Abstract

Objectives: The use of Function Point Analysis for Multimedia software estimation may lead to wrong estimates and incomplete tasks, so there is a need to introduce a new method which estimates Multimedia software and Web Application.

Methods/Statistical Analysis: The calculation of Multimedia Points (MP) is depended on the five factor of the software application. The factors are screen design, animation, graphics capabilities, audio and video; and further categorized as low, medium, or high. On the basis of this factor’s complexity, Unadjusted Multimedia Points (UMP) is calculated. Then the calculation of multimedia point advances via an adjustment level, in which the cost adjustment factor for multimedia (MCAF) is, depended on a collection of General System Characteristic for Multimedia (MGSC). These MGSC are rated according to the weightage of each characteristic for a software application, this range is known as level of impact and it varies from 0 to 5. Multimedia Point is computed as MP = UMP*MCAF.

Findings: Comparison of outcomes derived from the estimation and through the practical observation for the E-Learning Application - Class XII – Physics is that on an average this project takes approximately four man days to produce one hour of content using the defined platform. It was observed that approximately 110 hours content of the application, developer spent approximately 3520 man hours to produce it. This data has been taken from the documented report of the Company from their budgetary allocation file. In the experiment, the MPs calculated approximately 110 x 04 hours of content for class XII, Physics is 344 MPs, which translates 344 x 11 = 3784 hours of the estimated effort spent in the research project. The experimental outcomes derived from the estimation and through the practical observation for the E-Learning application which have already been validated and compared with the documented report. It was found to have comparable effort spent vis-a-vis estimated MPs within margin of error being in range of ± 7%.

Application/Improvements: Factors Considered as mixed media objects leads to complex counting. The scope of improvements will emphasize on how to use the prototype counting program, which impacts on production development environment or maintenance phase for complex system architecture.

Keywords: Cost Adjustment Factor for Multimedia, General System Characteristics for Multimedia, Level of Impact for Multimedia, Multimedia Point Analysis, Unadjusted Multimedia Points

1. Introduction

Multimedia Software and its applications are being widely used in different platforms and its domain. But the development methods, cost estimation and project duration were challenging task for developing the multimedia application. Hence there is need to introduce a method to calculate Multimedia Points (MPs) considering five factors and its general system characteristic that affect the size of the software for accurate estimation.

Despite the fact that, the Function Point Analysis (FPA) has recommended a thought to evaluate the software size in both planned and non-planned manner, but this method has its own limitations and it may not estimate the size of Multimedia software projects. The characteristics and specifications of FPA do not fulfill the purpose for estimation of Multimedia software applications. Hence the use of FPA for Multimedia software estimation may lead to wrong estimates and incomplete tasks which will end up into annoying all the stakeholders of the project. Use of interactive media tools has presented a distinctive way for delivering instruction. The fascinated gadget users in the arena of visual interfaces, are being attracted, gets benefited and have new learning capabilities.

Software metrics which are used for software project size and its cost evaluation plays a very important role for
all the stakeholders involved in the project and affects time consumed in its project duration. A new concept has been introduced to estimate Multimedia or Web Applications.

The most imperative objective of intuitive interactive media programming design should be its simplicity. To achieve this, audio and visual aids should be used only when necessary as their overuse can be distracting. Multimedia software Developer should use suitable methods for designing the package which will not only enhance its capabilities but should also be user friendly. As the software size and its cost is a most critical component, this MPs tool resolves the issues of scheduling, staffing, budgeting, planning of software projects, etc. Hence, the accuracy of software can be enhanced by chosen five factors and it eight no. of MGSC which would improve the correctness of software size estimation.

This is very motivating work to estimate developed multimedia software which is a new approach to calculate multimedia points (MPs) in which eight general system characteristic encapsulate all aspects to estimating mixed media applications. Nevertheless, Mixed media projects can be utilized to present data in the combined form in a number of ways and multimedia presentations can be made to create attention for the learning of themes at distinctive levels of appreciation.

