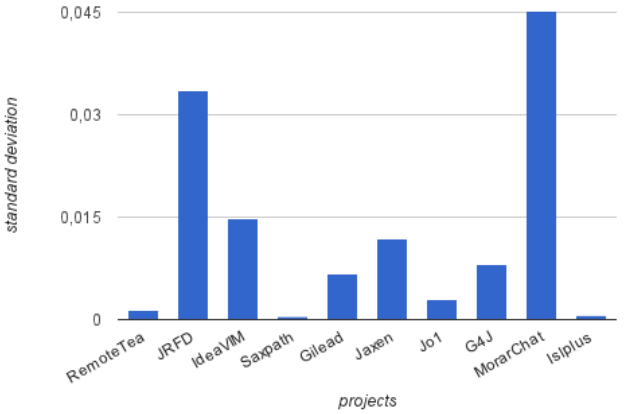

Fig. 17: Normalized Complexity standard deviation of discontinued FLOSS projects.

C. Correlation between complexity metrics

As we can see from the figures bellow, which show the Pearson's coefficient of the complexity metrics from the projects analysed, we can notice that the projects present different correlations for the same pairs of metrics analysed. For instance, the correlation between the LoC x Complexity by function metric is strongly positive for Jaxen application, but it is strongly negative for RemoteTea application and even non existent for Jo! application. Therefore, for each project, its class, function or file characteristics, has a stronger correlation with the other metrics, indicating that this particular metric has a greater impact over the characteristics of the project as a whole. What could determine which of them – the class, the function or the file – exerts this role, is the definition of its architecture and its definition of module. Once there is not a formal definition of a module, and what a module is in the context of a project, it largely depends on the nature and design of the project. So, this distribution of correlation may has to do of how the development team see the set of modules that compose the source code, and how they implement the additions and modifications in this software.

In order to have a clearer view of what metrics have deeper relation with the Complexity metric, we construct the table below. Take x for positive correlation with th Complexity metric and □x for negative correlation.

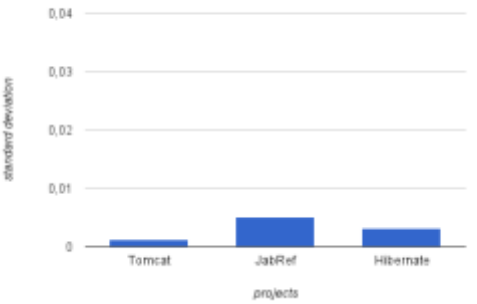

Fig. 18: Normalized Complexity standard deviation of actives FLOSS projects.

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.999006029963354	0.986603541621991	0.98867694169473	0.973897426170521	0.980210998920792
Complexity	0.999006029963354	1	0.9867273189408	0.989081245047754	0.970774040990912	0.987498230023685
Complexity..class	0.986603541621991	0.9867273189408	1	0.999569171226686	0.981268120625704	0.977261995604649
Complexity..file	0.98867694169473	0.989081245047754	0.999569171226686	1	0.977411043669536	0.979655692532217
Complexity..function	0.973897426170521	0.970774040990912	0.981268120625704	0.977411043669536	1	0.954061987052325
Normalized.Complexity	0.980210998920792	0.987498230023685	0.977261995604649	0.979655692532217	0.954061987052325	1

Fig. 19: Pearson's correlation coefficient of Jaxen Application.

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.766134307256283	0.177297794762568	0.0693159212090763	-0.0941816719658796	0.163476187232098
Complexity	0.766134307256283	1	0.424583504073707	0.524808421619532	0.217620976644509	0.75923263309458
Complexity..class	0.177297794762568	0.424583504073707	1	0.918313476142838	0.885846876816296	0.471452296654335
Complexity..file	0.0693159212090763	0.524808421619532	0.918313476142838	1	0.875052197236741	0.733874011600843
Complexity..function	-0.0941816719658796	0.217620976644509	0.885846876816296	0.875052197236741	1	0.431325834168898
Normalized.Complexity	0.163476187232098	0.75923263309458	0.471452296654335	0.733874011600843	0.431325834168898	1

Fig. 20: Pearson's correlation coefficient of Jaxen Application.

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.991552497832614	0.870785398808207	0.804206794314933	0.87251118653575	0.583526523163386
Complexity	0.991552497832614	1	0.885938946173016	0.850283521401091	0.90053054383745	0.669576857131758
Complexity..class	0.870785398808207	0.885938946173016	1	0.700116878157055	0.98641375087783	0.808477260408668
Complexity..file	0.804206794314933	0.850283521401091	0.700116878157055	1	0.75370140041611	0.674209964318504
Complexity..function	0.87251118653575	0.90053054383745	0.98641375087783	0.75370140041611	1	0.849232185411292
Normalized.Complexity	0.583526523163386	0.669576857131758	0.808477260408668	0.674209964318504	0.849232185411292	1

Fig. 21: Pearson's correlation coefficient of Jaxen Application.

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.975500729481666	-0.125345103971592	-0.378584076848076	-0.819491096420301	0.8601710997300076
Complexity	0.975500729481666	1	0.0953544909578434	-0.170401105944017	-0.920041980323021	0.951296410006756
Complexity..class	-0.125345103971592	0.0953544909578434	1	0.945610857689302	-0.444444444444441	0.396737894477829
Complexity..file	-0.378584076848076	-0.170401105944017	0.945610857689302	1	-0.218217890235992	0.135252427075564
Complexity..function	-0.819491096420301	-0.920041980323021	-0.444444444444441	-0.218217890235992	1	-0.984384621000937
Normalized.Complexity	0.8601710997300076	0.951296410006756	0.396737894477829	0.135252427075564	-0.984384621000937	1

Fig. 22: Pearson's correlation coefficient of Jaxen Application

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.997730270368385	-0.143313878931092	-0.136657866768724	-0.132237632739683	0.165207926539411
Complexity	0.997730270368385	1	-0.111909784254504	-0.102944278061979	-0.0880005850247567	0.222351826322608
Complexity..class	-0.143313878931092	-0.111909784254504	1	0.99878956802629	0.956454565926949	0.722134572827041
Complexity..file	-0.136657866768724	-0.102944278061979	0.99878956802629	1	0.967639877948548	0.750588228071347
Complexity..function	-0.132237632739683	-0.0880005850247567	0.956454565926949	0.967639877948548	1	0.855936750555237
Normalized.Complexity	0.165207926539411	0.222351826322608	0.722134572827041	0.750588228071347	0.855936750555237	1

Fig. 23 Pearson's correlation coefficient of Jaxen Application

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.998045818083132	-0.288303836981241	0.360852968853479	-0.914001216819736	0.968453360032262
Complexity	0.998045818083132	1	-0.343106351730618	0.307434489589209	-0.927465035695999	0.976333029086663
Complexity..class	-0.288303836981241	-0.343106351730618	1	0.785713882310894	0.531182680247434	-0.377421937675769
Complexity..file	0.360852968853479	0.307434489589209	0.785713882310894	1	-0.0742111060167819	0.269429766774059
Complexity..function	-0.914001216819736	-0.927465035695999	0.531182680247434	-0.0742111060167819	1	-0.943746108714496
Normalized.Complexity	0.968453360032262	0.976333029086663	-0.377421937675769	0.269429766774059	-0.943746108714496	1

Fig. 24 Pearson's correlation coefficient of Jaxen Application

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.997600310031068	0.138675584143178	0.0630068034542528	-0.247715207146147	0.708192109531482
Complexity	0.997600310031068	1	0.204874487012801	0.12936533739665	-0.181325220488548	0.754821654837729
Complexity..class	0.138675584143178	0.204874487012801	1	0.996082346619029	0.916222657200562	0.782430591426965
Complexity..file	0.0630088034542528	0.12936533739665	0.996082346619029	1	0.939617470104449	0.728813205583102
Complexity..function	-0.247715207146147	-0.181325220488548	0.916222657200562	0.939617470104449	1	0.48874383528207
Normalized.Complexity	0.708192109531482	0.754821654837729	0.782430591426965	0.728813205583102	0.48874383528207	1

Fig. 25 Pearson's correlation coefficient of Jaxen Application

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.991491510925087	0.993744734397134	0.98970484845393	0.565694915572679	0.458470919237787
Complexity	0.991491510925087	1	0.999037708220867	0.99955846108025	0.627863356322372	0.570254082990544
Complexity..class	0.993744734397134	0.999037708220867	1	0.998066563582892	0.59677820309685	0.54949727545893
Complexity..file	0.98970484845393	0.99955846108025	0.998066563582892	1	0.645497224367903	0.578457065711865
Complexity..function	0.565694915572679	0.627863356322372	0.59677820309685	0.645497224367903	1	0.715513039150639
Normalized.Complexity	0.458470919237787	0.570254082990544	0.54949727545893	0.578457065711865	0.715513039150639	1

Fig. 26 Pearson's correlation coefficient of Jaxen Application

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.975957353576485	0.636304571645248	0.772285647134311	0.417704969234509	-0.845415491848772
Complexity	0.975957353576485	1	0.784568793625485	0.890707340481633	0.546097499993221	-0.708956588937015
Complexity..class	0.636304571645248	0.784568793625485	1	0.977973486519339	0.82069109450425	-0.133219298221165
Complexity..file	0.772285647134311	0.890707340481633	0.977973486519339	1	0.755768737454067	-0.316554442587822
Complexity..function	0.417704969234509	0.546097499993221	0.82069109450425	0.755768737454067	1	0.00290071697510873
Normalized.Complexity	-0.845415491848772	-0.708956588937015	-0.133219298221165	-0.316554442587822	0.00290071697510873	1

Fig. 27: Pearson's correlation coefficient of MorarChat Application.

	Lines.of.code	Complexity	Complexity..class	Complexity..file	Complexity..function	Normalized.Complexity
Lines.of.code	1	0.995672283790243	0.93107388868554	-0.900370601353054	0.436178121615426	-0.839371517613618
Complexity	0.995672283790243	1	0.958734138022014	-0.887293630740858	0.432752051530142	-0.785232405437668
Complexity..class	0.93107388868554	0.958734138022014	1	-0.802562436124647	0.372104203767625	-0.596616740053613
Complexity..file	-0.900370601353054	-0.887293630740858	-0.802562436124647	1	-0.407232076729046	0.805443331384468
Complexity..function	0.436178121615426	0.432752051530142	0.372104203767625	-0.407232076729046	1	-0.374015295514953
Normalized.Complexity	-0.839371517613618	-0.785232405437668	-0.596616740053613	0.805443331384468	-0.374015295514953	1

Fig. 28: Pearson's correlation coefficient of LslPlus Application.

D. Detailed Analysis of Behaviours

Though we collected a small sample of projects, the outcome provided was very diversified, once the projects presented a very divergent behaviour from each other. The main difference is that the output of four out of the ten projects analysed (Jo!, RemoteTea, Saxpath and LslPlus projects) hold constant values for the studied complexity metrics. Thereby, we decided to further investigate these detached projects in order to verify what could have lead them to this result. Our aim was to answer the question: If the project has a

controlled complexity why did it become discontinued?

Accessing Saxpath webpage on SourceForge.net we found out that the project was not discontinued. In fact, it had the inactive tag because it had been incorporated into another software. In other words, the project did not show any signs of uncontrolled complexity and it was active indeed. Also, when accesing RemoteTea webpage on SourceForge.net we noticed that the inactive tag was gone, indicating that the project could be reactivated any time by its development team. That is, the project was not totally abandoned (even it did not receive any released version in the last six years). However, in the case of Jo! and LslPlus projects when accessing the respective webpages on SourceForge.net, we did not find any sign of activation by their development team. Posteriorly, we investigate the current state of the remaining six projects that had increasingly complexity. Through an inspection in their website, we found out that all of them, with the exception of IdeaVim project, remained inactive.

TABLE III: DISCONTINUED PROJECTS HOSTED ON SOURCEFORGE.NET HIGH WEEKLY DOWNLOAD RATE

Project ID	**First Version** *(dd-mm-yyyy)*	**Last Version** *(dd-mm-yyy)*	**Released versions**	**Number of downloads**
Saxpath	24-01-2001	26-04-2002	7	21,759
Jaxen	25-07-2001	26-04-2002	10	43,682
Jo1	18-03-2002	07-07-2005	9	23,502
IdeaVIM	17-04-2003	30-03-2010	53	9,568
RemoteTea	14-08-2003	02-01-2008	5	15,474
JRFD	20-12-2003	09-02-2011	26	18,172
G4J	16-09-2004	19-09-2005	11	45,158
Gilead	10-11-2008	22-05-2010	7	28,791
LslPlus	28-07-2008	06-03-2009	14	20,220
MorarChat	21-12-2009	14-01-2012	15	19,535

IdeaVim project is the only one that presents growth on its complexity only until the 4th version, and then have the metric under control, with small variations. Apart from that peak, its behaviour is more similar to an active project, once its complexity is very controlled as the application evolves. Thus, excluding the first 4 released versions, the remaining 39 versions had its complexity controlled, what shows that

development team keep up with the work at the level required by the software growth. Accessing their webpage on SourceForge.net, we found out that they left a note to users, informing that the project was discontinued because Jetbrains Fig. 17: Normalized Complexity standard deviation of discontinued FLOSS projects. took over the IdeamVIM plugin several years ago, and that any support should be directed to the company. Jetbrains is a software development company whose tools are targeted towards software developers and project managers (Hunger, 2010). The company lists 75,000 customers worldwide, among them Apple, LinkedIn, Siemens, and Bank of America.

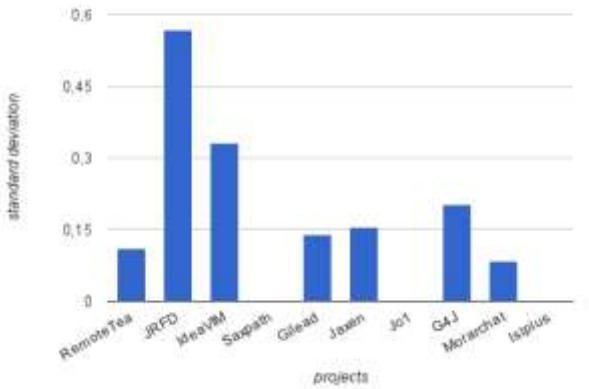

Fig. 29: Complexity by function standard deviation of discontinued s FLOSS projects.

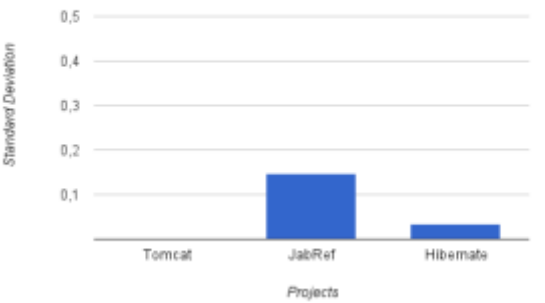

Fig. 30: Complexity by function standard deviation of actives FLOSS projects.

Therefore, it is suitable state that internal properties of the software can not be used to precisely determine whether a project will be abandoned or not, but it can provide an understanding about what could have contributed to software failure. In the case of complexity, we could notice that there is a relation between uncontrolled complexity and software abandonment. Thus, it is important that an analysis over external factors of the software be performed to explore another reasons that could influence the software abandonment. For instance, a study about social impact on software architecture, as we did in a previous work (Siebra, Anjos, & Rolim, 2014), also helps the developers to identify elements that commits the software evolution.

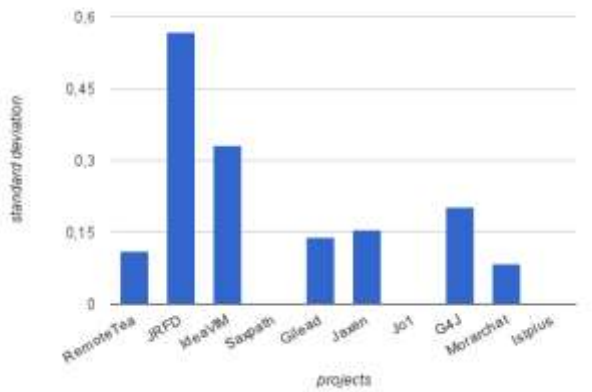

Fig. 29: Complexity by function standard deviation of discontinued s FLOSS projects.

VI. Future Work

This work has limitations that motivates future works. Firstly, the study should be replicated with a larger number of projects samples. The samples used in our work were restricted only to projects written in Java and hosted by SourceForge.net repository. The study would become more complete if it include projects written in other languages, such as C language, once a considerable number of FLOSS projects are written in this language. Also, future works could have projects hosted by other repositories, such as GitHub and Google Code. Moreover, we recognize that there are a lot of other metrics besides the ones we used in this study, that could help in exploring the FLOSS abandonment phenomenon. Additional investigation is necessary, in order to discover others variables that may influence the software failure.

Also, a more detailed analysis needs to be performed, in order to find correlation between another complexity metrics in order to confirm the relation of complexity and software inactivity. In future works, it could be used complex maintainability models instead of only simple complexity metrics, utilizing practical tools for measuring maintainability like SQALE, SIG, ColumbusQM and Quamoco.

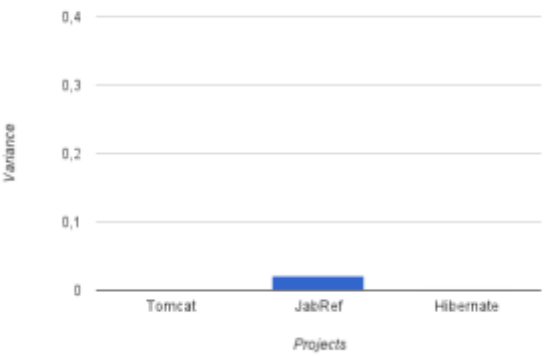

Fig. 32: Complexity by function variance of actives FLOSS projects.

Furthermore, it is import to point out that the results of our work could be also valid for every kind of project, once private systems are even more complex and cohesive than FLOSS (Maccormack et al., 2008). In order to confirm this assumption, future work in this theme is required. Moreover, a further investigation whether the projects analyzed could have evolved to other versions and given origin to successful projects, it would be also a worth future work as well.

Finally, this work has shown another direction for future work. We believe that one work that seems to be relevant is to understand if there is a way to classify if a project is doomed to failure. The failure could be based on known projects that would be used to train a classification mechanism, like a Neural Network. Studies have shown that we can learn with successful projects, but we always forget to learn with projects that have failed.

