International Journal of Information Technology and Knowledge Management
January-June 2011, Volume 4, No. 1, pp. 305-313

ESTIMATION BY ANALOGY IN THE PERSPECTIVE EDUCATIONAL WEB HYPERMEDIA APPLICATIONS

Dheerendra Singh[1] & K. S. Dhindsa[2]

The World Wide Web has become purely a hypermedia system these days. Numerous organizations have developed number of commercial and educational Web applications. The development of good quality applications has a high cost, mostly in time and amount of difficulty involved for the developers. On applying measurement techniques to measure the quality of applications and their development process, feedback can be obtained to help control, improve and predict products and process. Over the years numerous techniques for estimating development effort have been suggested by the various experts in the field of Web engineering. These techniques have been classified mainly as Expert judgment, algorithmic model, and Estimation by Analogy (machine learning). Now the Machine learning techniques are being used as a complement or an alternative to the previous two categories. Several studies have compared the prediction accuracy of these techniques, with emphasis placed on linear regression, stepwise regression, and Estimation by Analogy. The objective of this paper is to present a comparative study of the Estimation by Analogy techniques using result from one, two and three analogies and closest analogy among three to find the minimum Magnitude of Relative Error and to reach a conclusion for the best result that can be used for early cost estimation for any Web Hypermedia Application. The Web is a system of interlinked, hypertext documents that runs over the Internet. With the help of Web browser, a user can view Web page that may contain text, images and other multimedia and navigates between them using hyperlinks. The Web is simply defined as the universe of global network-accessible information [5]. Hypermedia applications are often advocated as the most powerful tool for an effective presentation and delivery of information. The result describes the Effort Eestimation using Eestimation by Analogy (a Non Algorithmic Models) for estimating the effort for developing Web hypermedia application projects and comparing the prediction accuracy of the estimation by analogies using Web hypermedia application's datasets. All the estimations were performed using a estimation by analogy Works. In this paper the estimation effort is calculated by using, estimation by one analogy, estimation by two analogies, estimation by three analogies, and estimation by closest analogy.

Keywords: Educational Web Hypermedia, Web Engineering, Estimation, Analogy

1. INTRODUCTION

An estimation process involves the following activities:

i. The capturing of data about past Web hypermedia applications or even past development phases within the same project.

ii. The identification of size metrics, cost drivers and the formulation of theories about their relationship with effort.

iii. The generation of prediction models to be applied to Web hypermedia applications.

iv. The assessment of how effective those prediction models are.

This paper focuses on effort prediction using Estimation by Analogy for design and authoring processes. Design covers the methods used for generating the structure and functionality of the application, and typically, does not include aspects such as hypermedia application requirements elicitation, feasibility consideration and applications maintenance. In addition, design phase also incorporates the conceptual design of the application, reflected in map diagrams showing documents and links.

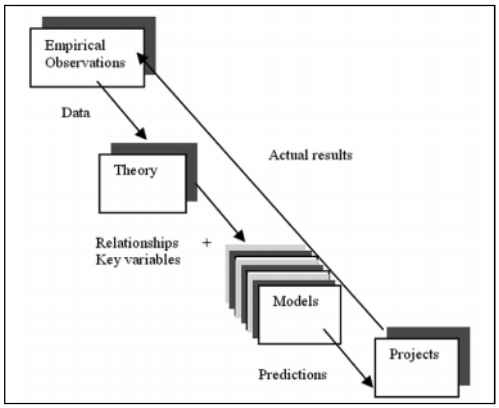

Fig. 1: General Estimation Process

Software experts developers recognise the importance of realistic estimates of effort to the successful management of software Web hypermedia applications, including Web applications. Having realistic estimates at an early stage in

[1]SRM IET Ambala, Haryana, India

[2]BBSBEC Fatehgarh Sahib, Pb, India

Email: [1]dheerendra_singh76@yahoo.co.in, [2]kdhindsa@gmail.com

a project's life cycle allow project managers and development organisations to manage resources effectively. Various techniques for cost and effort estimation have been evolved in the previous years. All these techniques have been classified into three general categories.

