A MATHEMATICAL PROGRAMMING APPROACH TO QUANTIFY STRUCTURAL COMPLEXITY OF A CONCEPTUAL MODEL

Tauqeer Hussain
Kutztown University of Pennsylvania
Department of Computer Science
Kutztown, PA, USA
hussain@kutztown.edu

ABSTRACT
Structural complexity is one of the characteristics that determine the quality of a conceptual model. Various attributes have been identified in research as measures of structural complexity. The research however lacks in devising a methodology that can use these measures to quantify structural complexity of a given model to a numeric value. In the absence of such an approach, it is not possible, at least quantitatively, to compare the quality of two given models for the same problem. In this research, we propose a methodology that can be applied to quantify the structural complexity of any type of conceptual model. We have modeled it as a multi-criteria decision problem which is solved in two stages. In the first stage, analytic hierarchy process (AHP) is applied whereas, in the second stage, an optimization technique namely goal programming is used. Using this approach, structural complexity of a conceptual model can be determined numerically. As a result, it also solves the bigger problem which is to compare two conceptual models to identify the one with better quality. We also present an example to demonstrate application of our proposed methodology to one type of conceptual models namely entity-relationship (ER) diagrams.

KEY WORDS
Entity-Relationship model, Structural complexity, Analytic hierarchy process, Goal programming, Conceptual model.

1. Introduction
Research in software engineering has been addressing the quality issues for a very long time. These issues are related to all the phases of systems development life cycle (SDLC). The research shows that the quality at later stages of SDLC is dependent on the quality “built” in early stages [1]. Hence, it becomes very important to devise methodologies, tools and metrics that can determine, monitor and improve quality at early stages. At systems design stage of SDLC, software engineers use various types of conceptual modeling tools; for example, entity-relationship (ER) diagram [2,3], and class diagram in unified modeling language (UML) [4]. They often find alternate designs of a given problem but deciding which design is “superior” is usually very subjective. Researchers have tried to determine the quality of conceptual models but their efforts are usually focussed on qualitative aspects. Some efforts done by [5-7] should be appreciated but they lack providing a basis to compute a numerical value for the quality. As a result, it is not possible to compare two different models, say those given in Figure 1 and Figure 2, and to determine which one is of better quality. The ISO/IEC 9126 standard [8] defines a number of quality attributes of a software system. In this paper, however, we are addressing one quality aspect of conceptual models namely structural complexity.

![Figure 1. A Sample ER Diagram](image1)

![Figure 2. An Alternate ER Diagram](image2)
not quantify the structural complexity of a given ER model. Hence, their approach cannot be used to compare two given models for their quality. There was another problem with their approach. AHP requires a set of decision alternatives which are to be ranked based on given criteria. Piattini et al. [5,10] have presented various constructs of an ER diagram (for example, number of entity types, number of relationship types, etc.) as measures of its structural complexity. These measures are listed in Table 1. [6] used these measures as decision alternatives, as shown in Figure 3. The problem with their approach [6] is that AHP requires its decision alternatives to be independent of one another; whereas, the measures of an ER diagram are not independent. These measures collectively contribute towards the complexity of a model and hence cannot be used individually as decision alternatives.

### Table 1

<table>
<thead>
<tr>
<th>Measures</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NE</td>
<td>Total number of entities within an ER diagram.</td>
</tr>
<tr>
<td>NA</td>
<td>Total number of attributes defined within an ER diagram.</td>
</tr>
<tr>
<td>NR</td>
<td>Total number of relationships in an ER diagram.</td>
</tr>
<tr>
<td>NM:NR</td>
<td>Total number of many-to-many relationships in an ER diagram.</td>
</tr>
<tr>
<td>N1:NR</td>
<td>Total number of one-to-one and one-to-many relationships.</td>
</tr>
<tr>
<td>NN-AryR</td>
<td>Total number of N-Ary relationships in an ER diagram.</td>
</tr>
<tr>
<td>NBinaryR</td>
<td>Total number of Binary Relationships in an ER diagram.</td>
</tr>
</tbody>
</table>

The problems mentioned above motivated us to propose a methodology that can be used to quantify the quality of an ER model. This quantification is to be done at a relative scale because the ultimate objective is to compare different models and to identify the one with the “best” quality. Therefore, no effort is made to compute a number which represents the “absolute” value for quality of a given model. It is worth mentioning here that our proposed approach is not specific to just ER models. It can be applied conveniently for any type of conceptual model.

In the next section of this paper, section 2, we explain the steps of our methodology. Section 3 demonstrates how these steps can be applied to ER models. Section 4 concludes our research and provides pointers to further research.