VII. Conclusion

This paper presents a study of discontinued FLOSS projects complexity. By sampling projects hosted at SourceForge.net, tagged as “inactive”, it was investigated whether there is a relation between the metrics used to analyse FLOSS projects complexity, and the

abandonment of them. Though several tests performed in this work, we found interesting results that confirmed some of this relationships. The majority of the discontinued FLOSS projects we have analysed (60%), showed a substantial complexity grow since its first release compared to its last one. While the active projects revealed a more controlled complexity, presenting almost a constant normalized complexity, some discontinued FLOSS do not seem to be able to control the system complexity. The inability to handle the extra work needed to control the system complexity seems to contribute to the failure of the system, as predicted by Lehman's law of software evolution mentioned earlier. One possible explanation of these failures is the use of anti-patterns, that are responses to recurrent problems that should not be done, because they can jeopardise the projects' evolution and control. We also observed that the complexity by function metric seems to have a stronger relationship with the abandonment of the projects, since it is the metric that shows a larger standard deviation for discontinued projects than the active ones.

Besides, we found some projects (40%) that exhibited a different behaviour, having its complexity's curve more similar to the curve of active projects. A detailed search showed that two out of these four projects was actually still under development.

Through a statistical analysis of the data extracted with SonarQube, we calculated the Pearson's coefficient of correlation between the complexity metrics and we found that the projects have variants behaviours. Some projects, as Jaxen Project and IdeaVim project present strong correlation between all the diverse complexities (class, file and function) and the total complexity. Some of them, as Jo! and Gilead, present a really weak correlation between all of the three variations and the total complexity. Based on this fact, we decided to investigate it further, and then started to investigate what have happened to each project separately.

Despite the fact that the IdeaVim project presents an increasingly growth on its complexity until its 4th released version, later its development team managed to keep the complexity under control. Accessing its website we found out that it was incorporated into another software.

These results lead us to conclude that although the complexity may contribute to the software abandonment, once an uncontrolled complexity demands a higher level of work from contributors, it requires more extra information about the project to guide improvement

decisions related to its activity. Therefore, it is mandatory to also evaluate external elements that may interfere in the software evolution.

We believe that the results of this study can be helpful for FLOSS projects practitioners, since they can find an explanation of abandonment of collaborators. We also believe that this work can be considered as a base for future works in the area of FLOSS projects, specifically in the area of discontinued projects, which still needs further investigation

References

Bakota, T., Hegedus, P., Ladanyi, G., Kortvelyesi, P., Ferenc, R., & Gyimothy, T. (2012). A cost model based on software maintainability. In *Icsm* (p. 316-325). IEEE Computer Society. Retrieved from http://dblp.uni-trier.de/db/conf/icsm/icsm2012.htmlBakotaHKFG12

Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of software engineering*. Wiley.

Beaver, J. M., Cui, X., Charles, J. S., & Potok, T. E. (2009). Modeling success in floss project groups. In T. J. Ostrand (Ed.), *Promise* (p. 16). ACM. Retrieved from http://dblp.uni-trier.de/db/conf/promise/promise2009.htmlBeaverCCP09

Beecher, K., Boldyreff, C., Rank, S., Beecher, K., Boldyreff, C., & Rank, S. (n.d.). Evolutionary success of open source software: An investigation into exogenous drivers. *Electronic Communications of the EASST*, 2008.

Capiluppi, A., & Beecher, K. (2009). Structural complexity and decay in floss systems: An inter-repository study. In A. Winter, R. Ferenc, & J. Knodel (Eds.), *Csmr* (p. 169-178). IEEE. Retrieved from http://dblp.uni-trier.de/db/conf/csmr/csmr2009.htmlCapiluppiB09

Capiluppi, A., Lago, P., & Morisio, M. (2003). Characteristics of open source projects. In *European conference on software maintenance and reengineering*.

Eick, S. G., Graves, T. L., Karr, A. F., Marron, J. S., & Mockus, A. (2001, January). Does code decay? assessing the evidence from change management data. *IEEE Trans. Softw. Eng.*, 27(1), 1–12. Retrieved from http://dx.doi.org/10.1109/32.895984 doi: 10.1109/32.895984

Encyclopedia of measurement and statistics. (2007). In N. J. Salkind (Ed.), *Encyclopedia of measurement and statistics* (??th ed.). SAGE Publications,

Inc. Retrieved from http://dx.doi.org/10.4135/9781412952644 doi: http://dx.doi.org/10.4135/9781412952644

English, R., & Schweik, C. M. (2007). Identifying success and tragedy of floss commons: A preliminary classification of sourceforge.net projects. In *Proceedings of the 29th international conference on software engineering workshops*. ICSEW '07.

Hunger, M. (2010). Jetbrains developer tools. In *infoq*.

Ivanov, A., Andy2222, Bergholm, D., Rosbach, H. K.,

Krakow, K., & Barry, T. (n.d.). *Ffdshow*. http://sourceforge.net/p/ffdshow.

Khondhu, J., Capiluppi, A., & Stol, K. (2013). Is it all lost? A study of inactive open source projects. In *Proceedings of the 9th international conference on open source systems*.

Lehman, M. M. (1978). Programs cities, students, limits to growth? *Programming Methodology*, 42-62.

Maccormack, A., Rusnak, J., Baldwin, C. Y., Maccormack, C. A., Rusnak, J., & Baldwin, C. Y. (2008). *Exploring the duality between product and organizational architectures: A test of the mirroring hypothesis, working paper* 08-039.

Michlmayr, M., Hunt, F., & Probert, D. (2005). Quality practices and problems in free software projects. In M. Scotto & G. Succi (Eds.), *Proceedings of the first international conference on open source systems* (p. 24-28). Genova, Italy.

Schweik, C. M., English, R., Paienjton, Q., & Haire, S. (2010,05/2010). Success and abandonment in open source commons: Selected findings from an empirical study of sourceforge.net projects. In *Second international workshop on building sustainable open source communities (oscomm 2010).*

Siebra, B., Anjos, E., & Rolim, G. (2014). Study on the social impact on software architecture through metrics of modularity. *ICCSA* (5), 618-632.

Stamelos, I., Angelis, L., Oikonomou, A., & Bleris, G. L. (2002). Code quality analysis in open source software development. In *Information systems journal* (Vol. 12, pp. 43–60).

Terceiro, A., & Chavez, C. (2009). Structural complexity evolution in free software projects: A case study. In M. Ali Babar, B. Lundell, & F. van der Linden

(Eds.), *Qacos-osspl 2009: Proceedings of the joint workshop on quality and architectural concerns in open source software (qacos) and open source software and product lines (osspl).*

Wang, Y. (2005). Prediction of success in open source software development. *University of California, Davis.*

Neural Estimation of Food Age with Adaline-based Multi-Layer Perceptron

Ima Okon Essiet

Department of Electrical Engineering,
Bayero University,
Kano, Nigeria
imaessiet82@gmail.com

Abstract: This study employs a 4-input and 1-output feedforward neural network with adalines used to implement learning via error back-propagation (EBP) using least mean square rule. The neural network is used to predict the condition of both cooked and uncooked food as well as fresh vegetables by determining food age (in days). Neurosolutions training software is used to simulate the neural network. Training data is obtained from a constructed metal oxide semiconductor (MOS) ammonia circuit. Results show that a 95% overall accuracy of neural network results is obtained. This demonstrates the capability of neural networks in accurate classification of sample data points. Food samples used to obtain inference database include rice, beans, fresh vegetables, yam and potatoes.

Keywords/Index Terms: neural network, supervised learning, back propagation, e-nose, artificial intelligence

1. Introduction

The advent of neural networks began in the early 1940's when Warren McCulloch and Walter Pitts posited that the behavior of biological neurons could be modeled in the form of propositional logic (McCulloch & Pitts, 1943). In essence, they concluded that the degree of accuracy of a neuron network is directly proportional to the degree of accuracy of the temporal propositional expressions (TPEs) describing it. This idea led to the creation of the artificial neural network (ANN). Since then, ANNs have been successfully used to predict trends in marketing and economics, as well as recognize data patterns in meteorology, medicine, science and technology. The field of neural networks has experienced many advances since 1960, when two of the earliest feedforward neural network algorithms were introduced. These were the least mean square algorithm by Widrow and Hoff in 1960 and the perceptron rule by Rosenblatt in 1962 (Widrow & Lehr, 1995). Madaline Rule I was the earliest learning rule for feedforward networks with multiple adaptive elements. Paul Werbos extended this discovery in 1971 by

developing the back propagation algorithm for multilayer neural networks. The discovery of back propagation has made it possible to employ neural networks in non-linear, high-precision problems (Widrow & Lehr, 1995). Back propagation neural networks have successfully been used in vehicle autonomy, expert systems, speech recognition and so on.

An artificial neural network has a structure similar to that of the biological neuron which consists of a soma (body of the neuron) and an axon (receptor). Neurons send and receive signals to and from each other via a structure called synapse. These synaptic signals can be either excitatory or inhibitory. Each neuron has a signal threshold which has to be overcome in order for excitation to occur within the neuron. The threshold varies from neuron to neuron which makes its value non-deterministic in nature. McCulloch and Pitts attribute this non-linear nature to either synaptic irreciprocity or varying neural anatomical configurations (McCulloch & Pitts, 1943). There are cases in which an applied stimulus does not excite a particular neuron even when the signal is increased to maximum value. In such cases, the neuron is said to exhibit neural inhibition. Inhibition could be either relative or absolute.

Ammonia is a volatile organic compound (VOC) which is present in nearly all biological matter and has been used as a measure of bacterial or microbial activity in both plants and animals. As a result, it can be used to measure the extent of decay of organic materials depending on the level of its concentration within them. Ammonia is also found in minute quantities in the atmosphere as a result of the decay of plant and animal matter. This decay process is called putrefaction and describes the decomposition of proteins due to bacterial or fungal action. This work investigates the use of an ANN to predict the age of food by estimating the concentration of ammonia in the food samples. The results of the work aim to realize a real-time monitoring system which could be adopted by farmers and other food-based organizations to monitor the condition of food while in storage. 50% of farming households, 30% of rural households and 20% of urban households across Africa still experience food insecurity as a result of food spoilage during storage (Mwaniki, 2004). The results of the work would help to reduce food insecurity caused by spoilage which is a huge problem both within Africa and the world as a whole.

An adaline (adaptive linear network) is a single-layer neural network consisting of a weight, a bias and a summation function. It differs from the McCulloch-Pitts perceptron in

the sense that weights are adjusted according to the weighted sum of inputs in the former, whereas in the latter, the total input is passed to the transfer function while the function's output is used for adjusting weights. In single-element neural networks, adaptive algorithms such as LMS or perceptron rule are used to adjust adaline weights to respond accurately to as many patterns as possible in a training set with binary desired responses (Widrow & Lehr, 1990). Once the weights are adjusted, responses of the trained element can be tested by applying various input patterns. Correct response to input patterns not included in the training set demonstrates that adalines are capable of generalization. The adaline's structure is shown in Fig. 1. A multi-layer perceptron (MLP) is a neural network structure consisting of at least one hidden (process) layer. MLPs are particularly suited for cases involving complex data classification. This is because multiple process layers help to avoid the problem of premature convergence associated with such cases. A madaline consists of several adaptive linear neurons and is shown in Fig. 2.

The metal oxide semiconductor (MOS) sensors have been used in the past for many sensing applications. For example, see the review by Huang and Wan (Huang & Wan, 2009) and also Arshak *et al* (Arshak, K. et al, 2004). There are many semiconducting metal oxides that are known to respond to various gases and other volatile organic compounds. These therefore have been used widely in food industries (Ray, 2005; Kizil & Lindley, 2009), medical applications (Kodogiannis, V.S. et al, 2008), agriculture (Persaud, K.C. et al, 1994), detection of hazardous gases (Noorsal & Sidek, 2004) and many other applications (Berna, 2010; Pawar et al, 2012). However, in spite of this wide application one of the greatest challenges facing electronic nose (e-nose) designs is that of cost. Most forms of the e-nose that have been successfully implemented are too expensive for use by the general public. As a result, their widespread use has been severely limited. The e-nose proposed in this article has been designed to address this limitation. The components in the sensor's circuitry are affordable compared to most of the existing implementations. The MOS sensor used in this work detects ammonia concentration in the parts per million (ppm) range.

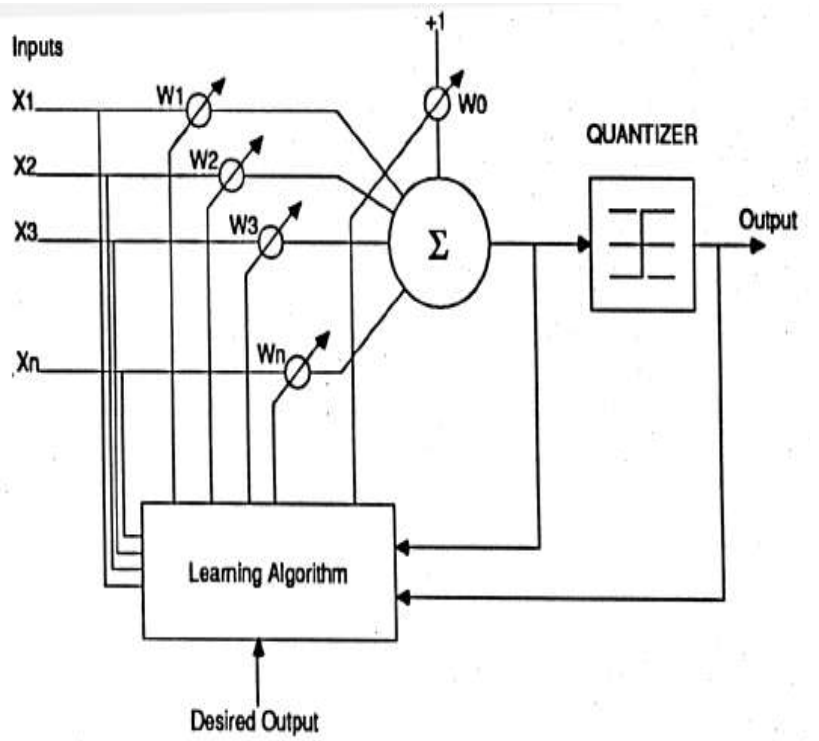

Figure 1. Adaptive linear neuron (Mehrotra, Mohan, & Ranka, 2003)

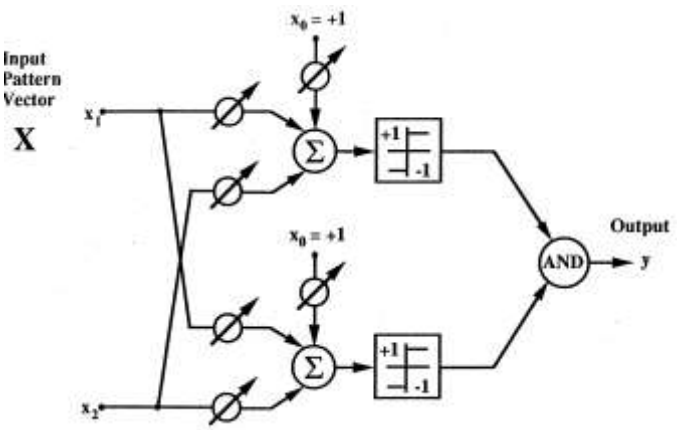

Figure 2. Madaline consisting of 2 adalines (Widrow & Lehr, Perceptrons, Adalines and Backpropagation, 1995)

2. Methodology

Adaline weights are adjusted according to the Widrow-Hoff learning rule, also known as the LMS rule. This is represented mathematically as:

$$w_{g,h} : w_{g,h} + \eta(t-a)x \qquad (1)$$

From (1), g and h are any two interconnected adalines, η is the learning rate, t and a are the target and actual adaline outputs. The adaline converges to the least square error given by:

$$E = (t-a)^2 \qquad (2)$$

Equation (2) represents the gradient descent update for linear regression and is justified by the following:

$$\frac{\partial E}{\partial w} = \frac{\partial (t-a)^2}{\partial w} = 2(t-a)x \qquad (3)$$

Equation (3) shows that moving in direction of $(t-a)x$ increases error while moving in the opposite direction decreases error. This scenario presents a kind of hill-climbing approach to obtaining the best possible solution. Fig. 3 shows the form of error landscape in the weight space.

The aim of LMS is to minimize the mean square error (MSE) according to the following:

$$MSE = \frac{1}{n}\sum_{j=1}^{n} e(j)^2 = \frac{1}{n}\sum_{j=1}^{n} (t(j) - a(j))^2$$

(4)

The following steps summarize the adaline's learning procedure:

(i) Initializing the network and threshold weights $(w_1 .....w_n)$ and w_t. This involves setting both network and threshold weights to small bipolar random values

(ii) The new input vector $x_1, x_2, ...x_n$ and the target output $t(j)$ are presented. x_0 is a fixed bias with a value of 1, while $t(j)$ can be ± 1.

(iii) the actual output $a(t)$ is obtained according to:

$$a(t) = \Gamma[\sum_{i=0}^{n} w_i(t) * x_i(t)] \qquad (5)$$

From (5), $\Gamma(E) = 1 \; when \; E > 0 \; and \; -1 \; when \; E \le 0.$

(iv) Adaptation of weights according to

$$w_i(t+1) = w_i(t) + \eta[t(j) - \sum_{i=1}^{n} w_i(t)x_i(t)]x_i(t)$$

(6)

Steps (ii) to (iv) are repeated until the target outputs and actual network outputs are all equal for all input vectors of the training set.

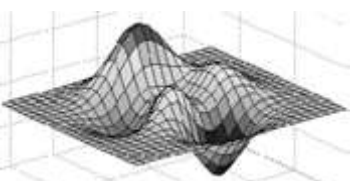

Figure 3. Error landscape in weight space (Mehrotra, Mohan, & Ranka, 2003)

The LMS algorithm therefore chooses a random input-output pair

from the training data set. The neural network then operates on the input to generate a corresponding output. This output is used to compute the output error between the target and actual outputs. Weights are then adjusted according to the input and error vectors, after which another random pair is selected and the error correction process repeated. Stopping criteria is checked at the end of each epoch. The stopping criteria may be a preset number of epochs or a threshold error value. In this work, the stopping criterion is the degree of similarity between training and cross-validation data. Neurosolutions training software performs cross-validation by splitting the entire data into a number of subsets of equal size. Each subset is used for testing and the remainder for training. The error estimates are averaged to yield an overall error estimate. In this work, half of the data is used for training and the other half is used for testing to maintain network symmetry. The algorithm below summarizes the LMS approach.