2. TYPES OF ESTIMATION TECHNIQUES

2.1. Expert Judgement

In this technique cost and effort estimation derived on the basis of expert's previous experience on similar Web hypermedia applications. These estimates are not repeatable because of the implicit nature of derivation of the estimation. However, the technique may be an effective tool on its own and for algorithmic models. This technique may be used for approximation.

2.2. Estimation by Analogy

In this model, the techniques that are based on artificial intelligence are being used. These techniques used as a complement or alternative to the expert judgement and algorithmic model techniques. Machine learning models encompass fuzzy logic models, regression trees, neural networks, and case-based reasoning (Estimation by Analogy). In any Web hypermedia project, the developers are required to focus on design and authoring. Designs covers the methods used for generating the structure and functionality of the application. In the authoring (Web site designing) the developers deals with the actual Web content, structure and presentation. Here selection of Estimation by Analogy as a prediction technique, aims for the following reasons:

i. Estimation by Analogy presented the best prediction accuracy when using a dataset of Web hypermedia applications.

ii. Estimation by Analogy is an intuitive method and there is evidence that experts apply analogic reasoning when making estimates.

iii. Estimation by Analogy is simple and flexible, compared to algorithmic models.

iv. Estimation by Analogy can be used on qualitative and quantitative data, reflecting closer types of datasets found in real life.

In the Estimation by Analogy approach project's characteristics, for which the estimation is being done are described in relation to a number of attributes (e.g. application complexity, link complexity etc). This description is used to find other similar already finished Web hypermedia applications, and an estimate for the new project is made based on the known effort values for those finished Web hypermedia applications.

The first step to estimation by analogy is to characterize the project for which the estimate is to be made in terms of a number of attributes [1]. These variables are then used to find other similar Web hypermedia applications (that have already been finished) and use the values from their variables to estimate effort. In other words, an estimate for the new project is made based on the known effort values for the finished Web hypermedia applications. In the scope of this paper, Web hypermedia applications are Web applications and size-related variables are hyper document size, connectivity, compactness and stratum. The range of variables must be limited to the information available at the point that the prediction is required. The variables should portray the project as precisely as possible. It is also essential to choose at least one variable that characterizes the project in terms of its size. To estimate the effort using analogy, the analogy effort estimation tool that automates the process and provides an environment where data can be stored, analogies found and estimates generated. The distance is measured in an optimum subset of the n-dimensional, normalized space. By normalized we mean that all dimensions are in the range 0 to 1, to ensure that they all have the equal influence. One of the important features of is the ability to determine the optimum combination of variables for finding analogies. This task can be time consuming depending on the number of variables as it employs a brute force algorithm or exhaustive search of all possible permutations. This is relatively slow for a large number of variables. Although the number of analogous Web hypermedia applications used to estimate the effort can vary, literature in the field of software engineering suggest that the closest analogous project (1 analogy), followed by the two closest analogous Web hypermedia applications (2 analogies), are generally the most effective options. However, it is possible that dissimilar datasets may reveal different characteristics. One, two and three analogies have been used to predict effort. Once the analogous Web hypermedia applications are identified, an unweighted mean of the known effort values is used to determine the predicted value. Consequently, the estimation process can take optimal advantage of the data available. The user chooses the number of variables and their names. The only exceptions are the project number, status and actual effort which are compulsory. The Status attribute shows whether or not a project has been completed. Web hypermedia applications whose status is completed, can be used as a source of analogy. All completed Web hypermedia applications must have values stored for their effort field as it supplies the foundation for prediction.

2.3. Algorithmic Models: Linear Regression and Stepwise Regression

This technique is used to represent the relationship between effort and some characteristic (one or more) of a project.

The cost estimation in this model is usually based on the size of the software. That is the number of lines of source code, number of pages, number of links etc. Few examples of this model are the COCOMO model, the SLIM model and Modular.

Table 1
Comparison: Non Algorithmic Vs Algorithmic Estimation Techniques

Non Algorithmic Estimation Techniques	Algorithmic Estimation Techniques
Machine learning techniques have in recent years been used as a complement or an alternative to the previous two categories.	Algorithmic models represent the relationship between effort and one or more project characteristics. The main cost driver used in such a model is usually some notion of. AM need calibration or to be adjusted to local circumstances.
Examples include fuzzy logic models, regression trees, neural networks, and case-based reasoning	Examples of this model are the COCOMO model, the SLIM model and Modular.

