

Web-based Evaluation System for Learning Management Systems

MOMANI, Alaa

January 27, 2010

Abstract

E-learning systems have become an issue in recent years. A learning management system (LMS) is an electronic environment helps the educational society to communicate, exchange information, manage, and schedule the learning process. This study has provided a web-based evaluation system that may help the users to choose the convenient system depending on their needs and their type of usage by using developed and intelligent decision making algorithm. This system was developed by using the MS -Visual Studio .NET and its database was designed by using MS-SQL Server. In order to evaluate the LMSs, we must consider the most important features of any LMS and classify them as unified criteria that may enable us to evaluate the systems. The system's database contains a full description on the most famous LMSs over the web and the widely-used over the world.

Keywords: Evaluation of LMS; LMS; decision making; choosing an LMS

1. Introduction

In these days, we can see clearly the rapidly changing and developing in technology. E-Learning becomes one of the important fields and one of the fastest growing areas of the high technology sector development, especially in the academic environments. It is the newest way for the individuals and organizations to get knowledge and meet their requirements. Knowing a little bit about learning styles can help you determine if online learning is designated to you. The interaction and delivery methods used in online classes are dramatically different from traditional classes, so understanding how you learn is a good part of the decision-making process. Undoubtedly, the knowledge in general can help you improve your study habits and be successful in any educational setting, regardless of what type of learner you are.

2. The Evaluation Method

The main goal of this study is to develop a web-based decision support system (DSS) to evaluate LMSs over the web by using a flexible and smart algorithm which will contain some steps to be completed. This algorithm is clever; using some artificial intelligent concepts with fuzzy logic values. This algorithm contains a mathematical equation depending on taking multi-attribute entities and a degree of importance for each attribute. This equation is the responsible to find the best LMSs from the group of systems included in the evaluation operation. This paper here contains a complete description of the algorithm from the logical, mathematical, programmatic, applicable, and the user's view points.

2.1 Logical View

Logically, the evaluation operation will contain a number of steps in order to be completed. Some of these steps need from us, as evaluators, to enter some values and entries depending on his/her own rating to the evaluation's elements. This data will be fuzzy values. From this project, we aim to get one result that refers to a LMS. In the following, a list of all steps necessary for a complete evaluation operation:

1. We firstly should give the number of features to our LMS depending on our needs.
2. We should give the needed number of LMSs which matches the features we want to include in the evaluation operation.
3. Now, we should weigh each feature included in the features' group and that your weight should depend on the degree of your needs. These values will enter the evaluation equation and affect on the result, as well.
4. Here, we should rank each feature to each LMS. This rank will give the degree of efficiency and performance of each feature inside each LMS.
5. In that case, we have to multiply each weight by the corresponding ranks of all LMSs.
6. Then, each multiplication result came out of the same LMS shall be added.
7. Each adding result came out of the previous step shall be divided by the number of features entered in the first step. The group of results will be the group of grades of all LMSs included in the evaluation operation.
8. Now, we have to apply the maximum method in order to get the maximum grade came out of all results in the previous step. Consequently, the LMS which gets the maximum value will be the system we seek.

2.2 Mathematical View

2.2.1 Multi-attribute Decision Making Method

Every multi-attribute decision making problem have a group of criteria and alternatives, consider that we have m criteria and n alternatives. Let C_1, \dots, C_m refers to the group of criteria and A_1, \dots, A_n refers to the group alternatives. The decision making table below (Table 2.1) shows the standard of multi-attribute decision making methodology. In this table, each row represents a criterion and each column describes the degree of performance of each alternative. The score a_{ij} describes the performance of alternative A_j for criterion C_i . The higher score value means a better performance.