Network training begins with randomly chosen initial weight values (between -1.0 and 1.0) (Mehrotra, Mohan, & Ranka, 2003). Larger magnitudes may drive hidden layer node outputs to saturation, thus requiring long training time to emerge from saturation. The learning rate η is varied by increasing its value at iterations that improve network performance significantly and decreasing it at iterations which worsen performance significantly. Iteration range is typically between 0.1 and 0.8. Two hidden layers are used in order to further reduce likelihood of premature convergence. The overall structure of the neural network is shown in fig. 4.

Training data points are obtained from a sensor circuit with ammonia sensitivity of 0.033V/ppm (parts per million). The sensor is the metal oxide semiconductor (MOS) type. Neurosolutions software was used in training and testing of the neural network's results for both cooked and uncooked food samples. Neural network training and testing points were obtained from the following relation:

$$Sample\ \ point(S_p) = \frac{ammonia\ concentration\ of\ test\ point}{60\,ppm} \quad (7)$$

60ppm was chosen because testing was limited to a maximum of 150 days. This concentration was observed for food samples 150 days and older.

Algorithm 1:

While MSE is undesirable and iterative boundaries are not exceeded,

do

for each input pattern $i_p : 1 < p < N$,

compute hidden node inputs;
compute hidden node outputs;
compute inputs to output nodes;
compute network outputs;

compute error btw actual and desired
outputs;
modify weights btw hidden and output nodes;
modify weights btw input and hidden
nodes;
end for
end

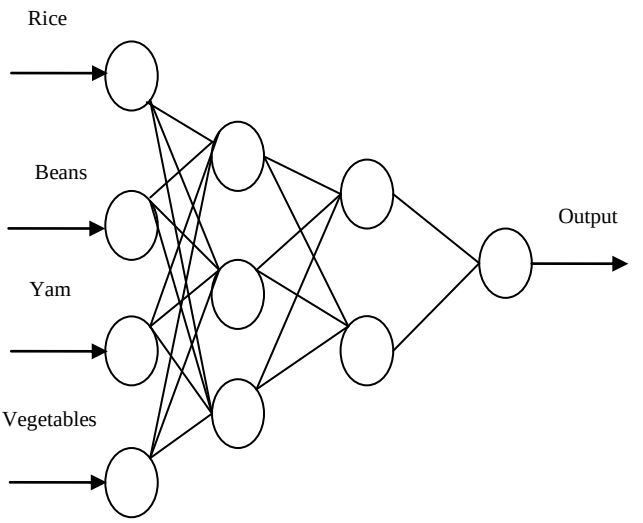

Figure 4. Adaline-based neural network structure

The network shown in fig. 4 has an activation (threshold) function described by a sigmoid transfer function. The sigmoid function is used because the behavior of ammonia sample points is semi-linear as build-up of ammonia is concerned. The behavior for both fresh and decayed samples is almost exponential as given by the function:

$$f(x) = \frac{1}{1 + e^{-\sum_i w_i x_i}} \quad (8)$$

The ammonia sensor resistance is obtained according to:

$$R_{sensor} = \frac{V_{supply} - V_{output}}{V_{output}} \times R_{load} \quad (9)$$

Equation (9) is used to obtain the sensor's ammonia sensitivity which is used to determine the concentration of individual test points for the different food types. Fig. 5 shows the constructed sensor circuit.

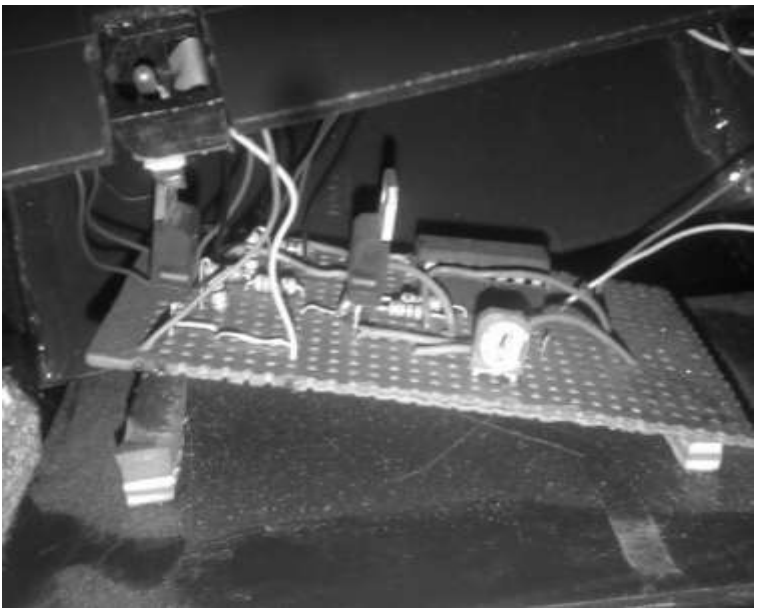

Figure 5. Constructed ammonia sensor circuit

3.Results and Discussion

The NeuroSolutions training network is shown in figure 5. The training breadboard shows the sigmoid transfer function being applied to the process layers. Data samples are entered to the neural network as an Excel spreadsheet. The training software allows specification of the training-t0-cross-validation data ratio. However, this ratio should not be too high to allow the network to determine how well it has learnt; neither should it be too low to avoid false convergence. Separate training sessions were performed for cooked and uncooked food samples as well as fresh vegetables respectively. Ten (10) training sessions were performed for each set of food samples with 50 test points per training session. Successful training outcomes for all sets of samples are shown in tables 2, 3 and 4. From results obtained, it can be seen that for all food samples tested, ammonia concentration increases as food samples advance in age. Table 2 shows that ammonia content in cooked food samples increased drastically as the samples aged beyond 2 days (5.10ppm to 21.36ppm). This is due to increased microbial activity in cooked food especially for the beans samples which had high protein content. A maximum ammonia concentration of about 25ppm was obtained for cooked food samples. Neural network results for uncooked food samples are shown in table 3. It was observed that ammonia concentration in these samples is less than those of cooked food samples. For instance, a yam sample at 60 days old had ammonia concentration

of 4.68ppm while the same sample in cooked form had 5.10ppm concentration at only 2 days old.

Vegetables had the highest values of ammonia content with 37.92ppm being obtained for 7-day old samples. It should be noted that testing was carried out without refrigeration of any food samples. The classification accuracy of the neural network for the three classes of samples (cooked, uncooked and vegetables) is summarised in table 4. An overall classification accuracy of 95% demonstrates that the selected neural network architecture is reliable in terms of using ammonia content to determine food age. Incorrect results are attributed to the subjective selection of learning rate η and internal weight assignment especially between the two process layers.

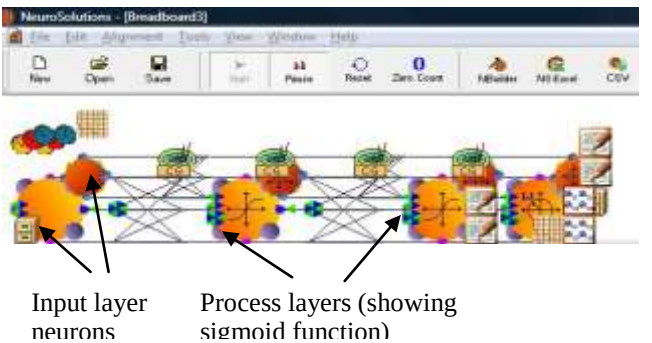

Figure 5. Neuro Solutions Simulation Breadboard

Table I. Neural simulation of cooked food samples (rice, beans, yam, and potatoes)

Training session	Neural Network output	Ammonia concentration (ppm)	Food age (days)
1	0.017, 0.050	1.02 – 3.00	0 – 2
2	0.021, 0.048	1.26 – 2.88	0 – 2
3	0.015, 0.012	0.72 – 0.90	0 – 2
4	0.045, 0.020	1.20 – 2.70	0 – 2
5	0.061, 0.085	3.66 – 5.10	0 – 2
6	0.40,	21.36 – 24.00	3 – 5

	0.356		
7	0.390, 0.413	23.40 – 24.78	3 – 5
8	0.415, 0.514	24.90 – 30.84	3 – 5
9	0.350, 0.398	21.00 – 23.88	3 – 5
10	0.417, 0.303	18.18 – 25.02	3 - 5

Table II**.** Neural simulation of uncooked food samples (rice, beans, yam, and potatoes)

Training session	Neural Network output	Ammonia concentration (ppm)	Food age (days)
1	0.073, 0.044	2.64 – 4.38	0 - 60
2	0.065, 0.040	2.40 – 4.00	0 -60
3	0.088, 0.091	5.28 – 5.46	0 - 60
4	0.097, 0.102	5.82 – 6.12	0 – 60
5	0.050, 0.078	3.00 – 4.68	0 – 60
6	0.083, 0.092	4.98 – 5.52	61 – 150
7	0.116, 0.101	6.06 – 6.96	61 - 150
8	0.122, 0.104	6.24 – 7.32	61 - 150
9	0.111, 0.106	6.36 – 6.66	61 – 150
10	0.113, 0.119	6.78 – 7.14	61 - 150

Table III. Neural simulation of fresh vegetables (cabbage, lettuce and carrots)

Training session	Neural Network output	Ammonia concentration (ppm)	Food age (days)
1	0.050, 0.113	3.00 – 6.78	0 - 3
2	0.092, 0.108	5.52 – 6.48	0 - 3
3	0.099, 0.112	5.94 – 6.72	0 – 3
4	0.108, 0.116	6.48 – 6.96	0 – 3
5	0.089, 0.101	5.34 – 6.06	0 – 3
6	0.253, 0.305	15.18 – 18.3	4 – 7
7	0.362, 0.486	21.72 – 29.16	4 - 7
8	0.549, 0.628	32.94 – 37.68	4 – 7
9	0.636, 0.645	38.16 – 38.70	4 – 7
10	0.608 , 0.632	36.48 – 37.92	4 - 7

Table IV. Overall Accuracy of Neural Simulation

	Correct (%)	Incorrect (%)	Accuracy (%)
Cooked	94	6	94.0
Uncooked	96	4	96.0
Vegetables	95	5	95.0
Average Overall accuracy			**95.0**

4. Conclusion

The obtained simulation results demonstrate that neural networks are capable of data classification to a high degree of accuracy when network parameters are well selected. Inaccuracies arise when network parameters are not well chosen. In this case, the network becomes saturated and therefore fails to converge. This observation informed the use of two process

layers in the adopted neural network model to provide adequate time for the network to settle and eventually converge. The learning rate was closely monitored to also prevent premature convergence. The technique of cross-validation applied to the training data also increased the accuracy of classification results. It has also been observed that ammonia concentration increases rapidly in cooked and protein-based food samples when compared with uncooked samples. Vegetable samples had the highest ammonia content. Further work will concentrate on improving network accuracy by determining a more objective approach to learning rate and internal weight manipulation.

References

Arshak, K. et al. (2004). A Review of Gas Sensors Employed in Electronic Nose Applications. *Sensor Review* , 181-198.

Berna, A. (2010). Metal Oxide Sensors for Electronic Noses and Their Application to Food Analysis. *Sensors* , 3882-3910.

Huang, J., & Wan, Q. (2009). Gas Sensors Based on Semiconducting Metal Oxide One-Dimensional Nanostructures. *Sensors* , 9903-9924.

Kizil, U., & Lindley, J. (2009). Potential Use of Gas Sensors in Beef Manure Nutrient Content Estimations. *African Journal of Biotechnology* , 2790-2795.

Kodogiannis, V.S. et al. (2008). Artificial Odor Discrimination System using Electronic Nose and Neural Networks for the Identification of Urinary Tract Infections. *IEEE Transactions on Info. Tech. in Biomedicine* , 707-708.

McCulloch, W. S., & Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* , 115-133.

Mehrotra, K., Mohan, C., & Ranka, S. (2003). *Elements of Artificial Neural Networks.* Chicago.

Mwaniki, A. (2004). *Food Insecurity in the Third World Nations.* Cornell.

Noorsal, E., & Sidek, O. (2004). Development of Odor Sensor System using QCM Sensor Array. *3rd Int. Conf. on Electrical and Computer Engineering*, (pp. 526-528).

Pawar, N.K.;Kajale, D.D. et al. (2012). Nanostructured Fe2O3 Thick Film as Ethanlo Sensor. *Int. Journal on Smart Sensing and Intelligent Systems* , 441-457.

Persaud, K.C. et al. (1994). *Odor Evaluation of Foods using Conducting Polymer Arrays and Neural Network Pattern Recognition* . Tokyo: Springer-Verlag.

Ray, B. (2005). *Fundamental Food Microbiology.* Boca Raton: CRC Press.

Widrow, B., & Lehr, M. (1990). 30

Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation. *Proceedings of the IEEE,* (pp. 1415-1442).

Widrow, B., & Lehr, M. (1995). *Perceptrons, Adalines and Backpropagation.* Cambridge: MIT Press

Institutional E-Waste Management: Comparison of Practices at two Tertiary Institutions in Nigeria

Omenogo, Mejabi1

1 University of Ilorin, Ilorin, Nigeria
Contact(s). ovmejabi@unilorin.edu.ng

Abstract—Electronic waste (e-waste) comprises of waste from information and communication technology (ICT) equipment, devices and materials as well as others such as refrigerators, televisions, and air conditioners. Particularly, e-waste handling has become a major issue in recent times due to the increasing number of computer and wireless telephone users. Mostly, in developing countries, e-waste are usually improperly disposed of by burying, burning, employing unconventional or unsafe recycling methods and in the extreme, by doing nothing – simply storing the unusable e-waste away. Having recognized that large institutions are major consumers of electronic products, this paper presents a study of e-waste management practices at two institutions in Nigeria that were established in the 70's. They were selected for their age in the expectation that electronic products would have been purchased and disposed over the past 30 years or more. A questionnaire was used to obtain data from departments in the institutions and the data collected analyzed using descriptive statistics and the chi-square test at the 0.05 level of significance. The findings from the study are presented as a comparison of both institutions and recommendations are made towards the green disposal of e-waste in the institutions.

Keywords/Index Terms—e-waste, EEE, green disposal, ICT waste, WEEE.

Introduction

The terms, electronic waste (e-waste) and waste electronic and electrical equipment (WEEE) are used interchangeably (Deathe et al., 2008; Monika, 2010; Schoenung, 2005) to describe almost any household or business item with circuitry or electrical components with power or battery supply that has or could enter the waste stream (Man et al., 2013; Wang et al, 2013). E-waste includes information and communication technology (ICT) equipment, home electrical appliances, audio and video products, and all of their peripherals (Bandyopadhyay, 2008). A comprehensive listing of equipment and appliances that are considered electronic or electrical have been provided by the European Union (UNEP, 2007; Shah & Shaikh, 2008). E-waste is a generic term encompassing various forms of electrical and electronic equipment (EEE) that are old, end-of-life electronic appliances and have ceased to be of any value to their owners (Bandyopadhyay, 2008).

It is well documented that e-waste is the fastest growing segment of municipal solid waste world-wide, with the United States, Western Europe, China, Japan, and Australia being the major (Deathe et al., 2008; Monika, 2010). Of note is that most of the waste was from large businesses and institutions (USEPA, n.d.). Countries in Africa are also contributing to the stream of e-waste particularly because most ICT acquisition and ownership of large and small household appliances such as air-conditioners and refrigerators, depends more on second-hand or refurbished electrical and electronic equipment which are usually imported without confirmatory testing for functionality (Osibanjo & Nnorom, 2007). Indeed many West African re-use markets prefer refrigerators and televisions from European countries because of compatibility with power formats and broadcasting systems (Ogungbuyi et al., 2012).

Financial constraints have been flagged as a major factor in the trade in second-hand electronic goods in the African region (Fagbohun, 2011). For instance, in 2009 trade in second-hand Blackberries was reported to be booming in Lagos, Nigeria, with prices ranging between $25 and $65 (Leyden, 2009). The fact that information products generally have short life cycles (three to six years) (United States Agency for Natural Resources, 2004) with personal computers dropping to two years by 2005 (Oteng-Ababio, 2010). This means that the contribution of ICT products to the waste stream is significant.

Aginam (2008) reported that dumping of e-waste into the African markets, especially Nigeria, continued despite measures put in place by regulatory authorities such as the Standards Organization of Nigeria and the Computer Professionals Registration Council of Nigeria. Other countries where heavy dumping of e-waste occurs include China and Ghana (Aginam, 2008) as well as India **(**Needhidasan et al**.,** 2014) and Kenya (Onderi, 2011).

Electronic and electrical equipment (EEE) become e-waste as a result of obsolescence due to advancement in technology, changes in fashion, style and status, and nearing the end of the product's useful life (Needhidasan et al., 2014; Ramachandra & Saira, 2004; Schwarzer et al., 2005). There are two ways that e-waste is handled at the end-of-life phase. It is either disposed of improperly and in an unsafe manner or it is done safely for environmental sustainability and good health. Improper ways of disposing e-waste include the following:

- Land-filling (i.e. burying) which can cause environmental and health hazards arising from mercury, cadmium, lead, etc, leaching into the soil and groundwater (Ramachandra & Saira, 2004; Chen et al., 2011).
- Incineration (i.e. burning) which can cause emission of toxic fumes and gases, thereby polluting the surrounding air. Also leads to ozone depletion by the release of gases such as polychlorinated biphenyls (Waema & Mureithi, 2011).
- Dumping i.e. shipping from industrialised countries to developing countries which was quite commonplace in the late 1980's (Osibanjo & Nnorom, 2007).
- Recycling by unconventional or unsafe recycling methods to produce useable equipment or extract economically viable components such as gold, silver, platinum and palladium (Alake & Ighalo, 2012; Needhidasan et al. 2014; Oresanya, 2011).
- Storage (a 'do nothing approach') which is common where there are no policies to guide the disposal of e-waste (Ramachandra & Saira, 2004).