All media segments, for example, design, sound and video and so forth it adds to learning. While developing multimedia programs, developer should mainly focus on end user receptive capabilities in such a way that the user feels easy to capture, understand and organize all necessary information and integrate all information into the user’s knowledge data bank.

This paper indicates that it has professional and industrial significance from designer point of view which enhances applicability and utility of Multimedia Point Analysis (MPA). The rest of paper is organized as section 2 contains progress on size estimation of Multimedia Software, section 3 describe the brief methodology used, section 4 and 5 discuss about UMP and MCAF calculation, section 6 explains details of experiential results and finally section 7 covers about Conclusion & Future Scope in this paper.

2. Progress on Size Estimation of Multimedia Software

Researcher’s points of view from published papers as mentioned below, it is observed that FPA has its own limitations. Due to its limitation, FPA is not applicable for certain environment hence need arises to identify the environment, where FPA is not applicable.

In the context of software metrics system in software engineering, it is very important for the Software Developer and Project Manager to use the metrics to measure the qualities of software. In direct metrics we can include Cost, Effort, Lines of Code (size), Speed, Memory size, Number of Errors and in indirect metrics we include: Function (size), Quality, Complexity, Efficiency, Reliability and Maintainability.

FPA method, developed by Albrecht’s was to measure Function points by sizing software. It gave the IT industry a standard against which productivity can be measured and compared across teams, divisions and organizations. Since its inception, many variations of FPA have appeared. The majority of these are concerned with applying FPA to very specialized work environments where Albrecht’s original method was felt to be inappropriate.

Function Point (FP) approach, also named Component Point (CP), to measure the system level size of a Component-Based Software System (CBSS) specified in UML. The CP measure reflects the main features of the FP-like general model namely the partitioning of a system into different types of constituents, different sizing criteria for each constituent type, sizing of the individual constituents, summation of the constituents sizes resulting in a system-level size and an overall adjustment of the system-level size to accommodate General System Characteristics.

It was to inspect the possible threats to consistency of human-performed transaction recognition in use cases, and – if required – to propose the means to improve the objectiveness of the transaction identification process. An effort distribution of three unrelated maintenance categories was experiment to evaluate the productivity and these categories include enhancive, corrective and reductive. It was found that the notion of use-case transactions is vague, as there are at least two different definitions of this concept. The first one states that use-case transaction consists of the actor’s request and a system response. The second one was based on the elementary process known from Function Point Analysis. The main question was whether there is a cause-effect relationship between the choice of the approach to transaction recognition and on-average number of use-case transactions recognized by people. For transaction recognition and intra-method variability, they proposed a set of methods that can be used to automatically recognize transactions in use cases according to each
definition of use-case transaction. The proposed methods were implemented as a proof-of-concept tool. The transaction counts provided by the tool were compared with the manual counts with repeatable results⁵.

In the maintenance types like enhanceive, corrective, and reductive, a controlled experiment was done to assess the productivity and effort distribution. The productivity of corrective maintenance was lower than that of the other types of maintenance and a large proportion of effort is devoted to program comprehension. It was more expensive to modify than add or delete a line of code. The effort models for explaining and estimating the individual programmer's time spent on the maintenance tasks. The deleted coding line is a statistically significant predictor of maintenance effort. It requires a sizable proportion of the software maintenance effort⁶. Computer software and its applications have become an integral part of any industry. Hence it requires an accurate forecasting of software development cost which have been also increased. Different software effort estimation models are being used, but these models have been produced for particular development conditions and they assist particular software development philosophies. Present day programming advancement has not limited to particular innovation or technique⁷.

3. Methodology

The interactive media element, such as graphics, menus, audio-visuals, hyper-links, scripts and so forth that adds to the illustration on the screen, is not distinguished, hence not considered for function point analysis. It has accentuated that the developer must concentrate on creating mixed media software, to get the user’s attention, helping the user to discover and organize relevant information, and finally to integrate all those information into the user’s knowledge structure.