Table 2.1: The decision making table

		x_1	.	.	.	X_n
		A_1	.	.	.	A_n
w_1	C_1	a_{11}	.	.	.	a_{m1}
.
.	.	.	.	a_{ij}	.	.
.
w_m	C_m	a_{m1}	.	.	.	a_{mn}

As shown in the table, the weights w_1, \dots, w_m are assigned to the criteria. For example, weight w_i represents the importance of criteria C_i in the decision, it is always positive. On the other hand, the values x_1, \dots, x_n assigned to the alternatives in the decision and they are the final ranking values of the alternatives. Usually, higher ranking value means a better performance of the alternative, so the alternative with the highest ranking value is the best of the alternatives.

2.2.2 Linear Weighted Attribute Model

One of the five models proposed by Anderson (1990) for software selection was the "*Linear Weighted Attribute Model*". In this model, each attribute should get a performance rating, or weight. The other variable is the rank. It is the frequency degree of a specific attribute of one element from the evaluation members.

In our case, this equation will give us the score of the (i^{th}) system included in the evaluation operation. It results the summation operation of multiplying the weights group by the corresponding ranks of the (i^{th}) system. In fact, we need to get an independent grade, or general score, for each LMS in the evaluation operation overall its features group. Means that, the score shall be common for all attributes in the operation, therefore, we have to divide the result came out of by the total number of attributers, or features included in the evaluation operation. The equation will be as follows:

$$G_i = \frac{\sum_{j=1}^n (W_j * R_{ij})}{n} \Bigg|_{i=1}^m, \text{ Where } (m) \text{ is the number of LMS's}$$

After applying this equation we thereupon will get a group of scores for all systems. When applying the maximum method to the scores to get the maximum one, then it will be clear that the system which takes the maximum score will be the one which we seek.

2.3 Applicable View (System View)

This section displays how we can apply the *Linear Attribute Model* equation to our evaluation system. Here, we will call the attributes as (the group features), and the evaluation elements or the alternatives as (the LMSs group) which are included in the evaluation operation. We can summarize this algorithm by looking at the table 2.2 and the equation that follows:

Assume we have these variables:

Group of features: F_1, F_2, \dots, F_n Group of LMSs: $LMS_1, LMS_2, \dots, LMS_m$
Weights values: W_1, \dots, W_n Ranks values: R_{11}, \dots, R_{mn}

Table 2.2: The representation of the relations between weights and ranks in the decision making table

<i>Features Group</i>	F_1	F_2	F_j	F_n
<i>Weights</i>	W_1	W_2	W_j	W_n
<i>LMS₁ Ranks</i>	R_{11}	R_{12}	R_{1j}	R_{1n}
<i>LMS₂ Ranks</i>	R_{21}	R_{22}	R_{2j}	R_{2n}
<i>LMS_i Ranks</i>	R_{i1}	R_{i2}	R_{ij}	R_{in}
<i>LMS_m Ranks</i>	R_{m1}	R_{m2}	R_{mj}	R_{mn}

The final equation for whole the evaluation operation will be as follows:

$$\text{MAX}_{i=1}^m \left(\frac{\sum_{j=1}^n (W_j * R_{ij})}{n} \right)$$

The Maximum from $i=1$ to m for the ((Summation from $j=1$ to n for (Weight _{j} * Rank _{ij}) Over n)

Where;
 m : the number of LMSs in the evaluation operation
 n : the number of features in the evaluation operation
 i, j : counters

2.4 Programmatic View

The following pseudo-code will explain this decision making algorithm and its method clearly. By using this pseudo-code, it will be possible to translate into any programming language we want.

Assume:

m = number of LMSs
 n = number of features
 W = array of all weights

```

R = array of all ranks
temp = temporary array

START
Dim m, n As Integer
Dim i, j As Integer
Dim sum As Integer
Dim W [n] As Double
Dim R [m, n] As Double
Dim temp [m, n] As Double
Dim result [m] As Double

Sum = 0
For i=1 To n
    Input W[i]
Next i

For i=1 To m
    For j=1 To n
        Input R[i,j]
        temp[i,j] = W[j] * R[i,j]
    Next j
Next i

For i=1 To m
    For j=1 TO n
        sum = sum + temp[i,j]
    Next j
    result[i] = sum / n
    sum = 0
Next i

Dim MAX As Double = 0
For i=1 To m
    If MAX < result[m] Then
        MAX = result[m]
    Print MAX
    End If
Next i

END