The general purpose of linear and stepwise multiple regression models is to learn more about the relationship between several independent or predictor variables and a dependent or response variable. In the scope of this paper the dependent variable is effort and the independent variables are hyper document size, reused documents, connectivity, compactness, stratum and structure [4]. Regression modeling is one of the most widely used statistical modeling techniques for fitting a quantitative response variable y as a function of one or more predictor variables x1, x2,...,xp. Regression models are widely used because they often provide an excellent fit to a response variable when the true functional relationship, if any, between the response and the predictors is unknown.

3. ESTIMATION PROCESS

Estimation by analogy is an application of a case-based reasoning approach. It is a form of analogical reasoning where cases stored on the case base and the target case are instances of the same category. A case is a Web project (new or finished). An effort estimate for a new Web project is obtained by searching one or more similar cases, each representing information about finished Web hypermedia applications. The rationale for EA is the use of historical information from completed Web hypermedia applications with known effort. It involves:

The characterizing of a new active project p, for which an estimate is required, with attributes (features) common to those completed Web hypermedia applications stored in the case base. The features represent size measures that have a bearing on effort. Feature values are normally standardized (between 0 and 1) so they can have the same degree of influence on the results. The use of this characterization as a basis for finding similar (analogous) completed Web hypermedia applications, for which effort is known. This process can be achieved by measuring the "Euclidean distance" between two Web hypermedia applications, based on the values for k features for these Web hypermedia applications. Although numerous techniques are available to measure similarity, nearest neighbour algorithms using the unweighted Euclidean distance have been the most widely used in Software and Web engineering. The generation of a predicted value of effort for a given project p based on the effort for those completed Web hypermedia applications that are similar to p. The number of similar Web hypermedia applications in general depends on the size of the data set. For small data sets typical numbers are 1, 2 and 3 closest neighbours (cases). The calculation of estimated effort is often obtained by using the same effort value of the closest neighbour, or the mean of effort values from 2 or more cases. In Software engineering and Web engineering, a common choice is the nearest neighbor or the mean for 2and 3 cases. Once the data had been collected, one non-algorithmic model- estimation by analogy is used to estimate Web Hypermedia Project effort. The predictive power of the estimation model was measured using the Mean Magnitude of Relative Error (MMRE).

Actual effort was calculated as:

$$AE = \sum_{i=1}^{i=n} PAE + \sum_{j=0}^{j=m} MAE + \sum_{k=0}^{k=0} PRE \qquad (1)$$

where PAE is the page authoring effort, MAE the media authoring effort and PRE the program authoring effort.

3.1. Advantages of Estimation by Analogy

Potential advantages of using EA for effort estimation are as follows:

i. EA has the potential to alleviate problems with calibration.

   EA can give reasonably high quality estimates provided the set of past Web hypermedia applications for a certain company has at least one project similar to a given target new project (which may belong to a different Company). EA can give

better effort estimates because, unlike an algorithmic model, it does not consider the whole set of past finished Web hypermedia applications in order to calculate an effort estimate. It is usual to consider only one to three finished Web hypermedia applications that are the most similar to the new project for which an effort estimate is needed. In the scope of this study each data set employed is assumed to contain data on Web hypermedia applications hypothetically having the same attributes.

ii. EA can be valuable where the domain is complex and difficult to model.

Many features (Web application's size, Web developer's experience etc) can potentially influence a Web project's development effort. However, identifying these features and understanding how they interact with each other is difficult. For example, there is no standard as to what size measures should be employed to measure a Web application's size. Proposals have been made, but to what extent the proposed size measures can be collected early enough in the development cycle such that they are useful as effort predictors, is still an open research question. This makes the Web cost estimation domain complex and difficult to model. In addition, to develop an algorithmic model, it is necessary to determine which features can predict effort, where the number of features has an influence on how much historical data is necessary to develop a model. Unfortunately, historical data is often in short supply and for those companies who develop Web hypermedia applications it is even more difficult to obtain given that:

a. Web Web hypermedia applications have short schedules and a fluidic scope.

b. A Web project's primary goal is to bring quality applications to market as quickly as possible, varying from a few weeks to 6 months.

c. Processes employed are in general heuristic, although some organizations are starting to look into the use of agile methods.