International outrage following indiscriminate dumping of waste, including e-waste, by the developed nations led to the drafting and adoption of the Basel Convention on the Control of Trans-boundary Movements of Hazardous Wastes and their Disposal, usually known simply as the Basel Convention (http://www.basel.int). The Convention was opened for signature on 22 March 1989, and entered into force on 05 May 1992. Nigeria signed up in 1990, Kenya in 2000, and Ghana in 2003, amongst others. The Convention provides assistance and guidelines on legal and technical issues, gathers statistical data, and conducts training on the proper management of hazardous waste (Sthiannopkao & Wong, 2013).

Green disposal is advocated as the best way of disposing e-waste. It emanates from green computing which has the goals of reducing the use of hazardous materials, maximizing energy efficiency during the product's lifetime, and promoting the recyclability or biodegradability of defunct products and factory waste (Bossuet, 2014). Green information technologies and systems refer to initiatives and programs that directly or indirectly address environmental sustainability in organizations (Jenkin et al., 2011). To promote green disposal of e-waste, Ramachandra and Saira (2004) advocate that management of e-waste should begin at the point of generation by minimizing the waste

and practising sustainable product design. They propose waste minimization strategies that revolve around inventory management, production-process modification, volume reduction, and recovery and reuse. Sustainable product design, according to them, should include rethinking the product design, use of renewable materials and energy, and use of non-renewable materials that are safer.

Improving final disposal is also advocated through setting strict guidelines for landfill management and building disposal capacities by the establishment of a system of national treatment facilities. In this regard, countries like the United Kingdom have reported significant reductions in the amount of waste sent to landfill (down by over a third since 2001), while households recycled over 38% of their waste in 2010 compared to only 9% in 2000 and recycling from green waste went up 13% in the last decade (The Green IT Review, 2010).

According to Ramachandra and Saira (2004) there are roles to be played by government, industries and citizens in promoting green disposal of e-waste. They suggest that governments should provide an adequate system of laws, controls and administrative procedures for hazardous waste management and educate e-waste generators on reuse/recycling options. On the role of industries, adoption of waste minimization techniques, which will make a significant reduction in the quantity of e-waste generated and thereby lessen the impact on the environment in addition to manufacturers, distributors, and retailers undertaking the responsibility of recycling/disposing of their own products. Furthermore, citizens are advised to donate EEE for reuse, taking care that such items are in working condition; never to dispose e-waste with garbage and other household wastes, but rather to take them to a designated collection point; and to opt for those with green manufacturing policies while buying electronic products.

Statement of Problem

Ramachandra and Saira (2004) reported that an estimated 75% of electronic items are stored due to uncertainty of how to manage it and they stay unattended in houses, offices, warehouses etc. until they are mixed with household wastes, which are finally disposed of at landfills. Osibanjo and Nnorom (2007) highlight the challenges facing the developing countries in e-waste management as an absence of infrastructure for appropriate waste management, an absence of legislation dealing specifically with e-waste, and an absence of any framework for end-of-life product

take-back or implementation of extended producer responsibility. The result is an e-waste problem for a country like Nigeria where formal equipment take-back, re-use and disposal structures are missing.

When the e-waste is not properly disposed, the toxic substances such as lead and mercury, present in components of EEE can be harmful to humans and other organisms (Yousif, 2009). For example, lead which is used in batteries, solders, as alloying element for machining metals, and printed circuit boards are very toxic to aquatic organisms and to humans. Also, mercury, which is used in in thermostats, sensors, relays and switches which is toxic by inhalation can damage the central nervous system and kidneys in humans in addition to long term effects in the aquatic environment. As a result, it is important to have empirical evidence of what is being done with e-waste in Nigeria. The study by Alake and Aghalo (2012) focused on Alaba International Market, Badagry road, Lagos, which is a major in-let for electronic products (especially second hand electronic goods) in Nigeria, and examined e-waste disposal practice by electronic repair technicians, domestic electronic goods consumers and some distributors of electronic products. While the study gives information on how e-waste is managed in relation to a major EEE market in Nigeria, it does not provide any insight into corporate use and disposal of EEE.

Objective of the Study

To get ready for formal e-waste management in Nigeria, the type and amount of e-waste being generated by different stakeholders must be ascertained in addition to identifying how disposal is presently being handled. In recognition of the fact that large institutions greatly contribute to the growing stream of e-waste[8] this study sought to determine the type and magnitude of e-waste being generated by large institutions as well as to identify e-waste management practices at the institutions. The contribution of academic institutions to the e-waste problem was recognised by Iyer (2014), hence the study of attitude towards e-waste collection and safe management in lection in academic institutions in Bangalore. Specifically, the sub-objectives are to:

1. Compare the amount of selected EEE acquired by the study institutions from 2008-2011
2. Quantify selected EEE at the study institutions that are not in working condition
3. Identify the source of acquired EEE by the

institutions
4. Identify e-waste disposal practices at the institutions
5. Make recommendations for green disposal of EEE where necessary.

As a preliminary study, the scope was limited to a study of two of the oldest tertiary institutions in Kwara State, Nigeria – the University of Ilorin (Unilorin) and the Kwara State Polytechnic (Kwara Poly), focusing only on the academic departments.

Policies and Practices that Promote Green Disposal of E-Waste

At the international level, the Basel Convention focuses on the control of trans-boundary movements of hazardous wastes and their disposal. Regional policies are also influenced by non-governmental bodies. For example, the Council of European Professional Informatics Societies (CEPIS) works at promoting the ideas of Green ICT among its members in order to contribute to the environment's protection (CEPIS, 2012). At the country level, such policies or guidelines usually outline the responsibilities and roles of government, industries, and consumers (organizations and citizens).

In Nigeria, institutional and legal frameworks to regulate e-waste management in the country include the establishment of the National Environmental Standards and Regulations Enforcement Agency (NESREA), in 2007, replacing the Federal Environmental Protection Agency Act Cap F 10 LFN 2004. Nigeria signed up on the Bamako Convention in 2008. Nigeria also has a Ministry of Environment at both the Federal level and at the State level. Although, plans have been underway for a National Policy on e-waste such a policy is yet to be released. Nevertheless, a memorandum of understanding was brokered between NESREA, the Standards Organization of Nigeria, the Consumer Protection Council and the Alaba International Market Amalgamated Traders Association (Alaba being a major trading centre for electronic equipment in Lagos, Nigeria), to fight e-waste and piracy (Fagbohun, 2011).

Ramachandra and Saira (2004) propose the following things governments should do to promote green disposal of ICT waste - Government should provide an adequate system of laws, controls and administrative procedures for hazardous waste management , come up with a comprehensive law that provides e-waste regulation and management and proper disposal of hazardous wastes, establish an agency that is responsible for waste management, encourage beneficial reuse of e-waste and encourage business activities that use e-waste,

set up programs to promote recycling among citizens and businesses, and educate e-waste generators on reuse/recycling options.

They recommend that industries that generate waste should take responsibility for it; all personnel involved in handling e-waste in industries including those at the policy, management, control and operational levels, should be properly qualified and trained; companies should adopt waste minimization techniques, which will make a significant reduction in the quantity of e-waste generated and thereby lessen the impact on the environment; manufacturers, distributors, and retailers should undertake the responsibility of recycling/disposal of their own products; manufacturers must be responsible for educating consumers and the general public regarding the potential threat to public health and the environment posed by their products. Furthermore, they recommend that consumers should donate EEE for reuse - but care should be taken while donating such items i.e. the items should be in working condition; e-wastes should never be disposed with garbage and other household wastes – take to a designated collection point; and while buying electronic products opt for those with green manufacturing policies.

Methodology

The tertiary institutions selected for this study were founded in 1973 (Kwara Poly) and 1975 (Unilorin) respectively. They are both located in the city of Ilorin in Kwara State, in the north-central zone of Nigeria. Academic activities at the Kwara Poly are carried out through 26 departments aggregated under five (5) institutes – Basic and Applied Sciences, Finance and Management Studies, Environmental Sciences, Information and Communication Technology, and Technology. Unilorin on the other hand has 81 academic departments clustered under 12 faculties. The faculties are: Agriculture, Arts, Basic Medical Sciences, Business and Social Sciences, Clinical Sciences, Communication and Information Sciences, Education, Engineering and Technology, Law, Pharmaceutical Sciences, Science and Veterinary Medicine.

Since the study was an institutional one, all 26 academic departments at Kwara Poly were selected as well as departments from closely related faculties at Unilorin. The related faculties at Unilorin were: Business and Social Sciences (six departments), Communication and Information Sciences (five departments), Engineering and Technology (seven departments) and Science (nine departments). This gave a total of 27 departments to be

sampled at Unilorin. The Stores department at both institutions was visited for additional information on methods of e-waste disposal.

In developing the data collection instrument, the first task was to determine what EEE were of interest in a large institution while still having a manageable list to deal with in the study. Starting with the list used in a national e-waste management study in Kenya (Waema & Mureithi, 2011) and using a combination of consumer electronics lists classified as information products by the US Agency for Natural Resources (2004) and the European Union list of equipment and appliances (UNEP, 2007) the 13-item list for the EEE used for this study was generated. Life expectancy for some products was as low as 2 years with some others reaching 15 years (Jackson, 2007; United States Agency for Natural Resources, 2004).

TABLE 1: ELECTRONIC AND ELECTRICAL EQUIPMENT CONSIDERED IN THE STUDY WITH THE LIFE EXPECTANCY

S/No.	Electronic/Electrical equipment	Life Expectancy (in years) †	Life Expectancy (in years) ‡
1.	Desktop computers	3 to 6	
2.	Laptop/Notebook computers		2.5
3.	Monitors (CRTs)	6 to 7	
4.	Flat screens (VCDs)	13 to 15	
5.	Printers	3 to 5	
6.	Mobile phones		2
7.	Radio sets	3 to 15	
8.	Televisions	13 to 15	
9.	Fridges		9 to 13
10.	Air conditioners		10 to 15
11.	Photocopiers		4
12.	Fax Machines	3 to 6	
13.	Uninterrupted Power Supply (UPS)		6

† Source: US Agency for Natural Resources (2004, p.4). Electronic Waste Management in Vermont. Available at http://www.anr.state.vt.us/dec/wastediv/recycling/pubs/Electronics_Legislative_Report_gl.pdf

‡ Source: "Study of Life Expectancy of Home Components", National Association of Home Builders/Bank of America Home Equity, February, 2007, p.7

The structured questionnaire developed for collecting the data had six main items as follows:

1. What year was this Department established?
2. How many items of the following types do you have in your department in working condition and also the number not in use. A column for items in working condition and another column for bad or obsolete items was provided. The aim of this question was to determine the quantities on ground.
3. How many items of the following types did your department purchase in the last four years? A column was provided for each year between

2011 and 2008, starting with 2011. This item was included to provide an estimate of recent acquisitions which could be compared to the numbers on ground.

4. Where do you usually acquire indicated equipment from?
5. Do you keep inventory of the equipment you discard or store?
6. Kindly use the table below to indicate what you do with items that you do not use anymore. Four options were given in addition to 'Others. Please specify'. The options were A – Store away in the department or unit, B – Throw away with other waste, C – Send them to the Stores department, D – Disassemble to reuse some parts to repair others, and E – Others. Please specify.

The developed instrument was pre-tested before administering the final version. Copies of the questionnaire were distributed to the selected departments with several follow-up visits thereafter. Several departments were unable to provide the required information either because the records were unreachable or due to unwillingness to complete the questionnaire. Of the 26 departments targeted at Kwara Poly, 16 completed the questionnaire while 16 out of the 27 at Unilorin also did, giving a return rate of 62 percent and 59 percent respectively. The data collected was analysed using frequency distributions, summations and the Chi-Square statistic to test for independence. When necessary, further computation such as finding the proportion of available equipment that was not in working condition was undertaken.

Results and discussion

Analysis of the data collected, revealed that the proportion of equipment that were not in working condition were generally low across all departments sampled except at Unilorin where a high proportion of desktop computers (38%) and photocopiers (24%) were not in working condition (Table 2). The equipment with the highest out of order proportion at Kwara Poly was the photocopying machine (17%).

Furthermore, the deduction could be made that most of the equipment found in the departments were purchased between 2008 and 2011, simply by comparing the total purchases during that period (Table 3) with the numbers found in the department (Table 2). This raises the question – if most of the departments in the study were established at least 20 years ago, what has happened to the e-waste generated from equipment end-of-life? Further esults are also presented in Table 4 and Table 5

TABLE 2: NUMBER OF EQUIPMENT IN WORKING OR NOT IN WORKING CONDITION AT THE SAMPLED DEPARTMENTS

S/No.	Equipment	Kwara Polytechnic			University of Ilorin		
		No. in working condition (A)	No. not in use (B)	Percent (%) not in working condition†	No. in working condition (A)	No. not in use (B)	Percent (%) not in working condition†
1	Desktop computers	67	4	5.6	29	18	38.3
2	Laptop/Notebook computers	20	1	4.8	42	0	0.0
3	Monitors (CRTs)	20	0	0.0	66	5	7.0
4	Flat screens (VCDs)	49	0	0.0	60	1	1.6
5	Printers	24	2	7.7	39	5	11.4
6	Mobile phones	37	0	0.0	3	0	0.0
7	Radio sets	10	1	9.1	3	0	0.0
8	Televisions	26	0	0.0	28	3	9.7
9	Fridges	39	1	2.5	17	3	15.0
10	Air conditioners	39	1	2.5	138	11	7.4
11	Photocopiers	5	1	16.7	16	5	23.8
12	Fax Machines	2	0	0.0	0	0	0.0
13	UPS	60	4	6.3	116	11	8.7

TABLE 3: ITEMS ACQUIRED IN FOUR YEARS (2008 – 2010) BY THE DEPARTMENTS SAMPLED

S/No	Equipment	No. bought by sampled departments in Kwara Polytechnic					No. bought by sampled departments in University of Ilorin				
		Year 2008	Year 2009	Year 2010	Year 2011	Total	Year 2008	Year 2009	Year 2010	Year 2011	Total
1	Desktop computers	4	29	5	23	61	6	9	7	3	25
2	Laptop/Notebook computers	3	1	10	8	22	7	3	86	1	97
3	Monitors (CRTs)	6	4	4	1	15	3	5	56		64
4	Flat screens (VCDs)	1	23	1	22	47		4	52	3	59
5	Printers	6	3	7	7	23	6	5	5	6	22
6	Mobile phones	16	5		16	37				2	2
7	Radio sets	5	1	1	5	12			1	3	4
8	Televisions	2	3	3	2	10	4	2	12	1	19
9	Fridges	10	4	1	3	18	3	2	4	3	12
10	Air conditioners	5	4	3	2	14	10	37	37	24	108
11	Photocopiers	2	1		2	5	2	2	7	2	13
12	Fax Machines		1			1		1		1	2
13	UPS	7	1	4	44	56	1	4	81	2	88

TABLE 4: SOURCE OF ACQUIRED EQUIPMENT IN SAMPLED DEPARTMENTS

S/No.	Source of equipment	Kwara Polytechnic		University of Ilorin	
		N	%	N	%
1	Retail outlet or store	11	64.1	12	66.7
2	General distributor	3	16.7	6	33.3
3	Leased	0	0.0	0	0.0
4	Formal second-hand market	1	5.6	0	0.0
5	Informal second-hand market	1	5.6	0	0.0
6	Others	2	11.1	0	0.0
	Total percent		100.0		100.0

TABLE 5: NUMBER OF DEPARTMENTS OUT OF 16 DEPARTMENTS AT EACH INSTITUTION SELECTING AN INDICATED METHOD OF E-WASTE DISPOSAL

S/No.	Equipment	Kwara Polytechnic					University of Ilorin				
		A	B	C	D	E	A	B	C	D	E
		Store away in the dept./unit	Throw away with other waste - discard	Send them to the Stores department	Disassemble to reuse some parts to repair others	Others	Store away in the dept./unit	Throw away with other waste - discard	Send them to the Stores department	Disassemble to reuse some parts to repair others	Others
1	Desktop computers	1		6	1		7		7		
2	Laptop/Notebook computers	1		3			2		2		
3	Monitors (CRTs)			4			5		4		
4	Flat screens (VCDs)			3			3		1		
5	Printers	2		6			7		3		
6	Mobile phones			1	1		3				
7	Radio sets	3		2			2				
8	Televisions	1		3			4		1		
9	Fridges	2		3			5		2		
10	Air conditioners	3		2			4		3		
11	Photocopiers	3		2	1		4		2		
12	Fax Machines			2			3				
13	UPS	3		2			6		1		

Two major sources for acquisition of electronic and electrical equipment were identified as (i) purchase from a retail outlet or store and (ii) purchase from a general distributor. At both organisations more than 60% of the distributors (33%). On the other hand, while 64% of the departments sampled at Kwara Poly sourced electronic equipment from retail outlets or stores, 17% sourced from general distributors while one department each indicated sourcing equipment from the formal and informal second-hand market. Of the two departments at the Kwara Poly that indicated 'Others' as source of equipment (Table 4) one indicated a supply source as being from the school store while the other indicated the Education Trust Fund (ETF) departments sampled reported obtaining electronic equipment from retail outlets or stores (Table 4). At Unilorin, equipment was sourced from either retail outlets or stores (67%) or from general Intervention as a source.

It was also established that there were two main disposal methods used by the departments sampled at the University of Ilorin. These were to either store the unused equipment away in the department or send them to the Stores department (Table 5). These were the disposal modes most used also at Kwara Poly, except that a few departments indicated disassembling the equipment in order to reuse parts therein to repair desktops, mobile phones and photocopiers. However, fewer

departments at Kwara Poly kept out of use equipment at their departments than did departments at Unilorin as shown in Table 5.

On the question of whether a department kept records of discarded e-waste, very few responses were obtained, most of which was 'No'. The large proportion of no responses to this data item further re-affirms the finding that at both institutions, e-waste is rarely discarded at the departmental level (Table 6), and shows adherence to due process in fixed-assets management.

TABLE 6: RESPONSE TO KEEPING INVENTORY OF DISCARDED ITEMS

Keep inventory of discarded items	Kwara Polytechnic		University of Ilorin		Chi-square	df	p
	n	%	n	%			
Yes	0	0.0	1	6.3	1.293	2	0.524
No	4	25.0	5	31.3			
No response	12	75.0	10	62.5			
Total	16	100.0	16	100.0			

The Chi-square statistic (1.293 with calculated probability of 0.524) obtained from the cross-tabulation of responses to whether inventory of discarded items were kept (Table 6) suggests that at the 0.05 level of significance, institution was not a factor in whether or not a department would keep records of discarded equipment.