```

2.5 User View

Our evaluation operation contains five (5) steps. Some of them are to insert the entries needed to the algorithm. The number of features (n) will be based on how many features we chose from the list of all features. Our system will select only the LMSs that match the group of features had chosen. In the following, you will find some explanations for the five steps of the evaluation operation:

1. Choosing the group of features you want to be included in your of LMS you seek.
2. The system will check and return a group of LMSs depending on your features selected. Your job here is to select which of LMSs you like to enter in the evaluation operation. You can choose all of them but not less than one system.
3. This step is to weigh each feature inserted in the features group you chose in the first step. Your weights given here will depend on your needs. The weighing method will use the values from 0 to 1. "One" is for the most important feature to you. "Zero" is for the unimportant feature.

4. This step is to rank each feature with each of LMS depending on the features' description given to you from the site. Here, your ranks will also use the values from 0 to 1. "One" when you give the system a full mark to its feature. "Zero" when you see that this feature is very bad to this system.
5. This is the getting results' step. Here, all the mathematical evaluation operations are made. The site will give you a lot of results in this step and the grade of each of LMS inside the evaluation operation.
 - a. The LMS which takes the best grade.
 - b. Summary to all similar evaluation operations done before with the same group of features you chose in the first step.
 - c. Full description for all LMSs included in the operation. Here, you can give your opinion and see all opinions had given before by the visitors or users about these systems. Also, you can discuss a feature for any of LMS in our operation.
 - d. The administrator's advice about the LMS he/she prefers to you, depending on the his/her viewpoint and your needs.

Taking all results shown here in consideration, you can decide which LMS suits you best and fits your type of using. As we see, these steps are very easy to the novice user. This means that the visitor to our site doesn't need any special skills to use it. In the figure below (Figure 2.1), you can see the five steps and the relationship among them:

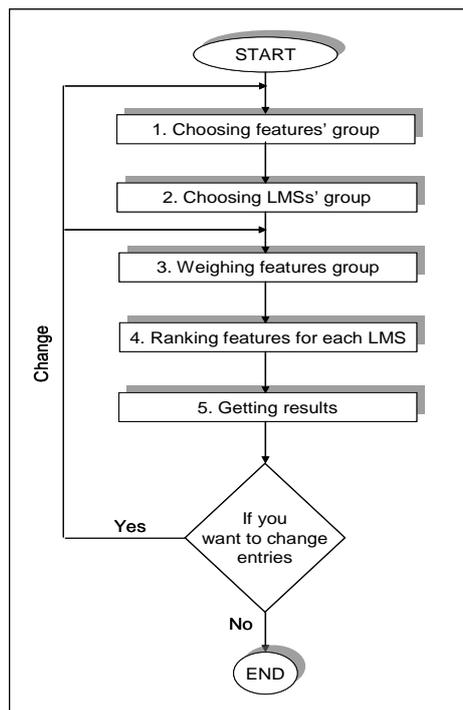


Figure 2.1: The evaluation operation flow-chart

3. Process Modeling:

The most popular technique used in process modeling is Data Flow Diagram (DFD). It is a graphical illustration of data flow between processes and data stores. This system will have two main types of its expected users which are system's administrator and visitors. The Level-0 DFD (figure 2.2) shows the relation between these two main users.