EA has the potential for successful use without a clear model of the relationship between effort and other project features; rather than assuming a general relationship between effort and other project features applicable to all Web hypermedia applications, it relies predominantly on selecting a past project that is similar to the target project.

iii. The basis for an estimate can be easily understood: Algorithmic models use a historical data set to derive cost models where effort is the output and size measures and cost drivers are used as input, with an empirical relation between input and output. Therefore, an algorithmic model generated using data from a given company A can only be applied to a different company B if the relation between effort and its inputs embodied in the model also applies to company B's estimation problem. EA, on the other hand, bases estimates on concrete past cases, a familiar mode of human problem solving. A large number of managers and developers is comfortable estimating in this manner.

iv. EA can be used with partial knowledge of the target project: Algorithmic models usually require a fixed set of input measures to make an estimate. However, measures available for estimating vary from project to project, where sometimes an input measure is not known at the time an effort estimate is required. EA addresses this issue by allowing the use of any of the input measures available, provided those measures are also available from past Web hypermedia applications. This approach avoids the need to impose specific input, fixed for all Web hypermedia applications.

4. IDENTIFYING TECHNIQUE OF ESTIMATION BY ANALOGY

The accuracy of estimates generated using EA relies upon three broad factors: availability of suitable similar cases, soundness of the strategy used for selecting them, and the way in which differences between the most similar cases and the target case are considered to derive an estimate. There is no recipe for applying each of those factors. Hence empirical investigation is necessary to help identify which combination of factors can work best given the characteristics of past and target Web hypermedia applications, and a particular data set.

5. TECHNIQUE OF ESTIMATION BY ANALOGY

Estimation by Analogy involves:

i. Characterising a project p for which an estimate is required, i.e., identifying project attributes (application size, link complexity etc) which can influence effort.

ii. Use of this characterisation as a basis for finding similar (analogous) completed Web hypermedia applications, for which effort is known.

iii. Use of these effort values, possibly with adjustment, to generate a predicted value of effort for p.

iv. When using CBR there are a number of parameters to decide upon:

a) Similarity Measure;

b) Scaling;

c) Number of analogies;

d) Analogy Adaptation.

Each parameter in turn can be split into more detail, and maybe incorporated for a given CBR tool, allowing several CBR configurations.

5.1. Similarity Measure

Similarity Measure measures the level of similarity between Web hypermedia applications, i.e., cases. The similarity measure most frequently used in Software engineering and Web engineering literature is the un weighted Euclidean distance. In the context of this, three measures of similarity, namely the un weighted Euclidean distance, the weighted Euclidean distance and the Maximum measure, have been used each of which are detailed below:

All formulas presented in this sub-section assume that x and y represent size attributes for Web hypermedia applications, e.g., x might be page-count and y page-complexity. The pair (x_i, y_i) represents instances of x and y for project i.

Un weighted Euclidean distance:

The Euclidean distance d between the points (x_0, y_0) and (x_1, y_1) is given by (2) and illustrated in Figure 2, by representing co-ordinates on a two-dimensional space:

$$d = \sqrt{[(X0 - X1)^2 + (Y0 - Y1)^2]} \qquad (2)$$

The number of attributes employed will determine the number of dimensions used.

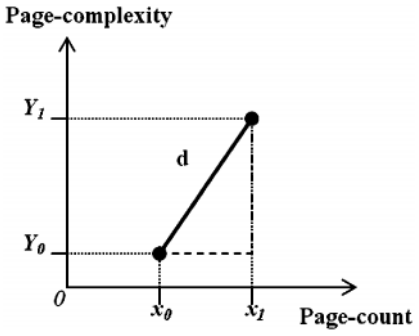

Fig. 2: Un Weighted Euclidean Distance Using Two Size Attributes [8]

Weighted Euclidean distance:

It is common in CBR for the attributes, i.e., features vectors to be weighted to reflect the relative importance of each feature. The weighted Euclidean distance d between the points (x_0, y_0) and (x_1, y_1) is given by the formula:

$$d = \sqrt{[Wx(X0 - X1)^2 + Wy(Y0 - 1)^2]} \qquad (3)$$

where w_x and w_y are the weights of x and y respectively.