To the question on whether the department kept an inventory of stored out of use items, the proportion of departments that answered 'Yes' was 44% for Kwara Poly and 63% for Unilorin (Table 7). From the Chi-square statistic of 1.348 with calculated probability of 0.510 (Table 7), it was inferred that institution was not a factor in whether or not a department would keep records of stored out of use items.

TABLE 7: RESPONSE TO KEEPING INVENTORY OF STORED ITEMS

Keep inventory of stored out of use items	Kwara Polytechnic		University of Ilorin		Chi-square	df	p
	n	%	n	%			
Yes	7	43.8	10	62.5	1.348	2	0.510
No	2	12.5	2	12.5			
No response	7	43.8	4	25.0			
Total	16	100	16	100			

At the Stores department of the Kwara Polytechnic, it was gathered that after receiving goods from departments across the polytechnic, an attempt was usually made to reuse some parts to repair others. This was the case for desktop computers, laptops, monitors (CRTs), flat screens (VCDs) and printers. Televisions, refrigerators, air conditioners, photocopiers, fax machines and UPS were also for this purpose. In addition to reuse for other repairs, the Polytechnic periodically auctioned electronic equipment at its end of life. Nevertheless, most of the e-waste still remained in the warehouse of the Stores department.

On the other hand, at the University of Ilorin Stores department, e-waste was disposed of in the following ways –

(a) Moved to another department or unit within the university that can still use the equipment
(b) Sold to the university staff at reduced price
(c) Donated
(d) Sold to other organisations/ firms as second hand equipment
(e) Auction the equipment.
(f) Recommending items for final disposal was done at the University of Ilorin through a 'Board of Survey' constituted by university management.

Just as in the case of Kwara Poly, it was also found that despite the efforts at discarding the e-waste, most of it still remained stored in the various warehouses of the Stores department at Unilorin.

Conclusions & Recommendations

Several conclusions can be reached as a result of this study. One is that most of the out of use electronic and electrical equipment (e-waste) still remained stored at the Stores department of both institutions which provides a pointer to answering the question earlier posed – what has happened to all the e-waste generated since inception of both institutions? The answer may be that most of the e-waste generated can largely be found in the warehouses

of the Stores departments – a 'do nothing approach' suggesting the need for a policy to guide real disposal that leads to emptying the warehouses. Furthermore, the findings suggest that improvements in inventory management are required at both institutions because not all departments in the sample kept records of out of use equipment.

There are lessons on maintenance of electronic and electrical equipment (EEE) that can be drawn from the findings. For instance, the lower proportion of EEE not in working condition that was found at Kwara Poly may be linked to the practice of dissembling to reuse some parts to repair others, and may be especially so for desktop computers. This is a type of waste minimisation. The University of Ilorin should consider adopting and institutionalising this technique as part of the disposal approaches to be explored by the Stores department. In practical terms, the Stores department could work with the University's Equipment Maintenance Centre to achieve this.

Another conclusion reached from the study is that the dominant source for acquiring EEE at both institutions is from retail outlets and stores. This has implications for green disposal as retailers in Nigeria are not known to offer reuse or recycling options. It is best if both institutions identify general distributors that offer reuse, recycle or disposal options. In the long run, using such distributors would reduce purchase costs and free up space in the Stores departments.

For future work, a study that focuses on the Stores department of public institutions and organisations is suggested since most of the e-waste generated would end there. Extending the study to cover other types of large organisations and individual consumers is also suggested.

References

Aginam, E. (2008, October 15). Challenges of e-waste management in the country's market. Vanguard newspaper. Retrieved from http://allafrica.com/stories/2008 10150249.html

Alake, T.J., & Ighalo, G.I. (2012). End of life strategies for effective electronic waste management in Nigeria. *International Journal of Scientific & Technology Research*, 1(7), 73-76.

Bandyopadhyay, A. (2008). Indian initiatives on e-waste management – A critical review. *Environmental Engineering Science*, 25(10), 1507-1526.

Bossuet. L. (2014). Sustainable electronics: On the trail of

reconfigurable computing. *Sustainable Computing: Informatics and Systems*, 4(3), 196-202.

CEPIS. (2012). Energy efficient enterprise in Europe: Green ICT awareness in organisations. Council of European Professional Informatics Societies (CEPIS). Retrieved from http://www.cepis.org/media/GreenICTSurveyReport_v1.20131.pdf

Chen, A., Dietrich, K. N., Huo, X., & Ho, S. (2011). Developmental neurotoxicants in e-waste: An emerging health concern. *Environmental Health Perspectives*, 119(4), 431-438.

Deathe, A. L. B., MacDonald, E., & Amos, W. (2008). E-waste management programmes and the promotion of design for the environment: Assessing Canada's contributions. *RECIEL*, 17(3), 321-336.

Fagbohun, O. (2011). An overview of Nigeria's regulatory approach. Paper presented at 2-Day International Summit on Regulations & Management of e-Waste in Nigeria, Lagos, 24-25 February, 2011. Retrieved from http://elri-ng.org/E-waste%20Summit%20Presentation-LAWMA.pdf

Iyer, L.S. (2014). A study on the attitude towards e-waste collection and safe management in academic institutions in Bangalore. *Asian Journal of Research in Social Sciences and Humanities*, 4(5), 97-112.

Jackson, J. (Ed.) (2007). Study of life expectancy of home components. Retrieved from http://www.nahb.org/fileUpload_details.aspx?contentID=99359

Jenkin, T.A., Webster, J., McShane, L. (2011). An agenda for 'green' information technology and systems research. *Information and Organization*, 21, 17–40.

Leyden, J. (2009). Trade in second hand Blackberries booming in Nigeria. Retrieved from http://www.pronetworks.org/index.php/content/post/trade_in_secondhand_blackberries_booming_in_nigeria/

Man, M., Naidu, R., & Wong, M.H. (2013). Persistent toxic substances released from uncontrolled e-waste recycling and actions for the future. *Science of The Total Environment*, Volumes 463–464 (October), 1133-1137.

Monika. J.K. (2010). E-waste management: As a challenge to public health in India. *Indian Journal of Community Medicine*, 35(3), 382-385.

Needhidasan, S., Samuel, M., & Chidambaram, R. (2014). Electronic waste – an emerging

threat to the environment of urban India. *Journal of Environmental Health Science and Engineering*, **12**:36. doi:10.1186/2052-336X-12-36

Ogungbuyi, O., Nnorom, I.C., Osibanjo, O. & Schluep, M. (2012). E-waste country assessment Nigeria: e-waste Africa project of the secretariat of the Basel Convention. Retrieved from http://www.basel.int/Portals/4/Basel%20Convention/docs/eWaste/EwasteAfrica_Nigeria-Assessment.pdf

Onderi, V.N. (Ed.) (2011). *Guidelines for e-Waste Management in Kenya.* Nairobi, Kenya: NEMA/MEMR.

Oresanya, O. (2011). E-waste management in Lagos State: The LAWMA experience. Retrieved from http://elri-ng.org/E-waste%20Summit%20Presentation-LAWMA.pdf

Osibanjo, O. & Nnorom, I. C. (2007). The challenge of electronic waste (e-waste) management in developing countries. *Waste Management & Research*, 25(6), 489-501. Doi:10.1177/0734242X07

Oteng-Ababio, M. (2010). E-waste: an emerging challenge to solid waste management in Ghana. *International Development Planning Review*, 32(2), 191-206. doi:10.3828/idpr.2010.02

Ramachandra, T.V. & Saira V.K. (2004). *Environmentally sound options for e-waste management. Envis Journal of Human Settlements*, March, 1-8. *Retrieved from* http://www.ces.iisc.ernet.in/energy/paper/ewaste/ewaste.html

Schoenung, J. M., Ogunseitan, O. A., Saphores, J. M., & Shapiro, A. A. (2005). Adopting lead-free electronics: Policy differences and knowledge gaps. *Journal of Industrial Ecology*. 8(4), 59-85.

Schwarzer, S., De Bono, A., Giuliani, G., Kluser, S., & Peduzzi, P. (2005). E-waste, the hidden side of IT equipment's manufacturing and use. *UNEP DEWA/GRID-Europe Environment Alert Bulletin*, 5. Retrieved from http://www.grid.unep.ch/product/publication/download/ew_ewaste.en.pdf

Sthiannopkao, S., & Wong, M.H. (2013) Handling e-waste in developed and developing countries: Initiatives, practices, and consequences. *Science of The Total Environment*, 463–464 (1), 1147–1153.

The Basel Convention at a glance. (n.d.). Retrieved from http://archive.basel.int/convention/bcPoster.pdf

The Green IT Review (2010). Green ICT in the UK will be impacted

by the government's 'fundamental' waste management review. Retrieved from http://www.thegreenitreview.com/2010/06/green-ict-in-uk-will-be-impacted-by.html

UNEP. (2007). E-waste manual: Volume 1. Retrieved from http://www.unep.or.jp/ietc/Publications/spc/EWasteManual_Vol1.pdf

United States Agency for Natural Resources. *(2004). Electronic waste* m*anagement in Vermont.* Retrieved from http://www.anr.state.vt.us/dec/wastediv/recycling/pubs/Electronics_Legislative_Report_gl.pdf

USEPA. (n.d.). Electronic waste and e-cycling. United States Environmental Protection Agency. Retrieved from http://www.epa.gov/region1/solidwaste/electronic/index.html

Waema, T., & Mureithi, M. (2011). *E-waste management in Kenya.* Retrieved from http://ewasteguide.info/files/Waema_2008_KICTANet.pdf

Wang, F., Kuehr, R., Ahlquist, D., & Li, J. (2013). E-waste in China: A country report. StEP Green Paper Series. Retrieved from http://isp.unu.edu/publications/scycle/files/ewaste-in-china.pdf

Yousif, M. (2009). Towards Green ICT (Keynote). *ERCIM News*, 79, October. Retrieved from http://ercim-news.ercim.eu/en79/keynote/664-keynote-towards-green-ict

Towards the Design of a Synchronous Virtual Learning System

Olayiwola W. Bello[1],

David N. Ehira[1],

James K. Ayeni[2],

Nasir Faruk[3], &

Naeem A. Balogun[1]

[1]Department of Information and Communication Science,
University of Ilorin, Ilorin, Nigeria.
[2]Department of Computer Science,
Kwara State Polytechnic, Ilorin, Nigeria
[3]Department of Telecommunication Science,
University of Ilorin, Ilorin, Nigeria.
laibello@unilorin.edu.ng,
ehiradavid@gmail.com ,
kennybetty2006@gmail.com ,
faruk.n@unilorin.edu.ng and neembaloo@yahoo.com

***Abstract*—** The field of education has undoubtedly been affected by the penetrating influence of information and communication technology, characterized by improved access to the internet, and the increasing use of computing devices. However, education in Africa generally and specifically in Nigeria and other developing countries still face a 21st Century challenge in making education available and accessible to all. To this end therefore, this paper presents a conceptual approach, as to how tailor made e-learning services could be realized and integrated with a real time video conference server and any existing learning management system in order to facilitate a synchronous virtual learning environment in making education accessible and available to both remote students (distance learning students) and onsite users in Universities and other related educational institutes. It proposes a functional framework to exemplify educational services such as file sharing to enhance collaboration, a digital resource center for retrieval of both free and paid relevant academic resource. A conference room for real time classroom participation which learning platform should provide in order to enhance both teaching and learning performance of course instructors and their students respectively is also proposed. It provides an operational design which describes how custom made e-learning portal integrated with an Open source Video Conference server could be realized, in facilitating a synchronous virtual learning service. Furthermore, it proposes a Virtual Learning Network architecture to show how both remote and onsite students could optimize quality network access in realizing these electronic learning services.

Keywords/Index Terms— Virtual Learning, Distance Learning, Synchronous, Open Source, Real time, Conference server

Introduction

As universities and colleges increasingly embrace new technologies and leverage them not only to enhance their traditional curriculum, but also to extend course offerings beyond the college campus, opportunities for increased collaboration between students and lecturers have been greatly enhanced (Buzzard et. al., 2011). The evolvement of new way of learning in today's environment is increasing rapidly. Methods through which teaching and learning take place are gradually changing. The new ways of technological usage in the educational system has indeed open a new vista in the teaching and learning sphere. With the advent of learning technologies, the educational paradigm has experienced a shift from the use of conventional classroom materials such as pencils and notepads, to the use of tablets, palmtops, interactive white boards and personal computers for resource sharing, learning and collaboration between students and their instructors("Teaching in a Participatory Digital world", 2013). According to Zafra et. al., (2011), the educational reform in higher institution had witness a considerable increase in the usage of these systems. All these devises riding on the Internet has provided educators at all levels, including universities, a range of innovative and potentially empowering tools for teaching, learning, course management, and assessment. It has prompted significant changes in teaching and learning approaches. This is achieved through its affordances, the growth of content it provides, and also new technologies that promote and support communication and interaction between and among the students, the content providers, and the instructors (Ahamad and Bokhari, 2011). It is however not surprising, that there is a continuous increase in the use of these technologies in the educational sector, as most of this technological artifacts has its underlying bearing from the educational stream. This paradigm of electronic learning has however metamorphosed into what is known today as Synchronous Learning Systems. This type of communication is typically used for delivering lectures, answering questions and receiving feedback from the students in real time. If the teacher needs to send out an assignment, he or she can simply post a bulletin on the class website, which the students also receive via an e-mail notifying them of the new assignment. Synchronous learning is done in real-time with a live instructor facilitating the training. Everyone logs in at a set time and can communicate directly with the instructor and with each other. It is usually fixed for a specific amount of

time, from a single session to several weeks, months or even years. This type of training usually takes place via Web sites, audio- or video-conferencing platforms, Internet telephony, or even two-way live broadcasts to students in a classroom (Skold, 2012). Furthermore, the physical distance that is often seen by many as a disadvantage can have positive effects by making learners and instructors more objective, less fearful of comment or criticism, less prone to cultural barriers (Praslova, 2004).

This work therefore is poised to the realization of a virtual learning environment adopted to solve the challenges faced by academic institutions in Africa and other developing countries, so as to bridge the gap of limited infrastructure and human resource and inadequate academic resource. In another vein, this work could go a long way in making room for inclusive education. This is to mean that, universities would now be able to admit more without having to consider infrastructural capacity. As posited by Praslova et. al., (2006), post-secondary education needs to prepare to build programs for millions of emerging open learning students as it is anticipated that approximately 150 million more online learning spots will be needed in the next decade.

II. Problem Definition

Due to the escalating number of students seeking admission to Nigerian Universities and the limited classroom infrastructure to match, there has always been a challenge to effective dissemination of lectures in such condition, characterized by having overcrowded classrooms. In 2013, out of the 1.7 million students that sat for University Tertiary Matriculation Examination (UTME), less than 500 thousand can be accommodated due to limited space, this is according to the Minister for Education in Nigeria (Dailypost, 2013). Also, the situation of having different lecturers take the same course due to having very large classes divided into groups leads to disparity in the quality of knowledge delivered, which consequently affect students performance. It is also a common practice enshrined in schools' policy that lecturers are granted study leave to further their studies for a period of time, however plans are not made to absorb more instructors thus leading to reduced man power for the time in question. Furthermore, distance learning students, suffer from inadequate interaction, close collaboration and supervision with their course lecturer since the interaction between them is not in real time.

III. Goal

The proposed Synchronous Virtual Learning Environment will facilitate real time learning experience between distance learning and onsite

students in Nigerian universities and her lecturers. It would help maximize human resource especially when the available instructors are gone for further studies for career development. Additionally, it would ensure collaboration and quicker access to relevant academic resource and also has the propensity to promote knowledge delivery from scholars across the world.

IV. Design Consideration

Several applications are available for deploying a Virtual Learning Environment and they come in different forms and flavours. It could be proprietary, open-source or self-designed. However it has been discovered that information systems are more successful, when they are tailor made to meet the requirements of a particular environment (Reeves and Minocha, 2010). According to Francese et. al. (2006), certain requirements should be put in place in the development of information systems in Africa. These include sustainability, affordability, socio-economic justification and community participation. However the case is not the same when deploying proprietary software, because its design concept is built on a set of requirements relative to the working environment that its designer had in mind. Although each proprietary software comes with its own peculiarity, it however comes with a cost. The cost ranges from the cost of its acquisition and deployment (financial cost), to the cost of maintenance and technical support from its developers. The case is not the same with self- designed applications as it affords its users to make use of applications that were developed with the peculiarity and system requirements of their environment in mind. Since it is custom made, it also makes it scalable for future advancement. It is on this premise that this paper proposes that a self-designed application be built in order to meet the peculiarity of Nigerian Universities.

V. Theoretical Framework

Virtual Learning Environments can be defined as an integrated digital environment driven by computer assistive technologies in order to support teaching, learning, academic administration and research in a collaborative learning environment (Salmon, 2009). They are often referred to as online learning environments, learning management systems or collaborative learning software that extends the traditional classroom systems by implementing a non-geographically biased classroom infrastructure. Often embedded online, but sometimes distributed as a software package (stand-alone application), virtual learning environments are “boxed” system that mediates between teachers and students. Yang (2010),

described the use of traditional classroom and technology in a half manner as "hybrid virtual-physical collaboration learning", in which he consider it not to do away with the traditional learning while virtual learning complement it. Access to the learning environment is often ubiquitous and supports nomadic learning; which implies lecturers and students alike can access it anytime and anywhere they have an internet connection or are within the virtual learning network (Deutschmann, 2009). Using this learning platform, new knowledge is established by learners during communication and interaction with learning materials, colleagues, and instructors (Lai, 2010). In a virtual classroom system, the teacher may communicate with the students in real-time using video or Web conferencing facilities. This type of communication is typically used for giving lectures and for question and answer sessions (Pankaja and Mukund (2013). Virtual communities within the eLearning environment might have the ability to rectify the issues faced by eLearning (Lai, 2010). The principal components of a Virtual Learning Environment package include curriculum mapping (breaking curriculum into sections that can be assigned and assessed), student tracking, online support for both teacher and student, electronic communication (e-mail, threaded discussions, chat, Web publishing), and Internet links to outside curriculum resources (Mursu and Olufokunbi, 2002).