### 2. Our Methodology

In this section, we propose two stages of our methodology. In stage 1, we define preparatory steps for AHP application and then define those steps where AHP is applied. AHP was developed by Thomas Saaty at Wharton School of Business to solve multi-criteria decision problems where subjective judgments of decision makers are involved. Due to page limit of this paper, AHP is not explained here. The interested reader should see [9] for details.

![Figure 3. Decision alternatives proposed in [6]](image)

The output of stage 1 is an overall priority vector which ranks a set of given conceptual models for their structural complexity. In stage 2, we formulate a mathematical model based on goal programming technique [11, 12] and then solve this model to obtain the relative weights of various measures of a conceptual model. It should be noted that, for any modeling tool, the steps proposed in our methodology are to be carried out only once. After that, the output of stage 2 (that is, relative weights) becomes a constant for the respective modeling tool, and hence can be used to quantify structural complexity of any given model.

#### 2.1 Stage 1

**Step 1: Identify quality attributes for the model**

Every conceptual model may have its own attributes that define its quality. Since we are proposing a general methodology applicable to any conceptual model, we leave it to practitioners and researchers to identify such attributes if they are not already defined. For example, for an ER model, researchers have already defined understandability, analyzability, and modifiability as its quality attributes [5,10].

**Step 2: Define a sample set**

Select a problem domain and design a number of models to define a sample set of varying complexity. For example, one can choose a problem domain and then design, say 10, different ER diagrams. The higher this
number is, the better it is. Note that, as stated earlier, this step is to be performed only once.

**Step 3: Rank models in the sample using AHP**

In this step, the models in the given sample (collected in step 2) are ranked in terms of their complexity. This ranking should be done by a team of experienced professionals who have been using the modeling tool under study for quite some time. Since subjectivity is involved in this ranking, AHP is the most appropriate technique to use.

The quality attributes defined in step 1 are the objectives (or criteria) whereas each model in the sample set (from step 2) is a decision alternative, as shown in Figure 4.

![Figure 4. Hierarchy in AHP](image)

The output of analytic hierarchy process is an overall priority vector which ranks the decision alternatives on a normalized scale. Hence, each element of this vector represents the complexity of the respective decision alternative which is a conceptual model in our case. Now, we are in a position to relate the constructs of a conceptual model (its measures) and the complexity of that specific model quantified in the overall priority vector. This initiates stage 2 of our methodology as explained in the next sub-section.

### 2.2 Stage 2

**Step 4: Preparing for mathematical model**

In this step, we construct a $m \times n$ matrix where $m$ represents number of decision alternatives (that is, number of models, in our case) and $n$ represents total number of constructs of the model that contribute towards goal or objectives (that is, measures of the model, in our case). A general format for this matrix is shown in Table 2. The first column of this table lists model numbers that are there in the sample set whereas the last column has the priorities of respective models from the overall priority vector (step 3). The priority of any model $k$ is shown as $p_k$ in this table. The remaining columns of this table define the matrix. An element of this matrix present in $k^{th}$ row and $j^{th}$ column, denoted by $a_{kj}$, represents number of occurrences of $j^{th}$ construct in $k^{th}$ model, where $k=1, 2, 3, \ldots m$.

Though each construct may have a different weight that shows its contribution towards the complexity of the model, the weight of one particular construct will always remain the same for every model. Therefore, we can define a set of linear equations from Table 2 as follows:

$$
\sum_{k=1}^{n} (a_{tk} w_k) = p_i \quad \forall i=1, 2, 3, \ldots, m
$$

This set of equations presents a system of strict equalities which might not be correct in reality. The reason is that the priority vector is based on subjective ratings of practitioners. Though inconsistency in these subjective ratings is tested in AHP through CR, it still cannot be claimed that the ratings are perfect. It is therefore more appropriate to treat the priorities as approximations. So, we modify our set of equations as:

$$
\sum_{k=1}^{n} (a_{tk} w_k) \approx p_i \quad \forall i=1, 2, 3, \ldots, m
$$

### Table 2

<table>
<thead>
<tr>
<th>Model#</th>
<th>Constructs defined for the conceptual model</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td>1</td>
<td>$a_{11}$</td>
</tr>
<tr>
<td>2</td>
<td>$a_{21}$</td>
</tr>
<tr>
<td>3</td>
<td>$a_{31}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>m</td>
<td>$a_{m1}$</td>
</tr>
</tbody>
</table>

**Step 5: Compute relative weight of each measure using Goal Programming**

It is important to note that the overall priority vector is an approximation for the measurement of structural complexity of each model. This is not an exact value. Hence, we cannot use linear programming to formulate our problem. The difference between exact values and respective approximate values is a deviation and hence the problem can be modeled conveniently using goal programming which is explained below.