Maximum measure:

Using the maximum measure, the maximum feature similarity defines the case similarity. For two points (x_0, y_0) and (x_1, y_1), the maximum measure d is equivalent to the formula:

$$d = \sqrt{[\max(X0 - X1)^2, (Y0 - Y1)^2]} \qquad (4)$$

This effectively reduces the similarity measure down to a single feature, although the maximum feature may differ for each retrieval episode.

5.2. Scaling or Standardisation

Standardisation represents the transformation of attribute values according to a defined rule such that all attributes are measured using the same unit. One possible solution is to assign one to the maximum observed value m and then divide all values by m.

5.3. Number of Analogies

The number of analogies refers to the number of similar educational Web hypermedia applications (cases) that will be used to generate the estimation. In general, only the most similar cases are selected. For Angelis and Stamelos when small sets of data are used it is reasonable to consider only a small number of analogies[2]. In this paper we have used 1, 2 and 3 analogies[7].

5.4. Analogy Adaptation

Once the most similar cases have been selected the next step is to decide how to generate the estimation. Choices of analogy adaptation techniques presented in the Software engineering literature vary from the nearest neighbour, the mean of the closest analogies, the median, inverse distance weighted mean and inverse rank weighted mean, to illustrate just a few. In the Web engineering literature, the adaptation used to date is the mean of the closest analogies. That is the way, the mean, median and the inverse rank weighted mean has been opted.

Mean: Represents the average of k analogies, when k>1.

Inverse rank weighted mean [9]: Allows higher ranked analogies to have more influence than lower ones. For example, using 3 analogies, the closest analogy (CA) would have a weight = 3, the second closest (SC) a weight = 2 and the last one (LA) a weight =1. The estimation would then be calculated as:

$$(3*CA + 2*SC + LA)/6 \quad (5)$$

Median: Represents the median of k analogies, when k > 2.

6. MEASURING THE PREDICTIVE POWER OF THE ESTIMATION MODELS

The predictive power of the estimation models was evaluated with test metric: Mean Magnitude of Relative Error (MMRE). The MMRE give an indication of the capability of a predictive model. Both are widely used and are not limited to regression based methods. The mean, used by the MMRE, takes into account the numerical value of every single observation in the data distribution. Although this characteristic makes it an integral component of statistical analysis it is not a good option for skewed distributions. Small MMREs and indicate good prediction models. Both MMRE use the Magnitude of Relative Error [6] in their calculations, which is defined as:

$$MRE = |Eact\_i - Epred\_i)/ Eact\_i| \quad (6)$$

Eact is the actual effort and Epred is the estimated effort.

MMRE [11] use the Magnitude of Relative Error (MRE) in their calculations, which is defined as:

$$MMRE = (1/n) \sum_{i=1}^{i=n} \bullet |Eact\, i - Epredi) / Eact\, i| \quad (7)$$

7. RESULTS USING DIFFERENT TECHNIQUES

Metrics included or selection of attributes of the target Web hypermedia application are given in the table 3. After Analyzing, the properties of the target Web hypermedia application, the next step is to find the data set of the hypermedia applications which are similar with the matrix of the target Web hypermedia applications. Table 3, contain the extension type of files and their description. Table 4, contain the data set of the Web hypermedia applications which have been finished in past and similar to the target Web hypermedia applications.

The metrics shown in the table 3, are the metrics for the target Web hypermedia application. Table 2, shows the type of files and description of these files used in developing the seven Web Hypermedia Applications.