VI. Related Works

Multiple studies in the area of designing virtual learning environments have arrived at strikingly similar conclusions regarding the major educational benefits of purposefully designed virtual learning environments (Ssekakubo, et al., 2011). In other words, virtual environments designed to accommodate a specific learning activity, may have positive effects on learning if it is compatible with the educational activity that takes place within the given academic institution. Conversely, a virtual learning environment that is ill–suited for a specific task might have adverse effects on the learning performance of students utilizing the platform (Meloni, 2010).

Following the line of reasoning, as cited by Ssekakubo et al., (2011), the architectural design of a virtual campus should manifest the pedagogy–related features of a given university in order to encourage a desired educational approach among students and staff.

However, it is important to draw a line of disparity between what type of virtual learning system an entity is, in the context of its design and use. While some are designed to

facilitate asynchronous communication between students and lecturers, classified as the learning management systems, others are designed to facilitate synchronous (real time) communication classified under the category of video conference servers.

A learning management system as cited by (Robert, 2014) is defined as a software application or Web-based technology used to plan, implement, and assess a specific learning process, while Synchronous e-learning involves online studies through chat and videoconferencing done in real time (Mindflash, 2014). Either of the aforementioned systems could be used as e-learning platforms depending on its context of use.

Konstantinidis and Tsiatsos (2008), focused on the computer supported collaborative learning virtual environment, to which a virtual world was created to teach students. However, users' difficulties were experienced while examining the tools and services.

Sawyerr and Hobbs (2014), worked on creating a means to strengthen condition of virtual environment applications within requirement engineering activities and later stage of development. Their research was due to issues of usability of 3D virtual environment user interface and designing 3D virtual environment for limited access groups. Sawyerr and Hobbs (2014), later observed that, what was referred to as 'designing virtual environments' was actually implementation and later deployment of the platforms. However, their research addresses formalising the characteristics of virtual environment applications and user interface by using a generic software development approach so as to have a reference point that leads the development process. This was done in order to formally treat it as the software development like other platforms such as mobiles and computers. In their work, they suggest more effort should be on design so that usability of virtual environment application can be improved and reduced usability problems usually found after deployment has taken place.

University of Nairobi has implemented three different learning management systems in the last five years: Wedusoft, Chisimba and Claroline. Wedusoft was specifically developed by a member of staff for the university while Chisimba was adopted and implemented through collaboration with development partners. Currently, the university is using Claroline LMS. However, none of the LMS have been utilized to their potential, and the success of LMS-supported e-learning at the university is described as minimal (Ssekakubo et al., (2011).

Currently, the University of Cape Town is using Sakai as the major LMS. This has been customized and branded Vula. In the past, the university has deployed Moodle and WebCT as well. However, they still continue to seek for virtual learning platforms that would satisfy most of their requirements (Ssekakubo et. al., 2011).

Share-point, a Microsoft content and document management system was used at, Nelson Mandela Metropolitan University to make courses available for sharing and collaboration in a blended environment. However, the platform was found to be less flexible, and had limited interactivity options. As a result it migrated to Moodle, and currently uses it as its Learning management System (Ssekakubo et. al., 2011).

In the University of Ilorin Moodle (a learning management System) was deployed in 2013 to facilitate learning and correspondence between the lecturers of university of Ilorin and its open distance learning students (Post-Doctoral Diploma in education). Via this platform lecturers can post lecture notes, send assignments, information notice, grade students and send results to every registered student on the platform. Although this platform has been in used for a while now, it has not been fully maximized and explored to cater for other courses in the open distance learning sector. More importantly it does not yet support a real time communication between the lecturers and the students (Center for Open and Distance learning-University of Ilorin, 2013).

From the generally held positions and observations described above, universities in Africa have invested in the use of LMSs like Moodle, Sakai, Blackboard etc. These LMSs has only been successful in maintaining the learning process between the students and their lecturers; but with little or no provisions and capacity to solve peculiar challenges such as overcrowded classrooms, and limited human resources as experienced within the Nigerian academic system. Hence, the need to have self-designed learning systems which takes into cognizance the peculiar challenges faced within an instructor-led classroom in the Nigerian context.

VII. Methodology

The Current Learning System in Nigerian universities assumes the following pattern. For regular students, at scheduled time, students of a particular course meet with the assigned lecturer to receive the course instruction within the confine of classroom structure at predefined period. However the presentation style is relative to the lecturer's choice of delivering his lectures. He

could deliver his lecture via the white board, with the use of a board marker and eraser or make a projection of his lectures via a projector from a well prepared power point slides. However for its remote students(Distance learning students) the assigned lecturer for the given course could interact with his students via any learning management system like Moodle, Sakai, Blackboard, Canvas, OLAT and the likes; via which course interactions is facilitated between the students and the lecturer.

A. *Conceptual Design*: the entire concept of the proposed e-learning system is based on three major sub-systems. These are Video Conference Server, Resource Center and Potal System sub-systems. For each of these subsystems, requirements will be gathered, its design and implementation will be carried out, and the subsystems will be evaluated and tested. All these activities are designed to be carried out in iterative order. The successful integration of of theses three will give rise to the conceived system. This is as presented in Figure 1.

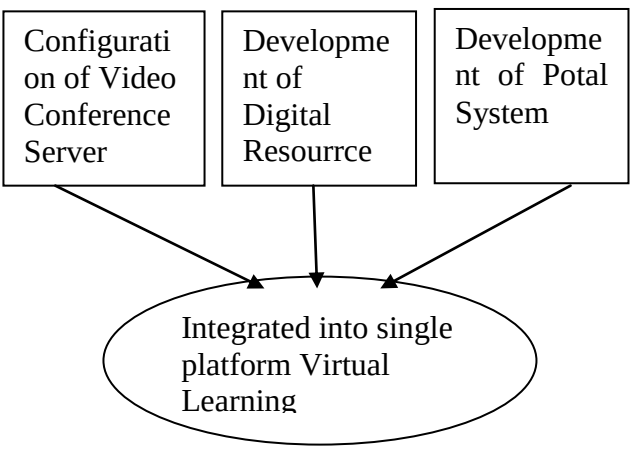

Figure 1: Conceptual Design of the Proposed System

B. *Functional Design***:** The proposed e-learning system could be designed as a web based e-learning portal to reflect custom classroom requirements as exemplified in figure 1. When adapted with BigBlueButton (which is an open source video conference server), it provides synchronous communication between students and lecturers. Also it could be integrated with any Learning management system, in order to enhance the

learning process. Functionally, the services provided by the entire system are as depicted in the Figure 2.

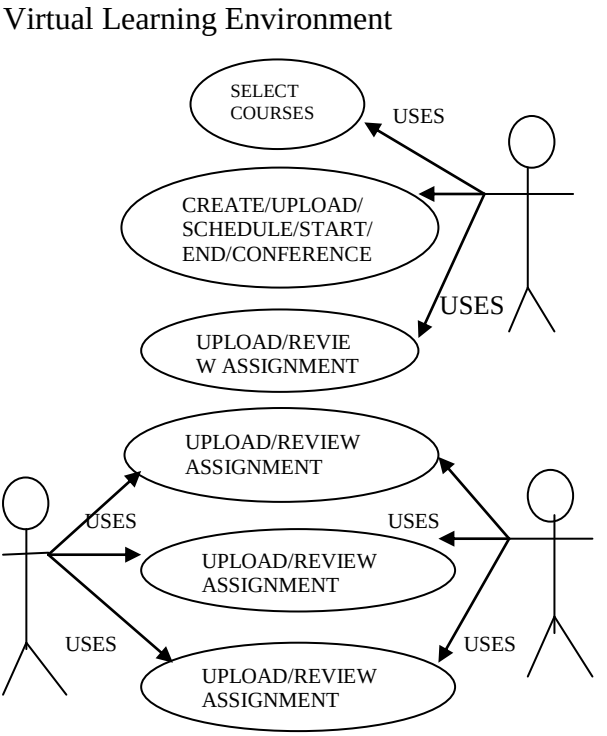

Figure 2: Use Case Diagram for the Functional Architecture of the Proposed System

As expressed in the functional design, the new system would allow lecturers to create assignments, share files, schedule and attend conferences and download academic resources from the resource center.

C. *Operational Design*: The system is primarily proposed for lecturers and students in Nigerian Universities. The Virtual Learning system would be made to extend the existing portal system of the University; hence both lecturers and student can use their existing login details to gain access into the Virtual learning portal. However all lecturers would be required to dynamically add all the courses they would take for each academic session. For the students, courses are dynamically added, after courses registration on the existing school portal. To have live classes the lecturer would have to open the conference room for the students to join. The conference room

utilizes BigBlueButton for its real time instructor-led learning process.

.

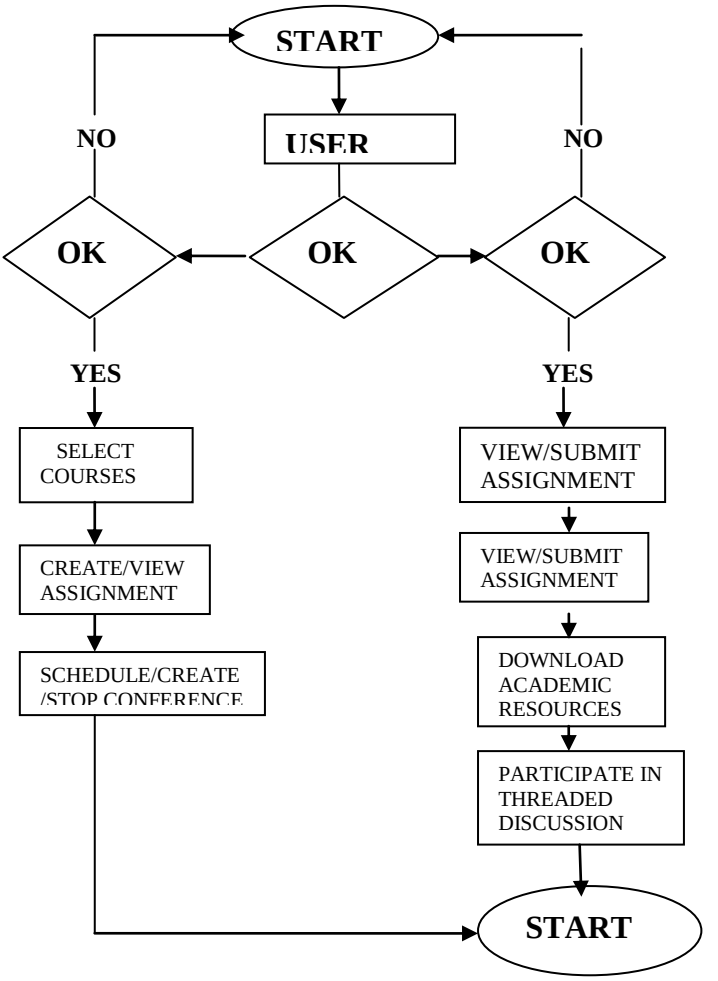

Figure 3: Flow Chart of the Proposed System

D. *E-Learning Network Architecture*

As part of the conceptual design, a virtual learning network architecture is proposed for Nigerian Universities to support both onsite users (Regular students within the campus) and remote users (Distance learning students). Thus a Video conference server (BigBlueButton), configured with a live IP address is to be hosted on a dedicated server at the campus network operating center, in order for both onsite and remote users to gain access to the e-learning system.

To compensate for Quality of Service metric deficiencies, universities deploying such services should make provision to facilitate high speed fiber optic networks, which enhances efficient propagation of video traffic and enables the local users to enjoy a maximum service level. However, traffic coming from a remote domain to access the e-learning system should be prioritized on the campus core router in order to increase the user's access privilege. Also administrators could configure the quality of video stream on the video configuration module of the video conference server (BigBlueButton) to the barest minimum, in order to help optimize network access in underserved areas with significantly low network bandwidth. Moreover, a link to a short guide could be placed on the landing page of the conference room, instructing the students to reduce the quality of videos being streamed out in such unfavourable network condition. Additionally, it is recommended that TV white space technology - which takes advantage of the available spaces in the UHF and VHF broadcast frequencies, in addition to its ability to penetrate unsuitable terrain environment will be of great advantage (Faruk et al., 2014).

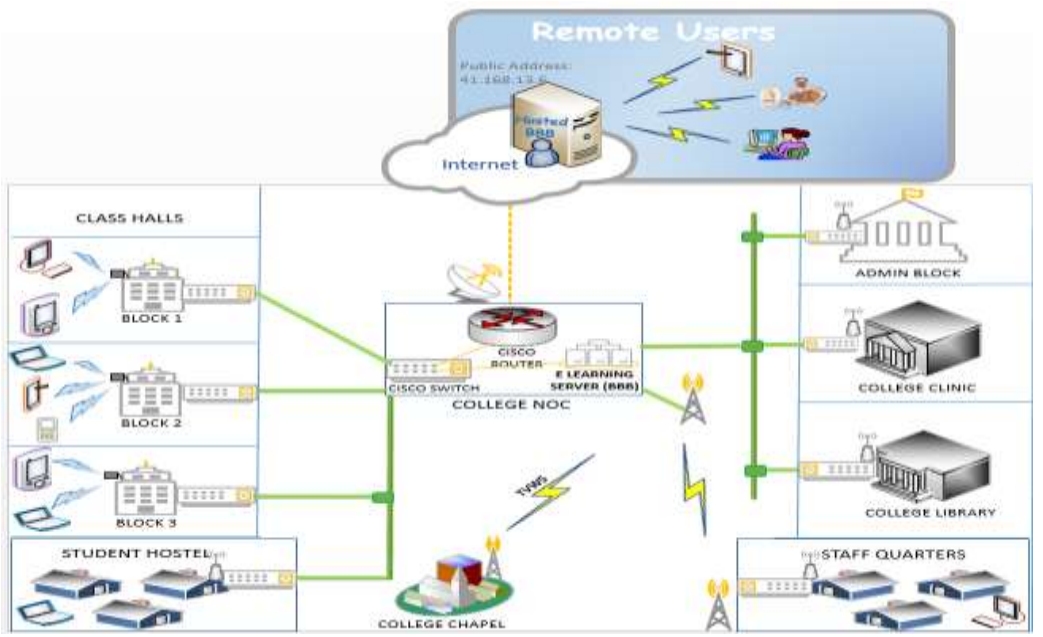

Figure 4. Proposed E-Learning Network Architecture

VI. A Comparative Analysis

In an attempt to make comparison between the proposed system and existing systems, the initial test run

of the conceptualized design, was assessed using the following parameters: Scalability, Deployment, Cost, Software Requirement and Network Requirement. This comparison was made against three top web conferencing software: AdobeConnect, CisoWebEx and DimDim

Scalability: while the proposed learning system can scale to hundreds of participants, depending on the processor and hard drive of the media server. The free version of AdobeConnect, CisoWebEx and DimDim free could scale up to 20, 25 and 10 participants respectively.

Deployment: the proposed system supports both local and cloud hosting of the server, AdobeConnect, CisoWebEx and DimDim supports cloud hosting of its server on their respective media server.

Cost: the proposed system does not require a service charge per user, and it is easy to install and maintain with little or no technical constraint, whereas AdobeConnect, CisoWebEx and DimDim requires service charge per user in addition to deployment and the fee for technical support and maintenance. By the time this cost is figured out in the context of a university, this could be turning to several millions.

Software Requirement: the proposed learning system requires adobe flash to run and can be run all browsers, while CiscoWebEx and DimDim would require certain plugins to be rendered on browsers effectively.

Network Requirement: for the proposed system, onsite users rely on the institutions network to participate in the virtual classroom, while remote students would require internet access, of which there are provisions to help optimize the quality of network access coming from areas with relatively low network bandwidth on BigBlueButton's Video configuration file. Whereas for AdobeConnect, CisoWebEx and DimDim both onsite and remote students would require a relatively good internet service

VII. Conclusion

In this paper, efforts has been made to show the logical structure and feasibility of implementing a synchronous learning system in order to mitigate the challenge of limited classroom infrastructure, inadequate academic resource and limited human resource associated with the Nigerian academic system. It exemplifies the operational design, network architecture and functional model required to implement a synchronous e-learning service, for future deployment. It recommends that tailor made e-learning applications which facilitates real time learning process be built, so that challenges of overcrowded

classrooms and limited human resources could be appropriately dealt with. Furthermore a comparative analysis was carried out to justify the need to have self-designed applications built on an open source video conference server, as opposed to using proprietary softwares. Additionally, lecturers can utilize the desktop sharing functionality to teach practical oriented courses to students, in situation where laboratories have limited facilities. Consequently when this environment is provided to distance learning students, it has the propensity to ensure greater collaboration and commitment between the students and the lecturers.

References

Ahmad, I. and Bokhari, M. (2011). Distance Education: From Asynchronous to Synchronous Elearning Volume 1, No. 1, December 2011 Retrieved on June 17, 2014 from: http://www.cpmr.org.in/CPMR-IJT/vol1/issue1/Articles/2.pdf.

Buzzard C., Criitend, L., Criitend, F., & McCarty, P. (2011). The Use of Digital Technologies in the Classroom: A Teaching and Learning Perspective. Journal of Marketing Education, vol. 33 no. 2 pp131-139. doi:10.1177/0273475311410845. Retrieved on August 10, 2014 from:

Center for Open and Distance learning, University of Ilorin. (2013). Retrieved on 12 June, 2014 from: http://www.uilcodl.unilorin.edu.ng/.

Croft, N., Dalton, A. and M. Grant. (2011). Overcoming isolation in Distance Learning: Building a Learning Community through time and space. CEBE Working paper, Volume 18. Retrieved on 16 June, 2014from :http://www.heacademy.ac.uk/assets/cebe/documents/resources/workingpapers/WorkingPaper_18.pdf.

Dailypost Nigeria Online Newspaper (2013). UTME: Over one million candidates will be denied admission-Minister. Retrieved on November 24, 2013 at http://dailypost.ng/2013/04/28/utme-over-one-million-candidates-will-be-denied-admission-minister/

Deutschmann, L. Panichi, and J. Molka Danielsen, (2009). "Designing oral participation in *Second Life*: A comparative study of two language proficiency courses," *ReCall*, volume 21, number 2, pp. 206-226. http://dx.doi.org/10.1017/S0958344009000196. Retrived on May 17, 2014

Francese R., Passero I., Tortora G. and De Lucia, (2008). "Development and evaluation of a virtual campus on *Second Life*: The case of SecondDMI," *Computers and Education*, volume 52, number 1, pp. 220–233.http://dx.doi.org/10.1016/j.compedu.2008.08.001. Retrieved July 17, 2014.