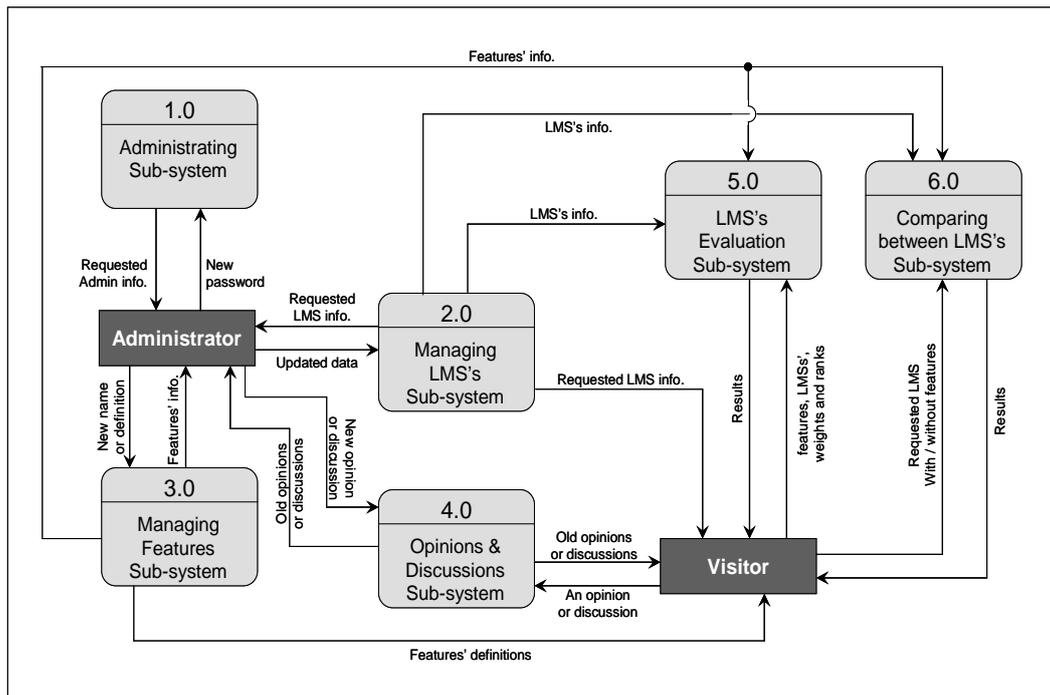


Figure 6.2: Level-0 DFD

From the diagram above, we can note the potential objects for the system:

1. Administrating the site
2. Products information
3. Evaluating sub-system
4. Comparison between products
5. Opinions and discussion

4. Application Programs and Programming Languages

In building this system the author used the Microsoft Visual Studio .NET 2005 (VB.NET, ADO.NET, ASP.NET) in developing and programming and for designing the system's database used the Microsoft SQL Server Management 2005

5. Conclusion

The system uses artificial intelligence methods and decision making algorithms with fuzzy logic comparisons to provide a smart process to help users in making their decision to find a system from a group of choices as a suitable choice for the user. The paper focused on the evaluation for LMSs and the method developed in order to help a user wanting to make a decision for choosing an LMS.

This paper described the decision making process in deferent ways and view point; the logical, mathematical, programmatic, applicable, and the user's view points. The web-based system described in the paper could be considered as a practical and useful solution to the problem of LMS selection.

References

- Anderson E. (1990). Choice Models for the Evaluation and Selection of Software Packages. *Journal of Management Information Systems*, 6(4):123–138, 1990
- Aronson J., Liang T., and Turban E. (2005). *Decision Support Systems and Intelligent Systems* (7th ed.). Prentice Hall.
- ATutor. Learning Content Management Systems. Retrieved January 17, 2010, from <http://www.atutor.ca/>

Blackboard. Learning Management System. Retrieved January 15, 2010, from <http://www.blackboard.com/>

Chavan A. and Pavri S. (December 1st, 2004). Open-Source Learning Management with Moodle. Linux Journal. Retrieved May 2, 2008 from <http://www.linuxjournal.com/article/7478>

Claroline. Learning Management System. Retrieved January 17, 2010, from <http://www.claroline.net/>

Moodle. Modular Object Oriented Dynamic Learning Environment. Retrieved January 15, 2010, from <http://www.moodle.org/>

Turban E. (1995). Decision Support and Expert Systems, Management Support Systems (4th ed.). Prentice Hall.

WebCT. Learning Management System. Retrieved January 15, 2010, from <http://www.webct.com/webct>