Table 2
Type of Files and their Descriptions

Type of file	Description
UTF8	UTF-8 (8-bit UCS/Unicode Transformation Format) is a variable-length character encoding for Unicode. Unicode is a character set supported across many commonly used software applications and operating systems. For example, many popular Web browser, e-mail, and word processing applications support Unicode.
INF	INF stands for information; An INF file is a text file that contains all the information necessary to install a device in Windows O.S.
PDF	Portable Document Format (PDF) is a file format created by Adobe Systems in 1993 for document exchange.
GIF	The Graphics Interchange Format (GIF)is a bitmap format that was introduced by CompuServe in 1987. The format supports up to 8 bits per pixel. It also supports animations.
PNG	Portable Network Graphics (PNG) is a bitmapped image format that employs lossless data compression.
SWF	It's full form is Shockwave flash. Adobe makes available plugging to play SWF files in Web browsers on many desktop operating systems.
JPEG	"JPEG" stands for Joint Photographic Experts Group. It is a method of compression of images.
XML	XML stands for EXtensible Markup Language. It is a markup language much like HTML. XML was designed to carry data, not to display data. XML tags are not predefined. You must define your own tags.
PHP	In early versions, PHP used to stand for Personal Home Page. It was renamed to Hypertext Preprocessor. PHP is a scripting language originally designed for producing dynamic Web pages.
ASPX	ASP.NET pages, known officially as "Web forms", are the main building block for application development. Web forms are contained in files with ASPX extension.
ASP	Active Server Pages (ASP) is Microsoft's first server-side script engine for dynamically-generated Web pages.
CSS	Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in a markup language. Its most common application is to style Web pages written in HTML and XHTML.

Table 3, shows the size metrics with their description and related file extension used in the development of Web Hypermedia Applications. Table 4, shows the seven Web Hypermedia Applications with the size metrics used in their development. The naming of the Web Hypermedia Applications is shown in table 4, as WHA-1 and so on.

Table 3
Size Metrics for Web Hypermedia Applications

Metric	Type of File Extension	Description
Page Count (PaC)	.html, .htm, .css, .pdf, .doc, .zip, .inf, .utf8, .xml	Total number of html or htm files
Media Count MeC)	.jpg, .jpeg, .gif, .png, .bmp, .swf	Total number of original media files
Program Count (PRC)	.aspx, .asp, .php, .js, .class .java	Total number of Java Script files and Java applets
Reused Media Count (RMC)	.jpg, .jpeg, .gif, .png, .bmp, .swf	Total number of reused/modified media files.
Reused Program Count(RPC)	.aspx, .asp, .php, .class, .java	Total number of reused/modified programs.

Using one Analogy, the estimated effort for target project is taken by choosing by actual effort of the one most similar Web hypermedia application among the several same types of Web hypermedia applications, which have been finished in the past. Using two analogies, the estimated effort is calculated by taken the mean of the actual effort of two Web hypermedia applications which are similar to the target Web hypermedia application in metrics.

Estimated effort using two analogy = (AE1 + AE2)/2 (2)

Estimated effort using three analogies is calculated as using inverse rank weighted mean of the three actual efforts of the past Web hypermedia application.

Estimated effort using three analogy =
(3*AE1+2*AE2+AE3)/6 (3)

Table 4
Data Set for Web Hypermedia Applications

Web Hypermedia Application ID	Page Count	Media Count	Program Count	Reused Media Count
WHA-1	114	327	0	138
WHA-2	173	504	0	253
WHA-3	110	404	0	2134
WHA-4	1094	1486	626	13847
WHA-5	187	49	0	86
WHA-6	297	143	0	1327
WHA-7	588	276	6	6911

Table 5, contains the page authoring effort, the media authoring effort, the program authoring effort and the actual effort for the seven Web Hypermedia Applications. The actual effort is the sum of the page authoring effort, the media authoring effort and the program authoring as given in equation(1). The detailed information regarding actual effort of a Web hypermedia application is given in the appendix.

Table 5
Effort for Web Hypermedia Applications

Wha -ID	Page Authoring Effort	Mediaauthouringeffort	Progauthouringeffort	Actual Effortinmin	Actual Effortin Hours
WHA-1	13528.83	8597.26	0	22126.10	368.77
WHA-2	15410.2	7112.4	0	22522.6	377.37
WHA-3	20832.02	9614.78	0	30446.81	507.45
WHA-4	34207.22	22053.66	2402.45	58663.34	977.72
WHA-5	13157.39	8387.27	0	21544.69	359.08
WHA-6	18108.93	8357.97	0	26466.89	441.11
WHA-7	21185.56	11636.99	1973.65	34796.19	579.94

Table 6, contain the % magnitude of the relative error of the Web hypermedia applications. The Magnitudes of the relative errors in table 5, have been calculated using equation (4).

$$\%MRE = 100 \times | (Eact\_i - Epred\_i) / Eact\_i | \quad (4)$$

Eact is the actual effort and Epred is the estimated effort.