Developers must find out the way and locate the navigational buttons, content display control buttons, status, gauge the progress and the areas should be isolated with each other. For better results the consistency between screens should be maintained. The location of functional areas should not change and the devices used in the design should remain the same throughout as defined at the beginning of the program.

To make mix media application interactive, motivating, and relevant, a system has to be produced. The successful development of mix media modules must get attention; inform about its objectives to the end user and motivate them, create awareness of prior learning, provide learning guidance, elicit performance, provide feedback to the user, assess performance, and enhance retention for maintaining quality. If these things are taken care of in the software package then it will effectively lead to easy learning.

To highlight developer’s objective, the following multimedia components has been chosen as important factors which affect the size of for multimedia software. These factors are arranged in the Block Diagram in Figure 1.

![Block Diagram-Five Factors to Develop a Proposed Model](image)

*Figure 1. Block diagram-five factors to develop a proposed model.*
3.1 Screen Design
The designing of Screen is very important for multimedia developers as it is the area of various contents. The screen design should include the textual and graphic elements for presenting the content in a sequential manner to highlight the learning specifics. On the other hand, the content could vary as per the necessities of the package being developed. Input field should present suitable navigation tools for effective instruction to the user.

3.2 Navigation/Animation
The Navigation or Animation features make the multimedia modules interactive which enhances learning and user’s ease to handle it. Navigational elements compose a program, perform housekeeping tasks and also provide the user some control over the events. This should be defined within the system and should remain consistent.

3.3 Graphics Capabilities
In Multimedia software the information can be presented in both the text and graphics mode. Users unable to read the information given in text may easily understand it, if the same is presented by use of various visuals. Moreover, complex and difficult topics could be understood easily when it is used in appropriate graphics.

3.4 Audio and Video
Many multimedia programs use texts as a critical instructional component which is difficult for the beginners to understand. Text type of information is easy and inexpensive to develop so at times developers use it extensively as it also used minimal computer memory. The text should be supported by audio, but the multimedia software is incomplete without motion video, like home videos, commercial tapes, and movies. However the video display needs more storage space in the computer than simple animations. This requires special hardware and/or software.

For multimedia presentation, some basic elements like Scripts- in HTML/XML which require for generating reports, Web building blocks – used standard web components and the Links - to link applications to databases and other applications also required effort.

All these multimedia files audio, video and images required effort to incorporate. Domain of Multimedia System is increasing year by year. Because of under estimation or over estimation of Multimedia software, an estimation tool has to be evolved. FPA also may not be taken as accurate tool which lacks the necessary attribute to estimate Multimedia software.

3.5 Multimedia Point Analysis (MPA)
Five types of parameters have been identified for estimating the multimedia software size after studying the criteria for each parameter separately.

- Screen design
- Navigation / Animation
- Graphics Capabilities
- Audio
- Video

The weights for each factor have been assigned/Rated (Row-wise) in visually presented in Graph I.

Graph I. Values for transactions.

All elements like text, graphics icon and audio visual elements need to be utilized at the maximum to create visual appeal and organized in structured program, and thus for the purpose of systematic organization it is convenient to divide the screen into functional areas by the Developers.

3.6 Rating of Components
For estimating the multimedia software size following parameters and it complexities listed in the Table 1 being considered.

3.6.1 Estimation Part
The model gives the basis of counting parameters and initiates the Guidelines for determining its complexities.

3.6.2 Screen Design
Criteria for selecting Screen Design considered as and its parameters are listed in Table 2.
1. Nos. of Input Fields,
2. Nos. of Static pages, and  