Konstantinidis, A. & Tsiatsos, Th.(2008). Selecting a Networked Virtual Environment Platform and the Design of a Collaborative e-Learning Environment. Advanced Information Networking and Applications Workshop (AINAW 2008), 22nd International Conference. 44-49.

Lai, H. F. (2010). Determining the sustainability of virtual learning communities in E-learning platform. The 5th International Conference on Computer Science & Education Hefei, China. August 24–27, 2010.1581-1586.

Meloni, J. (2010). Tools for synchronous and asynchronous classroom discussion [online blog]. *The Chronicle of Higher Education*. Retrieved from http://chronicle.com/blogs/profhacker/tools-for-synchronousasynchronous-classroom-discussion/22902 Retrieved on June 2, 2014.

Mindflash (2014). Asynchronous E-Learning Vs. Synchronous E-Learning. Retrieved from: http://www.mindflash.com/elearning/asynchronous-synchronous. Retrived on June 17, 2014.

Mursu, A. & Olufokunbi, K (2002). Information systems development in developing countries. Risk management and sustainability analysis in Nigerian software companies. Citeseer journals doi=10.1.1.127.4743. Retrieved on September 17, 2013.

Faruk, N. Ayeni, A.A. and Adediran, Y.A, (2013)" DTV coverage and protection contour estimation for spatial white space" in Proc IEEE International Conference on Emerging & Sustainable Technologies for Power & ICT in a Developing Society, pp 96-99, Imo, Nigeria July 31st, 2013

Pankaja, N. & Mukund Raj, P.K(2013). Proprietary software versus Open Source Software for Education. American Journal of Engineering Research.Vol.2, issue 7. Retrieved from:http://www.ajer.org. on 23 September 2013.

Praslova, E., (2004). "Virtual spaces as artifacts: Implications for the design of educational CVEs," *International Journal of Distance Education Technologies*, volume 2, number 4, pp. 94-115. http://dx.doi.org/10.4018/jdet.2004100106
Retrieved on November 17, 2013.

Praslova, E., Sourin, A, and Sourina, O. (2006). "Cybercampuses: Design issues and future directions," *Visual Computer*, volume 22, issue 12, pp. 1,015–1,028.

Reeves A. and Minocha, (2010). "Design of learning spaces in 3D virtual worlds: An empirical investigation of *Second Life*," *Learning, Media and Technology*, volume 35, number 2, pp.111–137.http://dx.doi.org/10.1080/17439884.2010.494419.
Retrieved on June 11, 2014

Robert, R. (2014) Tech Terms: Learning management system. Retrieved from: http://searchcio.techtarget.com/definition/learning-management-system.

Salmon, 2009. "The future for (second) life and learning," *British Journal of Educational Technology*, volume 40, number 3, pp. 526–538.http://dx.doi.org/10.1111/j.1467-8535.2009.00967.
Retrieved on May 17, 2013.

Sawyerr, W. and Hobbs, M. (2014). Designing for usability in 3D virtual environments. Usability and Accessibility Focused Requirements Engineering (UsARE), IEEE 2nd International Workshop. Karlskrona, Sweden. 32 – 35.

Skold, O. (2012). Effect of virtual space on learning:A Literature review. First Monday, Volume 17, number1-2 http://firstmonday.org/ojs/index.php/fm/article/view/3496/3133.
Doi:10.5210/fm.v17i1.3496.
Retrieved on December 17, 2013.

Ssekakubo, G., Suleman, H., and Marsden, G. (2011). Issues of Adoption: Have E-Learning Management Systems Fulfilled their Potential in Developing Countries? Retrieved September 19 2013, from:http://pubs.cs.uct.ac.za/archive/00000712/01/p231-ssekakubo.pdf

Teaching in a Participatory Digital World. (2013). Education Canada Magazine. Volume 53, issue 5.

Yang, Z. (2010). Design and Implementation of a Hybrid Virtual-physical Collaboration Learning System: Architecture and Solution. *2nd International Conference on Education Technology and Computer (ICETC), 2*, 402-405.

Zafra, A., Gibaja, E., Luque, M., and Ventura, S. (2011). An Evaluation of the Effectiveness of e-learning system as Support for Traditional Classes. *7th International Conference on Next Generation Web Services Practices.* 431-435.

Implementation of Secured Message Transmission using DES and RSA Cryptosystem

Jelili Oyelade[1,*], Itunuoluwa Isewon[1], Olufunke Oladipupo [1] & Abolaji Famuyiwa [2]

[1]Department of Computer and Information Sciences
Covenant University
PMB 1023, Ota, Nigeria.

[2]Department of Computer Science and Technology
Bells University of Technology
P.M.B. 1015, Ota, Nigeria.

* Corresponding Author; ola.oyelade@covenantuniversity.edu.ng.

Abstract: In the past, Cryptography was used in keeping military information, diplomatic correspondence secure and in protection of national security. Nowadays, the range of cryptography applications have been expanded a lot in the modern area after the development of communication means; cryptography is essentially required to ensure that data are protected against penetrations. This is used to ensure that the contents of a message are confidentially transmitted and would not be altered. In this paper, we have implemented a cryptosystem (encrypting/decryption) for text data using both Data Encryption Standard (DES) and Rivest-Shamir-Adleman (RSA) cryptosystem. The asymmetric algorithm was used for the key encryption and decryption process because it provides a means to deliver keys on a secure channel, while the data to be sent will be encrypted and decrypted using the symmetric algorithm. This system was designed to accomplish a number of security features such as authentication, confidentiality, integrity, and non-repudiation. Also the combination of the speed and strength of the symmetric algorithm with the robustness and key management capability of the asymmetric algorithm, thereby producing an enhanced encryption algorithm and we employed text data as our experimental data.

Key words: Cryptography, encrypion, decryption, cryptosystem

1.0 Introduction

The evolution of the Internet has rendered cryptography more essential and crucial subject in electronic application systems. Except the system is capable of offering some mechanisms to ascertain security services, the system will have difficulties to be acknowledged. More reliable cryptosystems is needed to be recommended and cryptography

being a critical part of today's information systems. Cryptography can be defined as the science of using employing mathematics to encrypt data. It allows us to retain or transfer delicate information across unsafe networks such as the Internet. So that it is made impossible to be interpreted by anyone asides the intended recipient (Schneier and John, 1996). Cryptography is seen as a technological methods that offer security to data being conveyed on information and communications systems. A cryptography system that offers two accompanying functions, decryption and encryption is known as cryptosystem. Cryptosystems utilize encryption algorithms to define the encryption method, the required software components, and the key to implement the encryption and decryption of the data (Schneier and John, 1996). Cryptography techniques are constantly used to secure critical and confidential information against malicious attack from the invaders. There are two major categories of cryptographic algorithms: asymmetric key and sysmmetric key cryptography (Stalling, 2006). There exist various cryptographic methods and algorithms that are well-defined in the literature such as RSA, DES and AES (Schneier and John, 1996).

In the field of cryptography, encryption can be described as the process of altering information (known as plaintext) employing an algorithm (termed a cipher) to make it illegible to anyone apart from those who have unique knowledge, normally represented as a key. The outcome of the method is encrypted information (in cryptography, known as cipher text). The inverse procedure, that is to render the encrypted information legible again is known as decryption, in other words to render it unencrypted (Fouché and Helen, 1956).

Encrypted data transmitted across network guarantees confidentiality, even if it is successfully retrieved from the network by attackers who compromise some security measures, the confidentiality of the file data is maintained, as the data is stored in encrypted format. Encryption is simply a process of keeping data private or confidential.

Encryption has been used ever since by governments and militaries to aid confidential communication and transmission. Presently encryption is often used in safeguarding information in various types of civilian for instance. For instance, according to the account of Computer Security Institute in 2007, 71% of the companies considered applied encryption for most of their data in transit, furthermore 53% employed encryption for most of their data in storage. Encryption is able to safeguard data "at rest", like files on computers and backup drives (for example USB flash drives). In modern times there have been various reports of secret data for example customers' private files

being unprotected via damage or thievery of laptops or storage drives. Encrypting such files at rest aids safeguard them, incase physical security measures collapse. Digital rights management systems which block illegitimate usage or imitation of patent material and safeguard software against reverse engineering are to some extent a discrete instance of applying encryption on data at rest (Fouché and Helen, 1956)

In this work, we implemented a cryptosystem for text document data encryption/decryption by combining the features of both symmetric key and asymmetric key cryptography. Using the combination of symmetric algorithm (public key) and asymmetric algorithm (private key) increases the overall encryption speed and equally provides the same level of security as the asymmetric technique when used alone. Since the resulting system will combine the speed and strength of the symmetric algorithm with the robustness and key management capability of the asymmetric algorithm, thereby producing an enhanced encryption algorithm which is the motivation for this work.

This paper is organized as follows: in the next Section, we give a brief review of related works; Symmetric and Asymmetric Encryption are presented. In section 3, we discussed briefly the algorithms development of both symmetric and asymmetric encription employed in this work. Section 4 discusses the results and discussion and we conclude the paper in Section 5.

2.0 Related work

RSA encryption is most commonly used for the transport of symmetric-key encryption algorithm keys and for the encryption of small data items. But this algorithm is very slow compare to the commonly used symmetric-key encryption algorithms such as DES (Menezes and Vanstone, 1996). In Subasree and N. K. Sakthivel (2010), a Dual-RSA scheme using Chinese Remainder Theorem (CRT) for its Decryption that improved roughly ¼ times faster performance of RSA in terms of computation cost and memory storage requirements was developed. The Omar *et al*.(2012) proposed a framework for the combination of both Symmetric and Asymmetric Cryptographic Techniques for a secured communication.

2.1. Symmetric and Asymmetric Encryption

Encryption is one of the strongest and the safest way in securing data. Encryption systems are divided into two major parts, symmetric and asymmetric. Symmetric encryption is known as secret key or single key, The receiver uses the same key which the sender uses to encrypt the data to decrypt the message,. This system was the only system used before discovering and developing the public key., A safe way of data

transfer must be used to moving the secret key between the sender and the receiver in symmetric encryption. Figure 2.1 shows how the system works. Symmetric encryption occurs either by substitution transposition technique, or by a mixture of both. Substitution maps each plaintext element into cipher text element, but transposition transposes the positions of plaintext elements.

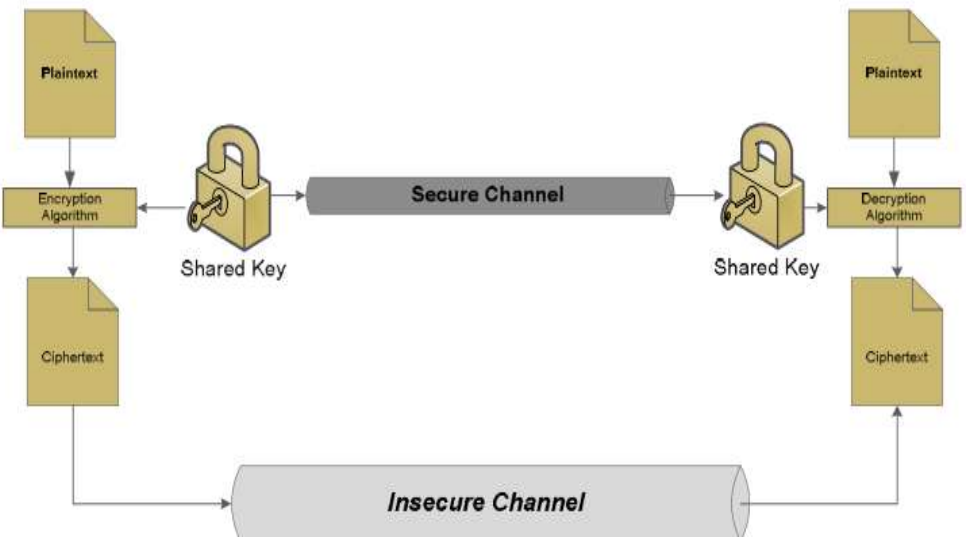

FIGURE .2.1 : Simplified model of conventional encryption (Mohammed *et al.*, 2011)

In 1976 Diffie and Helman invented new encryption technique called public key encryption or asymmetric encryption; Asymmetric encryption is the opposite of symmetric encryption in safety, since it doesn't require sharing the secret key between the sender and the receiver. And this is the main difference between symmetric and asymmetric encryption, the sender has the public key of the receiver. Because the receiver has his own secret key which is extremely difficult or impossible to know through the public key, no shared key is needed; the receiver is responsible for establishing his private and public key, and the receiver sends the public key to all senders by any channel he needs, even unsecured channels to send his public key, asymmetric key can use either the public or secret key to encrypt the data. Also it can use any of the keys in decryption, asymmetric encryption can be used to implement the authentication and non-repudiation security services, and also it can be used for digital signature and other

application that never be implemented using symmetric encryption. Figure.2.2 shows how the system works.

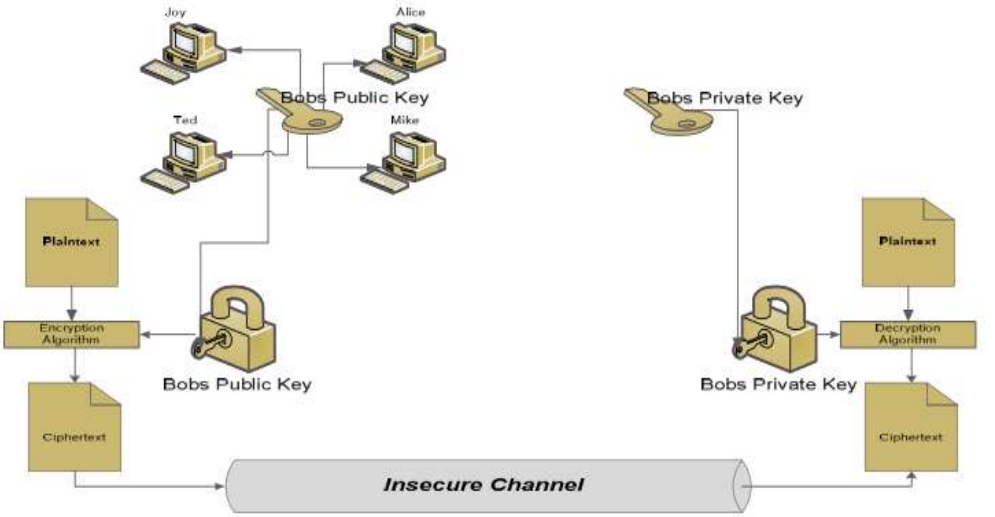

FIGURE 2.2 : Simplified model of asymmetric encryption (Mohammed *et al.*, 2011)

Asymmetric encryption is slower and very complicated in calculations than symmetric encryption. Therefore, asymmetric encryption deals with plaintext as a group of numbers which are manipulated in mathematics, while the plaintext in symmetric encryption deal as group of symbols and characters, the encryption process may permute these symbols, or may substitute one symbol by another. So the nature of the data determines the system of encryption type. And every system has its own uses. For example, asymmetric encryption may be used in authentication or in sending secret key for decryption. Before stated the algorithm in section two, we will explain the following three definitions (Douglas,2006):

Definition 1*:* Let a, n are relatively prime(gcd(a,n)=1), then there is at least one integer m that satisfies $a^m \bmod n=1$. m is referred as the order of a (mod n).

Definition 2: If p is a prime number. An element α having order p-1 is called a primitive element modulo p

Definition 3: Let p be a prime number and α is a primitive element modulo p. any element $\beta \in Z_p$ can be written as $\alpha^i = \beta, 0 \le i \le p-2$ in a unique way i.e., $\alpha^i \equiv \beta (\text{mod}) p$, i is called the unique **discrete logarithm.**

3.0 The Algorithms Development

Since the symmetric algorithm is not suitable for network used by itself unless it is being used with the asymmetric algorithm because of its poor key management technique (Paul, 2004), a combination of both techniques was used in this work. The asymmetric algorithm was used for the key encryption and decryption process because it provides a means to deliver keys on a secure channel (Bruce and John, 1996). While, the data to be sent will be encrypted and decrypted using the symmetric algorithm.

This justifies the selection of the RSA encryption algorithm for the asymmetric technique and the DES encryption algorithm for symmetric technique. In this work, the algorithm design is divided into two parts;

i. The key generation process using the asymmetric encryption technique.
ii. The encrypting and decryption process using the symmetric encryption technique.

3.1. RSA Algorithm (Key generation process using asymmetric encryption technique)

RSA can be described as an Internet authentication and encryption system that applies an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman (RSA). RSA algorithm is known to be the best frequently employed encryption and authentication algorithm. The algorithm makes use of the fact that, it is simple to produce a dual huge prime numbers and multiply them but very tough to determine the product. In the RSA algorithm, designing the key generating technique involves taking a dual grand prime numbers, say p and q that are independently and randomly chosen such that they have about 200 decimal digits (but not less than 150) each. Multiply these two numbers to give a new number, N. Also get another value Q by multiplying $(p-1)$ with $(q-1)$. The mathematical expression is given below:

$$N = p * q$$
$$Q = (p-1) * (q-1) \quad 2.1$$

Next, a random integer e known as the "encryption exponent" is selected between 1 and Q such that gcd (e, Q) = 1 (i.e. the Greatest Common divisor of e and Q). This is expressed below;

gcd (e, Q) = 1 such that **(1 < e < Q)**

Then, using the extended Euclidean algorithm, a unique integer for d will be computed as follows:

$e\,d$ = 1(mod Q) such that **(1 < d < Q)**

This implies that $d = e^{-1} (\text{mod } Q)$

2.2

This generates the public key which is (N, e) and the private key d.

The public and private keys are thus (N, e) and d respectively, where p = the first prime factor chosen, a nonnegative integer and q = the second prime factor chosen, and a nonnegative integer and N is the modulus.

gcd (e, Q) = Greatest Common divisor between e and Q.

e = encryption exponent.

d = decryption exponent.

(N, e) = the public key.

d = the private key.