MMRE use the Magnitude of Relative Error (MRE) in their calculations, which is calculated using equation (5).

$$\%\ MMRE = 100 \times (1/n) \sum_{i=1}^{i=n} | (Eact\_i - Epred\_i) / Eact\_i | \quad (5)$$

Table 7, contains the summary of the percentage magnitude of the relative error taken from the table 6, for the seven Web Hypermedia Applications. MRE-1A stands for the magnitude of the relative error calculated by using one analogy, similarly MRE-2, MRE-3A, MRE-CA, represents the magnitudes of the relative errors calculated by using two, three and closest analogies respectively.

Table 6
Estimated Efforts & %MRE for Web Hypermedia Applications

WHA ID	ACT EFT	1-A	MRE	2-A	MRE	3-A	MRE	C-A
WHA-1	368.77	579.94	0.5726	709.06	0.9228	489.53	0.3275	0.3275
WHA-2	375.38	441.11	0.1751	510.53	0.3600	446.37	0.1891	0.1751
WHA-3	507.45	579.94	0.1429	668.40	0.3176	576.82	0.1367	0.1367
WHA-4	977.72	441.11	0.5488	543.70	0.4439	576.82	0.4100	0.4100
WHA-5	359.08	579.94	0.6151	510.53	0.4218	489.53	0.3633	0.3633
WHA-6	441.11	977.72	1.216	543.7	0.2311	446.37	0.0772	0.0772
WHA-7	579.94	375.38	0.3527	668.40	0.1525	489.53	0.1559	0.1525
SUM	3609.45	3975.14	3.6232	4154.32	2.8497	3514.97	1.6597	1.6423
%MMRE			51.76		40.71		23.71	23.46

Table 7
Summary of the % Magnitude of the Relative Errors

WHAID	MRE-1A	MRE-2A	MRE-3A	MRE-CA
WHA-1	0.5726	0.9228	0.3275	0.3275
WHA-2	0.1751	0.36	0.1891	0.1751
WHA-3	0.1429	0.3176	0.1367	0.1367
WHA-4	0.5488	0.4439	0.41	0.41
WHA-5	0.6151	0.4218	0.3633	0.3633
WHA-6	1.216	0.2311	0.0772	0.0772
WHA-7	0.3527	0.1525	0.1559	0.1525

Figure 3, shows the variation of the percentage magnitudes of the relative errors using one analogy, two analogies, three analogies and closest analogy for seven Web hypermedia applications. The percentage magnitudes of relative errors have been taken on positive Y axis and the Web hypermedia applications have been taken on positive X- axis. The range of percentage magnitudes of relative errors is from 0 to 1.4.

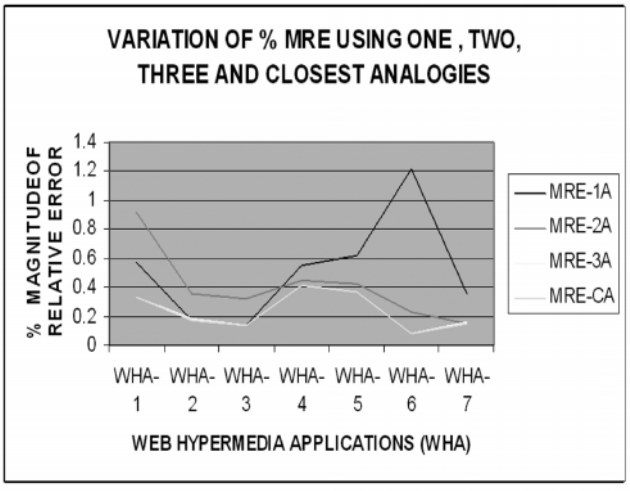

Fig. 3: Variation of the %MRE with Respect to the Web Applications taken on the X-Axis

The main aspect of the table 8 is that in all cases, the Analogy method outperforms the best Algorithmic method, suggesting that at least for the datasets used the closest analogy approach is a superior technique to the other methods. A similar result has also been reported in Shepperd et al., where they suggest the following explanation for the phenomenon: Regression analysis is based upon minimizing the sum of the squares of the residuals. A residual is the difference between actual effort and predicted effort.