Let $m$ be the total number of decision alternatives defined for each criterion, and $P = [p_1, p_2, p_3, \ldots, p_m]$ be the overall priority vector obtained in step 1. It should be noted that for any $p_i \in P$,

$$
0 \leq p_i \leq 1 \quad \forall i=1, 2, 3, \ldots, m, \text{ and } \sum_{i=1}^{m} p_i = 1.
$$

**Decision Variables:** Let $w_j = \text{weight for } j^{th} \text{ measure contributing towards structural complexity of the conceptual model, where } i=1, 2, 3, \ldots, n$

$\mathbf{d}_i = \text{deviation in the priority value from the actual value for } i^{th} \text{ model,}$

**Constants:** $a_{ij} = \text{number of times } j^{th} \text{ construct (measure) occurs in } k^{th} \text{ model, where } k=1, 2, 3, \ldots, n$

$p_i = \text{priority of } i^{th} \text{ model in the overall priority vector where } i=1, 2, 3, \ldots, m$
**Constraints**: A set of goal constraints can now be formulated as:

\[ \sum_{k=1}^{n} (a_{ik} w_k - d_i) = p_i \quad \forall i=1, 2, 3, \ldots, m \]

Since there are \( m \) models, there are \( m \) goal constraints (also known as soft constraints) for this problem.

**Objective Function**: The objective function for a goal programming problem is to minimize the deviations (that is, difference between the calculated value and the actual value) which in mathematical form can be written as:

\[ \text{Min } Z = \sum_{i=1}^{m} d_i \]

The complete goal programming model developed for this problem is given below:

\[ \text{Min } Z = \sum_{i=1}^{m} d_i \]

subject to:

\[ \sum_{k=1}^{n} (a_{ik} w_k - d_i) = p_i \quad \forall i=1, 2, 3, \ldots, m \]

\[ w_i, d_i \geq 0 \quad \forall i=1, 2, 3, \ldots, n \quad \text{(Non-negativity)} \]

This model can be solved using any appropriate optimization software. We have used LINGO [12] for this purpose. The solution to this model, as discussed in the next section, yields the relative weight of each construct which can be used to determine structural complexity of any given conceptual model.

### 3. Application of our proposed approach to determine structural complexity of an ER model

It is important to note that the steps defined for our methodology in Section 2 above are to be carried out only once. The purpose is to obtain the weight of each measure contributing towards the goal. Once these weights are known, the structural complexity (or any other quality characteristic) can be calculated for any given value by plugging in these weights in the following expression:

\[ \sum_{k=1}^{n} c_k w_k \]

where \( c_k \) is the count of each measure in the given model and \( w_k \) is the weight of the corresponding measure found in section 2. As the only thing that will be changing from one model to any other model is \( c_k \) (\( w_k \) is a constant), these computations become trivial. Now, we demonstrate all of these steps with the following example.

Let us say that the conceptual modeling tool is ER diagram and we are interested in computing structural complexity of a given ER model. As stated earlier, weights of the measures of an ER diagram are to be computed only once and then we can use these weights for any ER model.

#### 3.1 Application of Stage 1

**Stage 1 Step 1**: Let us say that we have identified understandability and modifiability as two quality attributes for the structural complexity of an ER model. Hence there are two objectives for our AHP model.

**Stage 1 Step 2**: Let us say that we have defined a sample set of 4 ER diagrams. These diagrams are labelled as ERD1, ERD2, ERD3, and ERD4. Each of these ERDs becomes a decision alternative in our AHP model. The AHP hierarchy is shown in Figure 5.

![Figure 5. AHP Hierarchy for ER model](image)

**Table 6**

<table>
<thead>
<tr>
<th>Weight</th>
<th>Verbal Scale</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Equal importance of both elements</td>
<td>Two ER diagrams contribute equally</td>
</tr>
<tr>
<td>3</td>
<td>Moderate importance of one element over another</td>
<td>Experience and judgment favor one ER diagram over another</td>
</tr>
<tr>
<td>5</td>
<td>Strong importance of one element over another</td>
<td>An ER diagram is strongly favored</td>
</tr>
<tr>
<td>7</td>
<td>Very strong importance of one element over another</td>
<td>An ER diagram is very strongly dominant</td>
</tr>
<tr>
<td>9</td>
<td>Extreme importance of one element over another</td>
<td>An ER diagram is favored by at least an order of magnitude</td>
</tr>
<tr>
<td>2,4,6,8</td>
<td>Intermediate values</td>
<td>Used to compromise between two judgments</td>
</tr>
</tbody>
</table>

In our case, the priority vector for the objectives is straightforward as there is only one pairwise comparison between understandability and modifiability. Say, this comes out to be \([0.4 \ 0.6]\). For a sample set of four ER diagrams, there are two pairwise comparison matrices.
(one for each objective) which are given in Table 7 and Table 8.