**Table 1.** Parameters and its complexity.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Complexities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low</td>
</tr>
<tr>
<td>Text/ Input Fields</td>
<td>0 to 150</td>
</tr>
<tr>
<td>Static Pages</td>
<td>0 to 150</td>
</tr>
<tr>
<td>Dynamic Pages</td>
<td>0 to 150</td>
</tr>
<tr>
<td>Graphics Used</td>
<td>1 to 25</td>
</tr>
<tr>
<td>Animated Pictures</td>
<td>200 to 400</td>
</tr>
<tr>
<td>Length of audio file</td>
<td>2 to 4 Hrs.</td>
</tr>
<tr>
<td>Number of audio files</td>
<td>2 to 40</td>
</tr>
<tr>
<td>Audio download</td>
<td>(0.5 – 2.0 MB)</td>
</tr>
<tr>
<td>Length of video file</td>
<td>2 to 4 Hrs.</td>
</tr>
<tr>
<td>Number of video files</td>
<td>2 to 40</td>
</tr>
<tr>
<td>Video download</td>
<td>(0.5 – 2.0 MB)</td>
</tr>
</tbody>
</table>

**Table 2.** Screen design.

<table>
<thead>
<tr>
<th>Screen</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 – 150</td>
</tr>
<tr>
<td>Input Field</td>
<td>L</td>
</tr>
<tr>
<td>Static Page</td>
<td>L</td>
</tr>
<tr>
<td>Dynamic Page</td>
<td>L</td>
</tr>
</tbody>
</table>

**3.6.3 Navigation / Animation**

Criteria for selecting an animation considered as and its parameters are listed in Table 3.  
1. No. of live animation in the application,  
2. No. of embedded animation,  
3. No. of dynamic animation: animation respond to clicks is only counted, and  
4. The number of animations in the application.

**Table 3.** Navigation/animation.

<table>
<thead>
<tr>
<th>Animation</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>200 - 400</td>
</tr>
<tr>
<td>Live</td>
<td>L</td>
</tr>
<tr>
<td>Embedded</td>
<td>L</td>
</tr>
<tr>
<td>Dynamic</td>
<td>M</td>
</tr>
</tbody>
</table>

**3.6.4 Graphics Capabilities**

Criteria for selecting graphics considered as and its parameters are listed in Table 4.  
1. Nos. of Graphics Button used,  
2. Nos. of Photos Embedded, and  
3. Nos. of Scanned Images Used.

**Table 4.** Graphics capabilities.

<table>
<thead>
<tr>
<th>Graphics</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 – 25</td>
</tr>
<tr>
<td>Graphics Button/ Icons</td>
<td>L</td>
</tr>
<tr>
<td>Photos</td>
<td>L</td>
</tr>
<tr>
<td>Scanned Images</td>
<td>M</td>
</tr>
</tbody>
</table>

**3.6.5 Audio**

Criteria for selecting an audio considered as:  
a) Length of audio file embedded  
b) Number of audio files  
c) An option for audio (on or off)  
d) Option for audio download (yes or no).

**Table 5.** Shared audio and video.

<table>
<thead>
<tr>
<th>Audio/Video</th>
<th>Media</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 – 40</td>
</tr>
<tr>
<td>No of Files</td>
<td>L</td>
</tr>
<tr>
<td>Length of Audio/ Video Files</td>
<td>L</td>
</tr>
<tr>
<td>Audio/Video Download</td>
<td>L</td>
</tr>
</tbody>
</table>

**3.6.6 Video**

Criteria for selecting a video considered as:  
a) It could be based on video size or duration  
b) Number of video  
c) Flexibility to turn on and off the video  
d) Option for video download (yes or no).
Criteria for selecting Audio and Video considered and its parameters are listed in Table 5.

### 4. Unadjusted Multimedia Points Count Calculation Table

Table 6 has been facilitated the calculation of the contribution to the unadjusted multimedia point count.

#### 4.1 Calculation of the Proposed Model

#### 4.1.1 Cost Adjustment Factor (General System Characteristics) Determination

The Cost Adjustment Factor (CAF) is based on 8 General System Characteristics (GSC’s) that rate the general functionality of the application being counted. Each characteristic has associated descriptions that help determine the Level of Impact of the characteristics. The Level of Impact for Multimedia range on a scale of zero to five, from no influence to strong influence.