Table 8
Summary of the Predictive Power Using Estimation By Analogy

Web Hyper-Media Applic-Ations Set	Estimation By One Analogy MMRE (%)	Estimation By Two Analogies MMRE (%)	Estimation By Three Analogies MMRE (%)	Estimation By Closest Analogy MMRE (%)
	51.76	40.71	23.71	23.46

As the MMRE is based upon the average of the sum of the residuals, poor predictions have far more influence when trying to fit a regression equation than they do when assessing the overall predictive performance of the method. This can lead to small anomalies in the relative performance of linear regression and stepwise regression models.

8. CONCLUSION

The result describes the Effort Eestimation using Eestimation by Analogy (a Non Algorithmic Models) for estimating the effort for developing Web hypermedia application projects and comparing the prediction accuracy of the estimation by analogies using Web hypermedia application's datasets. All the estimations were performed using a estimation by analogy Works. In this work the estimation effort is calculated by using, estimation by one analogy, estimation by two analogies, estimation by three analogies, and estimation by closest analogy. The best prediction on Web hypermedia applications datasets were obtained using the result with eestimation by closest analogy technique. Further the better prediction on Web hypermedia applications datasets were obtained using the result with Estimation by three Analogies from combination of these three Estimation by Analogy techniques, this second best result among three techniques. These results suggest that estimation by analogy is a candidate technique to effort estimation and that with the aid of an automated environment it is a practical technique to apply to Web hypermedia applications development effort prediction. Further empirical investigation is also necessary to determine if these results can also apply to Web software applications. As part of the future work, comparison of the performance of estimation by analogy technique can be done against human estimation and regression techniques in a similar way. To conclude, there is an urgent need for adequate hypermedia Web hypermedia applications development effort prediction at an early stage in the development. Further, comparison of the performance of estimation by analogy technique can be done on large set of the different types of the Web hypermedia application projects and regression techniques in a similar way. As the use of the Web as a delivery environment increases, effort estimation by analogy can contribute significantly to the reduction of costs and time involved in developing Web hypermedia applications.

REFERENCES

[1] A.J.C. Cowderoy (2000), "Measures of Size and Complexity for Web-site Content", Proceedings of the Combined 11th European Software Control and Metrics Conference and the SCOPE conference on Software Product Quality, Munich, Germany, pp: 423-431.

[2] Angelis, L., and Stamelos, I., "A Simulation Tool for Efficient Analogy based Cost Estimation", Empirical Software Engineering, 5, 2000, 35-68.

[3] Brown, P. J. (1990), "Assessing the Quality of Hypertext Documents", In Hypertext.: Concepts, Systems and Applications: Proceedings of ECHT '90, A. Rizk, N. Streitz, and J. Andre, Eds., Cambridge University Press, Cambridge, pp. 1-12.

[4] E. Mendes, Mosley, N., and Counsell, January-March (2001), S., "Web Metrics -Estimating Design and Authoring Effort", IEEE Multimedia, Special Issue on Web Engineering, pp. 50-57.

[5] G. Booch, J. Rumbaugh and I. Jacobson (1998), "The Unified Modeling Language User Guide", Addison-Wesley, Reading MA, 1998.Hansen, S., University of Western Sydney Macarthur, Presentation in the AusWeb97 Conference, 1997 and International Workshop on Web Engineering.

[6] Kemerer, C.F., "An Empirical Validation of Software Cost Estimation Models", Communications of the ACM, 30(5), 1986, 416-429.

[7] Mendes, E., Counsell, S., and Mosley, N., "Measurement and Effort Prediction of Web Applications", Proc. Second ICSE Workshop on Web Engineering, 4 and 5, June 2000; Limerick, Ireland, 2000.

[8] Mendes, E., Mosley, N., and Watson, I., "A Comparison of Case-based Reasoning Approaches to Web Hypermedia Project Cost Estimation", Proceedings of the WWW'2002 Conference, 2002.

[9] Mendes, E., Watson, I., Triggs, C., Mosley, N., and Counsell, S. "A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications", Proceedings of the IEEE Metrics Symposium, 2002, 131-140.

[10] Roger S. Pressman (2005), "Software Engineering", A Practitioner's Approach, Sixth Edition,, McGraw Hill.

[11] Shepperd, M.J., Schofield, C., and Kitchenham, B., "Effort Estimation using Analogy", Proceedings ICSE-18, IEEE Computer Society Press, Berlin, 1996, 170-178.