By making each of the primes about 200 decimals digits long, the product of **p** and **q** can be calculated easily in a fraction of a second. However, factoring **N** is extremely difficult to achieve, even schroeppel, the fastest known algorithm when used would requires billions of years at the rate of one step per microsecond to arrive at the answer (Davies *et al*., 1991). Using a computer might be faster, but for decimal digit of about four hundred (400) and larger, it will run for approximately Ten thousands, one hundred and seventy six (10176) times the life of the universe to determine the product of **p** and **q** supposing a computer can test one million (1,000000) factorizations for every second in the lifespan of the universe (The universe's lifespan is about 1018 seconds; 18 digit number). The use of large primes for **p** and **q** is the strength of this method (Bellare *et al*., 1998).

Therefore, the RSA algorithm steps are stated below:

RSA Algorithm Steps:

- Every user generate a public/private key duo by choosing two huge primes arbitrary p, q
- Computing modular value $n = p * q$
- Calculating the Euler's function $\varphi(n) = (p-1)(q-1)$
- Selecting at randomly the public encryption key e, where $1 < e < \varphi(n)$ and e is prime relative to the $\varphi(n)$.
- Solving the following equation to find private decryption key d:
  - $e * d = 1 \text{ mod } \varphi(n)$. such that $(0 \le d \le n)$
- Publishing their public encryption key: $P_K = (e, n)$
- Keeping secret private decryption key: $P_R = (d, n)$
- At the encryption side the sender uses encryption mathematical equation $C = P^e \text{ mod } n$

- At the decryption side the reciever uses decryption mathematical equation $P = C^d \bmod n$

3.1.1 The key encryption and decryption processes

A Key encryption process

Since we now have our public and private keys, the next step is to encrypt the key.

The key to be encrypted can be represented by **m**, where **m** is an integer in the interval (0, **N**-1). We can calculate the cipher text, **C** (encrypted data format) employing the formula below.

$\mathbf{C} = \mathbf{m}^e \bmod \mathbf{N}$

where **C =** the cipher text (encrypted data) and

m = representation of the key in integer

B. Key decryption process

The recipient gets an encrypted key that is of no value unless it is decrypted. For the original data **m** to be retrieved from the cipher text **C,** The private key is used to perform the decryption: $\mathbf{m} = \mathbf{C}^{\mathbf{d}} \bmod \mathbf{N}$.

The illustration shown below explains the process that occur using RSA algorithm,

Illustration

Let **p** = 2357 and **q** = 2551

N = p * q = 2357 * 2551 = 6012707

Ø = (p - 1) * (q - 1) = (2357 - 1) * (2551 - 1) = 6007800

Choosing e = 3674911 and using the Euclidean algorithm to find **d** we have that;

$\mathbf{d} = e^{-1} \mathbf{(mod\ Ø)} = 3674911^{-1} \pmod{6007800} = 422191$

This generate a public key, **N =** 6012707 and e = 3674911, where **d is** the private key

To encrypt, say message, **m** = 5234673;

The cipher text **C** will be $\mathbf{C} = \mathbf{m}^e \bmod \mathbf{N} = 5234673^{3674911} \bmod 6012707 = 3650502$

To decrypt, the original message is recovered at the recipient end by decryption using the formula below:

$\mathbf{m} = \mathbf{C}^{\mathbf{d}} \bmod \mathbf{N} = 3650502^{422191} \bmod 6012707 = 5234673.$

3.2 The encryption and decryption process using symmetric technique

DES utilities series of procedures involving various substitution and permutation primitives to encrypt a data block. These primitives are later employed to invert the encryption operation. *Horst Feistel algorithm* described a range of substitution and permutation primitives which are repeatedly applied to data blocks for a particular number of times, each set of primitive operations is referred to as a "round". The DES algorithm employs 16 rounds to certify that the data are appropriately scrabbled

to meet up with the security goals. DES is a block product and also a figure 2.1 shows a typical implementation of the DES algorithm.

i. *The Initial Permutation (IP):* This is the initial stage. The 64-bit plaintext is permuted built on an Initial Permutation table, that restructures the bits and generates the permuted input. After IP phase, then the next step which is made up of 16 rounds of corresponding function **F().** The procedures involved in each of the rounds is described by these formulas.

Li = Ri – 1o

Ri = Li – 1 XOR F (Ri -1, ki)

where **Ri** = i numbers of rounds.

Li = i numbers of left circle shift.

Ki = i numbers of permutation choice.

XOR = XOR operation.

F () = function involving both permutation and substitution.

ii. *Permuted Choice One (P1):* The Permuted Choice One (P1) implements permuted choice of 64 bits and returns 56 bits, the remaining 8bits are used for parity (error checking) bit.

iii. *Left Circular Shift (LCS):* The 56-bit output from permutated choice one (P1) is divided into 28-bit block cipher (Shah *et al.*, 2005). The flow chart in blocks each. After having these two 28-bit blocks, the dual now go through a circular left shift of their bits, the number of shifts stated from a list of shifts for each of the rounds.

iv. *Permuted Choice Two (P2):* Following LCS and every one round, a new permuted choice is executed, which leads to the production of a 48-bit sub-key. The *P2* procedure iterates up util sixteen (16) 48-bit sub-keys are generated.

v. *32-bit Swap:* The 64 bits of output from round 16 has left 32 and right 32 bits. These left and right 32 bits blocks are swopped.

vi. *Inverse Permutation (IP^{-1}):* This particular stage is the reverse of the inverse permutation. It gets the input of 64 bits, and alters their sequence again to get a cipher text.

vii. *Encryption and decryption:* DES works on 64-bit "plaintext" data blocks, passing them under the manipulation of a 56-bit key to generate 64 bits of encrypted cipher text as shown in figure 3.1. Likewise, the DES decryption technique runs on a 64-bit cipher text block employing the same 56-bit key to generate the initial 64t plaintext block. This is a reverse of the encryption process (Kavitha *et al*., 2008).

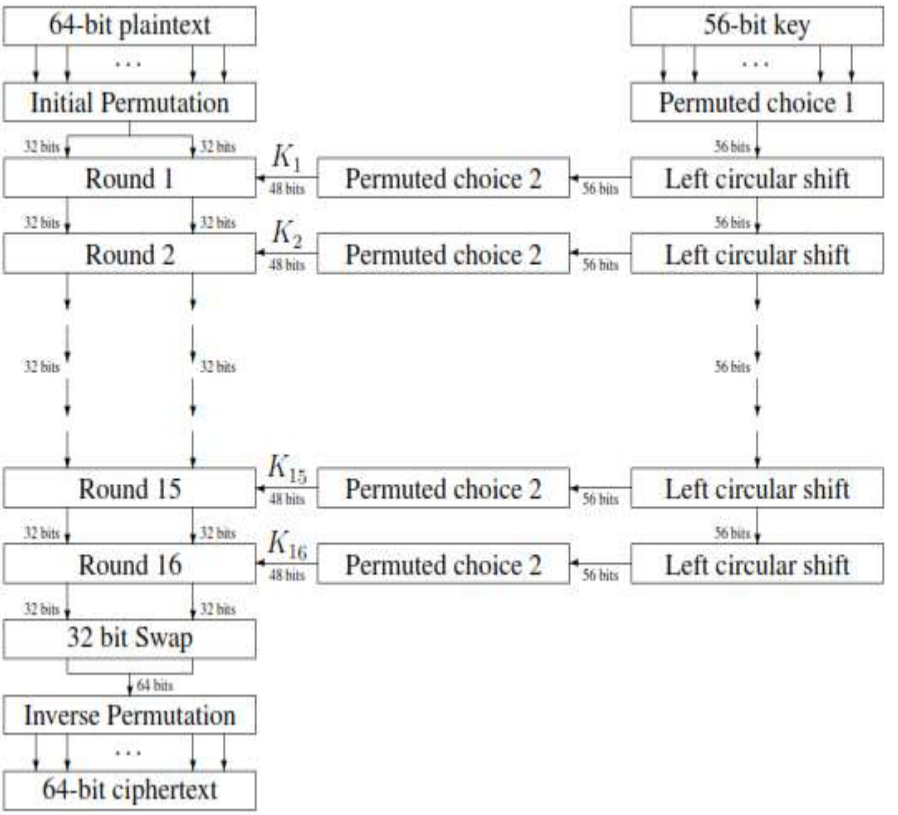

Fig 3.1: General Depiction of DES Encryption Algorithm for encrypting data (Bellare *et al.*, 1998)

4.1 Results and Discussion

4.1.1 Results

The Design implementation for this work is divided into two parts, namely;

The Communication Client implementation

The Communication Server implementation

Both implementations, the Communication client and the Communication server implementation uses Sockets for communication between each other which enables the sending and recipient of data between both ends. The transport protocol used in the implementation design is the Transmission/Transfer control Protocol (TCP). Also, threading and cryptographic function of .NET was used for the implementation. The role of the two implementations is described in the following as follows:

A. The Communication Client Implementation

In this implementation, which is the client side, files are selected, encrypted and sent by specifying the Server`s Internet Protocol (IP) address. The communication client implementation is shown in Fig 4.1 below and its different components explained below;

i. The File Type Combo Box: Here the file type is specified. in this work, the file type can be either .txt or .doc document.

ii. The File Text Box: This displays the name of the selected file using the "browser button" located on its right side.

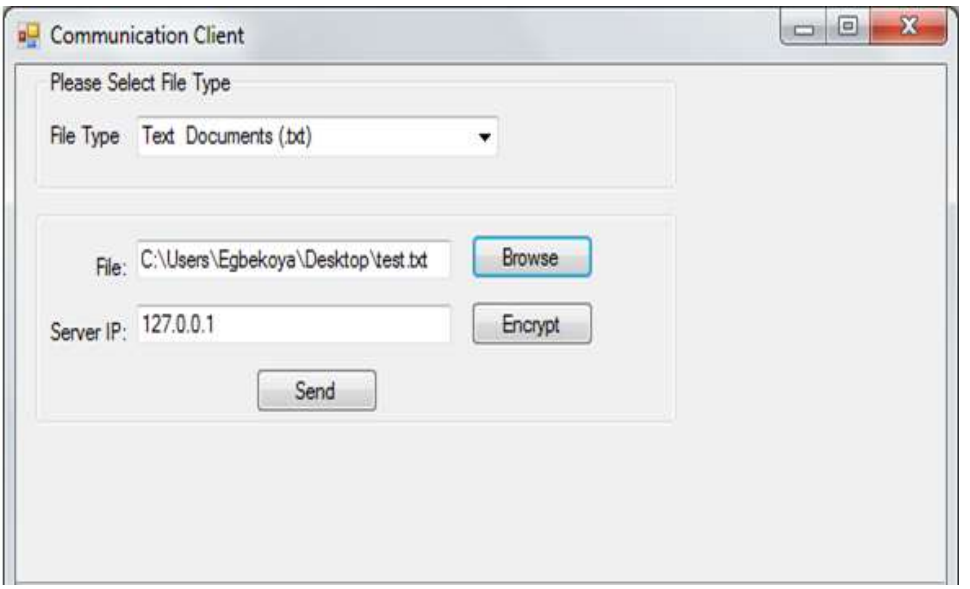

Fig 4.1 : Communication Client implementation

iii. The Browse Button: This is used to browse file for selection.

iv. *The Encrypt Button*: This encrypts the file displayed in the "file text box" using the enhanced algorithm (combination of RSA and DES).

v. The Server IP Textbox: Here the IP address of the server is specified.

vi. The Send Button: This sends the encrypted file to the specified server IP address.

B. Communication Server Implementation

This is the module at the server end. Here the encrypted file is received and decrypted using the required keys. Also, the server is started and stopped here. The communication server module is displayed in Fig

4.2 below and its component described.

i. *The Start Server Button*: This button starts the server and creates the "NCEUploads folder" on the primary hard drive the first time it is run. If the folder already exists, it does not create a new folder but uses the existing one.
ii. *The stop Server Button*: This button stops the server.
iii. *The Reset Button*: This is used to reset the counter, although not usually used.
iv. *The Clear Screen Button*: This is used to clear the progress information displayed in the text area below it.
v. *The Text Area*: It displays progress information about the received files from the client end. Above the text area is the "server IP address". Below the text area is the value of the "file size acquired" and the value of the "last block size read".
vi. The Browse Buttons: This is used to select the encrypted file and the two keys (RSA and DES) for the decryption.
vii. *The Decrypt Button*: This button decrypt the selected file with the selected keys as specified in "vii" above.

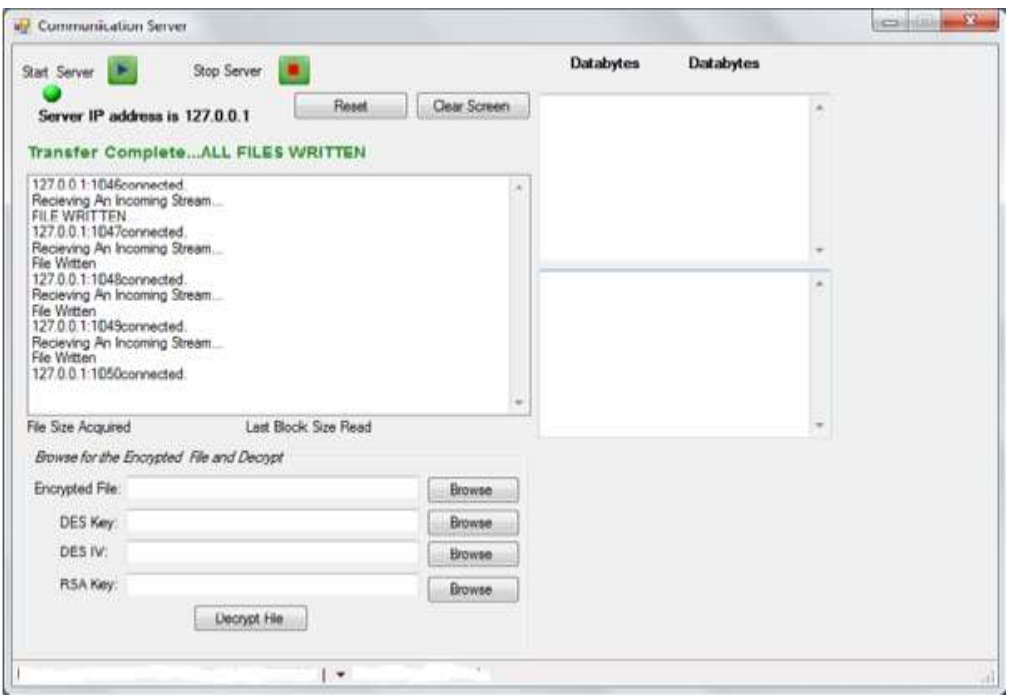

Fig 4.2 Communication Client Implementation

4.1.2 Discussion

After exploring the modules and its components above, The client and server applications are run and the server started, this creates the "NCEUploads" and NCEKeys folder in the primary hard drive where all transferred file will be stored.

On the client module, the file type is selected and the browse button is used to browse to the desired file which will be selected. The selected file is encrypted with the encryption button and sent to the typed-in IP address by clicking the send button.

On the server end, the transfer progress is displayed as in fig 4.2 above. When the transfer is completed, a pop-up indicates "file transfer successful" and the last line on text areas displays "file written". Using the browse button on the server end, select the received (encrypted) file, and the decryption keys by pressing the "decrypt button", decrypt the file which gives us back the original file.

5. Conclusion

Information Security is a means by which an organization can protect or extend a competitive advantage over others, this involves ensuring that access to the network is controlled, and that data is not vulnerable to attack during transmission across the network. Cryptography is used to ensure that the contents of a message are confidentiality transmitted and would not be altered. Confidentiality means nobody can understand the received message except the one that has the decipher key, and "data cannot be changed" that is, the original information would not changed or modified. In this work we implemented both symmetric and asymmetric encryption techniques. The asymmetric algorithm was used for the key encryption and decryption process, while the data to be sent will be encrypted and decrypted using the symmetric algorithm. The enhanced encryption algorithm used in this work combines the strength of the asymmetric and symmetric algorithm and balances their weaknesses. This technique provides a better security compared to the asymmetric or symmetric algorithm when used alone. This was implemented in Microsoft visual basic .NET

References

Bellare et al. Bellare M., Canetti R., and Krawczyk H. (1998). A modular approach to the design and analysis of authentication and key exchange protocols, *13th Symposium on Theory of Computing (STOC), ACM,* New York, 419–428.

Bellare et al., Bellare M., Desai A., Pointcheval D., Rogaway P. (1998) Relations Among Notions of Security for Public- Key Encryption Schemes, *Crypto' 98, LNCS*

Springer-Verlag, Berlin, 1462, 26–45

Bruce Schneier, John Wiley & Sons, (1996). ISBN 0-471-11709-9.

Davies, D. W. (1991). Advances in Cryptography – *EUROCRYPT'91*, *LNCS* 547, 205-220, Springer-verlag.

Douglas R. Stinson (2006). *CRYPTOGRAPHY Theory and Practice*.

Fouché Gaines, Helen (1956 (1939)). *Cryptanalysis - a study of ciphers and their solution*. Dover. ISBN 0-486-20097-3.

Kavitha Ammayappan, V. N. Sastry, Atul Negi (2008). Cluster based Multihop Security Protocol in MANET using ECC, *TENCON, IEEE Conference*.

Menezes P. and Vanstone S. (1996). Handbook of Applied Cryptography. CRC Press.

Mohammed Abutaha, Mousa Farajallah, Radwan Tahboub & Mohammad(2011). Odeh Survey Paper: Cryptography Is The Science Of Information Security. *International Journal of Computer Science and Security (IJCSS)*, 5(3).

Omar M.Barukab, Asif Irshad Khan, Mahaboob Sharief Shaik , MV Ramana Murthy (2012). Secure Communication using Symmetric and Asymmetric Cryptographic Techniques, *I.J. Information Engineering and Electronic Business*, 2, 36-42.

Paul Reid (2004). Biometrics for Network Security. *Prentice Hall*, ISBN: 0-13-101549-4

Shah, M. H., Khan, S. and Xu, M. (2005). A Survey of Critical Success Factors in E-Banking, European, *Mediterranean & Middle Eastern Conference on Information Systems*, Cairo Egypt, 7-8.

Schneier B. and John Wiley & Sons (1996). Applied Cryptography: *Protocols, Algorithms, and Source Code in C*.

Stalling W (2006). Cryptography and Network Security: *Principles and Practices*. Prentice Hall,.

Subasree S. and Sakthivel N. (2010). Design of a new security protocol using hybrid cryptography algorithms, *IJRRAS*, 2.