<table>
<thead>
<tr>
<th>Table 7</th>
<th>Pairwise Comparison Matrix for Understandability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ERD1</td>
</tr>
<tr>
<td>ERD1</td>
<td>1</td>
</tr>
<tr>
<td>ERD2</td>
<td>1/3</td>
</tr>
<tr>
<td>ERD3</td>
<td>1/5</td>
</tr>
<tr>
<td>ERD4</td>
<td>¼</td>
</tr>
</tbody>
</table>

A software package ExpertChoice [13] was used to enter this data and find the overall priority vector. Figure 6 shows one of the intermediate results by ExpertChoice where priority vector for modifiability is computed by the software. This came out to be:

\[
\begin{bmatrix}
0.6817 & 0.0577 & 0.1051 & 0.1555
\end{bmatrix}
\]

which means that the time required to modify ER diagram 1 would be maximum amongst all alternatives. It should be noted that consistency ratios (CR) for understandability and modifiability turned out to be 0.10 and 0.07 respectively. As none of these values is greater than 0.10, it satisfies the AHP criterion concluding that practitioners’ ratings are reliable. As shown in Figure 7, the overall priority vector calculated by ExpertChoice is:

\[
\begin{bmatrix}
0.6144 & 0.1466 & 0.1179 & 0.1211
\end{bmatrix}
\]

This means that structural complexity of ER diagram 1 in the sample set is the highest and that of ER diagram 3 is the lowest.

To compute the relative weights of ER measures, we proceed to stage 2 of our methodology.

3.2 Application of Stage 2

Stage 2 Step 4: Let us say, we have identified only five measures for structural complexity of an ER diagram, which are:

- NE: Number of regular entity types,
- NWE: Number of weak entity types,
- NBR: Number of binary relationship types,
- NTR: Number of ternary relationship types, and
- NM:N: Number of many-to-many relationship types

Hence, the size of our m x n matrix is 4 x 5. This matrix along with the overall priority vector is given in Table 9.

<table>
<thead>
<tr>
<th>Table 9</th>
<th>Analysis of given Sample Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model#</td>
<td>NE</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

As we have found the weights, computing structural complexity of any given model, or comparing the complexity of two given models is trivial. Just to demonstrate, let us apply these findings to our ER diagrams shown in Figure 1 and Figure 2. The counts for five measures for each of these ER diagrams are given in Table 10. Multiplying these counts by respective weights (0, 0.06055, 0, 0.117692, 0.006375) and adding them up yields the structural complexity which is shown in the last column of Table 10.

<table>
<thead>
<tr>
<th>Table 10</th>
<th>Comparing two ER models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>NE</td>
</tr>
<tr>
<td>Fig. 1</td>
<td>4</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>3</td>
</tr>
</tbody>
</table>

It is therefore concluded that the ER diagram given in Figure 2 is structurally more complex. The screenshots of the software used are shown at the end of this paper in Figures 6-8.

4. Conclusion

In this paper, we have proposed a methodology that can be used to measure structural complexity of any conceptual model. Structural complexity is an important quality attribute and its measurement enables us to compare alternative designs. In our proposed methodology, we have developed a mathematical model based on analytic hierarch process (AHP) and goal programming. This model is generic and can be used for any conceptual modeling tool (for example, ER diagrams...
Figure 6. AHP: priority of Alternatives for Modifiability using ExpertChoice [13]

Figure 7. AHP: Overall priority vector found using ExpertChoice [13]

Figure 8. Goal Programming Model solved by LINGO [12]
or class diagrams). We have demonstrated its application to ER models and have presented an example to compute relative weights of various constructs of an ER diagram. These weights provide the basis to quantify the structural complexity and then compare any number of alternate models.

Future research can be done to carry out experiments by utilizing more input from the practitioners and with reasonable sample size. The proposed methodology can also be studied for other types of conceptual models, for example, class diagrams. Such experiments can lead to determining more reliable weights that can be used to quantify the quality attributes of respective types of models.

References