**Table 6.** Unadjusted multimedia point calculation for e-learning application in Physics.

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Functional Complexity</th>
<th>Complexity Totals</th>
<th>Function Type Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen Design</td>
<td><em>6</em> Low</td>
<td>6 * 4 = 24</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td><em>4</em> Medium</td>
<td>4 * 7 = 28</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td><em>5</em> High</td>
<td>5 * 10 = 50</td>
<td>50</td>
</tr>
<tr>
<td>Navigation/Animation</td>
<td><em>6</em> Low</td>
<td>6 * 4 = 24</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td><em>8</em> Medium</td>
<td>8 * 7 = 56</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td><em>1</em> High</td>
<td>1 * 10 = 10</td>
<td>10</td>
</tr>
</tbody>
</table>

**Table 7.** Matrix for description to decide MLI verses Score As.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Description to Decide MLI</th>
<th>Score As</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Flash used in Application</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Static Screen</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Dynamic Screen</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>No. of Images in a Screen</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>No. of Static Images</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>Audio (if Playing)</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>Video (if Playing)</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>No. of Rolling Images</td>
<td>0</td>
</tr>
</tbody>
</table>

The 8 general system characteristics are summarized into the cost adjustment factor. When applied, the cost adjustment factor adjusts the unadjusted multimedia point count +/-35 percent to produce the adjusted multimedia point count.

GSC’s are set of 8 questions that evaluate the overall complexity of the application. They are enumerated as:

1. Flash used in Application
2. Static Screen
3. Dynamic Screen
4. No. of Images in a Screen
5. No. of Static Images
6. Audio (if Playing)
7. Video (if Playing)
8. No. of Rolling Images

Level of Impact are based on the stated user requirements, each General System Characteristic for Multimedia...
(MGSC) must be evaluated in terms of its Level of Impact (LI) on a scale of zero to five:

0 Not present, or no influence
1 Incidental influence
2 Moderate influence
3 Average influence
4 Significant influence
5 Strong influence throughout

Each of the following GSCs descriptions includes guidelines to determine the degree of influence.

Table 8. Determine the Cost Adjustment Factor for Multimedia Software (MCAF).

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Characteristics</th>
<th>Descriptions</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Flash used in Application</td>
<td>How many Flashes are used in the Application?</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>Static Screen</td>
<td>How many Static Screens are used in the Application?</td>
<td>5</td>
</tr>
<tr>
<td>3.</td>
<td>Dynamic Screen</td>
<td>How many Dynamic Screens are used in the Application?</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>No. of Images in a Screen</td>
<td>How many Images used per Screen?</td>
<td>0</td>
</tr>
<tr>
<td>5.</td>
<td>Audio (if Playing)</td>
<td>No. of Audio Playing</td>
<td>5</td>
</tr>
<tr>
<td>6.</td>
<td>Video (if Playing)</td>
<td>No. of Video Playing</td>
<td>0</td>
</tr>
<tr>
<td>7.</td>
<td>No. of Static Images</td>
<td>No. of Static Images</td>
<td>0</td>
</tr>
<tr>
<td>8.</td>
<td>No. of Rolling Images</td>
<td>How many Rolling Images have been used</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Total MTLI</td>
<td></td>
<td>18</td>
</tr>
</tbody>
</table>

4.1.2 Guidelines to Determine Level of Impact for Multimedia (MLI)

This section presents the guidelines to determine the Level of Impact for Multimedia (MLI) for each general system characteristic.

The Matrix for Description to Decide MLI verses Score as in the Table 7 given in this section is a guide, in which one of the guideline descriptions fits the application exactly depending on the number of media elements used as MGSC. Thus a decision can be taken to determine which level of impact is the most appropriate for the multimedia application which is being estimated.

5. Determine the Cost Adjustment Factor for Multimedia Software (MCAF)

The Cost Adjustment Factor for Multimedia (MCAF) consists of 8 “General System Characteristics” which is described in the Table 8.

5.1 Procedures to Determine MCAF

The following steps outline the procedures to determine the cost adjustment factor:

Step: Action

1. Evaluate each of the 8 general system characteristics on a scale from zero to five to determine the Level of Impact for Multimedia (MLI).
2. Add the degrees of influence for all 8 general system characteristics to produce the total Level of Impact for Multimedia (TMLI).
3. Insert the TMLI into the following equation no. 1 to produce the cost adjustment factor.

\[ MCAF = (TMLI \times 0.01) + 0.75. \quad \ldots (1) \]

The Total MTLI calculated in step 4 is 18 for each of the 8 MGSC as shown in the Table 8.

So, Final MCAF can be calculated as:

\[ MCAF = (18 \times 0.01) + 0.75 \]

\[ MCAF = 0.93 \quad \ldots (3) \]

Where MCAF can vary in range from 0.75 (when all MGSCs are low) to 1.15 (when all MGSCs are high).

6. Experimental Result and Discussion

After summing up the values of these 8 MGSCs, we get a value named “Total Level of Impact for Multimedia”, or MTLI. As MTLI may vary from 0 (when all MGSCs are low) to 40 (when all MGSCs are high) as mentioned above.

MGSC items such as Dynamic Screen, Rolling Images, No. of Audio and No. of Video usually score higher for E-Learning applications. On the other hand, Flashes, No.
6.1 Calculate the Adjusted Multimedia Point Count

The final step in to determine the Adjusted Multimedia Points Count. For application counts, this is easily determined with the following equation:

\[
\text{Adjusted MP Count} = \text{UMP Count} \times \text{MCAF} \\
= 370 \times 0.93 \\
= 344 \text{ Adjusted MPs for this Multimedia Software.}
\]

Since, the MCAF can vary from 0.75 to 1.15, so the MCAF exerts an influence of ±20% on the final Adjusted MP Count (Figure 2: Screen Shot taken from the Experimental Work).

**Figure 2.** Screen shot taken from experimental work.

7. Conclusion and Future Scope

From this developed Software Size Estimation Model, a web application or an E-Learning Application or an apps which is being used widely in smart-phone devices, can be estimated. Much effort has gone into the project and hence cost is an important factor to consider for its commercial applications. The cost estimation for a multimedia application is specifically targeted and intended for commercial software development organizations that need to estimate analyze and utilize different vital metrics (i.e., Complexity, Size, Effort, Schedule, Manpower, Cost, etc.) involved in a software project during the conceptualization, planning, development, and launching phases.

Comparison of outcomes derived from the estimation and through the practical observation for the E-Learning Application - Class XII – Physics is as follow:

On an average this project takes an approximately four man days to produce one hour of content using the defined platform. This does not include time required for project setup, project initiation and the project documentation. It was observed that approximately 110 hours of Physics content for class XII, Top Chalks, an esteemed software development company, spent approximately 3520 man hours to produce it. This data has been taken from the documented report of the company from their budgetary allocation file.

From the experiment, the MPs calculated approximately 110 x 04 hours of content for class XII, Physics is 344 MPs, which is translates 344 x 11= 3784 hours of the estimated effort spent in the research project.

Likewise two more applications; Multimedia Web Application - The Read My Language and E-Learning Multimedia Application for Class XII – Mathematics have already been validated and compared.

After comparison of outcomes derived from estimation and through the practical observation from two E-Learning Applications and one web based application, the author has found to have comparable effort spent vis-a-vis estimated MPs within margin of error being in ± 7%, whereas Cheung et al.11 have demonstrated that the variety between manual counters is 10% or less. The introductory information exhibited in the article demonstrates a variety of 9.7%.

Our study is of professional and industrial significance from designer point of view which enhances applicability and utility of Multimedia Point Analysis (MPA) by uprooting a few obstacles to more extensive selection of the technique.

The consequences of this study elaborate the preparatory results that will be utilized to adjust the calculation through MPA which will provide progressively precise results. Future investigation will endeavor to recalibrate this model. As of now it appears that the most exact results from subjects that use frames based applications. This MPA Model may lead to complex counting due to mixed media objects. The Author has described the
scope of future studies which will highlight on how the use of the prototype counting program impacts when used in a production development environment or it is under maintenance stage involved in more intricate system architecture.

8. References