# Entropy as a Measure of Quality of XML Schema Document

Dilek Basci[1] and Sanjay Misra[2]
[1]BILGI Geographic Information Conversion & Management System, LTD, Turkey
[2]Department of Computer Engineering, Atilim University, Turkey

**Abstract**: *In this paper, a metric for the assessment of the structural complexity of eXtensible Markup Language schema document is formulated. The present metric 'Schema Entropy is based on entropy concept and intended to measure the complexity of the schema documents written in W3C XML Schema Language due to diversity in the structures of its elements. The SE is useful in evaluating the efficiency of the design of Schemas. A good design reduces the maintainability efforts. Therefore, our metric provides valuable information about the reliability and maintainability of systems. In this respect, this metric is believed to be a valuable contribution for improving the quality of XML-based systems. It is demonstrated with examples and validated empirically through actual test cases.*

**Keywords:** *XML Schema documents, W3C XML schema language, complexity metric, entropy, maintainability, and validation.*

*Received December 22, 2008; accepted August 3, 2009*

## 1. Introduction

In recent years, Web services [32, 35] based on XML technologies have been emerging as the de-facto mechanism for exchanging structured information among organizations and in mostly applications. Due to its flexible nature and ease of implementation, XML [11] serves as a ubiquitous, platform-independent data representation and transport format and, hence, has been easily adopted in diverse fields. The data representations are made by designing the schema, which can be written by a series of XML schema languages. Today, the literature in this regard provides a variety of XML schema languages; amongst which the Document Definition Type (DTD) [17], W3C XML Schema [4, 14, 15, 16] and relaxing [13, 21] are the major proposals. Amongst them, W3C XML Schema is a comparatively favored schema language for generating XML documents. This is because XML Schema has the strongest expressive power, and incorporates many commonly-recurring schema constraints in its language specification [26].

The effective and proper implementation of XML in diverse domains requires well-designed XML schemas. From this point of view, the design of XML schemas plays an important role in the software development process and needs to be quantified for the ease of maintainability. Further, schema metrics should be developed–as for software products-to enable the quantification of the schema size, complexity, quality, and other properties. Although XML schemas have been used in diverse fields of software industry and have been playing an important role in many such projects, up to present only few researches have been attempted in terms of the quantification of XML schema documents through applying schema metrics. This has also been an incentive for us to start work in this area. Further, the entropy concept has been applied by many researchers for the assessment of the complexity of software products that are developed by using procedural or OO programming technologies. These measures were developed by adapting Shannon's entropy theory [33] as a measure of uncertainty or variety. Davis and LeBlanc [8] used entropy that was adopted from Shonnon's [33] notion of entropy to assess syntactic complexity of FORTRAN and COBOL code. However, the usage of entropy as a complexity metric has not yet been extended and validated for the assessment of XML Schema documents. In the present paper, we have proposed a metric named 'Schema Entropy (SE) metric' based on entropy concept. The *SE* metric is intended to measure the complexity of XSD document due to diversity in the structures of its elements and is established by following similar approach that was taken by Davis and LeBlanc [8]. We review the available literature for complexity measurement of XSDs in section 2. Our motivation and proposal of new metric is explained in section 3 and 4. The experimentations and empirical validation of SE metric is given in section 5. Lastly, conclusions drawn from this work appear in section 6. In this paper, we have intended to use 'schema' with lower-case's' to denote the general class of schema, and 'Schema' with upper-case 'S' to the schema document written in the W3C XML schema language.

## 2. Existing Measures for XML Schema Document

One can very easily find a number of metrics for software products. Also, there are ongoing efforts [2, 7, 28] in this area to develop comprehensive metrics which can be better evaluators of quality attributes. Although an extensive collection of online articles [12, 18], books, [9, 22] and huge number of publications dealing with software measures for different types of software products is available, only a few researches has been done for the assessment of the quality of the XML schema documents. For example, Sahuguet *et al.* [34] analyzed the typical characteristics of DTD's, and presented count-based measures such as the number of elements, attributes, ID (unique ID name for the attribute) and IDREF entities. 'IDREF' represents attribute values which should match some IDs' (reference to an element). Choi [5] has discussed some criteria about how DTD's should be designed to insure the quality of XML projects. Klettke *et al.* [24] have applied some well- known metrics (on DTDs) that were originally developed for traditional software products such as Line Of Code (LOC), McCabe cyclomatic complexity, Fan-in and Fan-out, Depth of Inheritance Tree (DIT). Mustafa *et al.* [30] have demonstrated that the XML documents that are generated by the DTD with higher nesting levels have higher weights and therefore,are more complicated compared to the documents with lower nesting levels. Basci *et al.* [29] have proposed a metric for DTD which follows a similar approach taken by Davis and LeBlanc [8]. However, no theoretical or empirical validation has been done for the proof of practical applicability of Basci *et al.*'s metric. The first attempt regarding the metrics for XSD's was made by Mc Dowell *et al.* [31] who proposed eleven metrics for XSD's and two formulae that use these metrics to calculate the quality indices for XSD's and the complexity indices to conform XML documents. Theses metrics [31] are mostly related with XSD components' counts such as the number of elements, complex and simple types, annotations, type references, and the unbounded elements definitions or declarations. Lammel *et al.* [25] presented a comprehensive analysis to extract the quantitative and qualitative information from sixty XML Schemas which have been measured through systematic algorithms, on the basis of the intrinsic feature model of the XSD language. Visser [36] has also adopted some well-known existing metrics developed for other software artifacts to XSDs to deal with the structural complexity of XSDs.

## 3. Motivation

In reference [3], a complexity metric C(XSD) was presented for the assessment of the quality of XML Schema. In this proposal, complexity value for a given XSD is evaluated by considering all the complexity of its components. In this evaluation the degree of each components complexity is reflected by a weight value which is assigned based on the components' internal architecture. While assigning a weight value for each component of the Schema document, we observed that the calculation of weight values is easier when most of the Schema components having similar structures appear more frequently and harder when the Schema has mostly diverse-structured elements. In other words, in the Schema document the occurrence of similarly-structured components with high frequency made the calculation easier due to gained familiarity. In addition, it was also easier to understand and remember the structures of the Schema components when most of the Schema elements have the identical structures. Although a number of measures have been developed as shown in section 2 for the assessment of the quality of the Schema documents; none of them has considered the variety in the Schema's elements structure and the resulting complexity. These observations have led to proposing that the repetition of the similarly-structured elements makes *XSD* easier to understand. In order to support our suggestion, we developed the Schema Entropy (SE) metric which is based on the entropy concept from information theory [33]. We believe that the SE metric can differentiate XSDs in terms of their physiological complexities due to the diversity and repetitions in their elements' structures. Since it takes into account the repetition of similar structured elements, SE metric can capture decreasing physiological complexity of XSD due to familiarity gained by navigating the similar structures many times. It is our opinion that the SE metric can be useful to evaluate and maintain the quality of XML Schema document in terms of its maintainability.

## 4. Proposed Metric: Schema Entropy Metric

The SE metric exploits a directed graph representation of a schema document, known as the G(XSD). In section 4.1, we begin by explaining how to represent a given schema document using a directed graph. The proposal of SE metric is given in section 4.2.

### 4.1. Directed Graph Representation of XSD and its Equivalence Classes

The directed graph representation of the Schema document, G(XSD), can be defined as G(XSD) = (N, E), where N is a set of nodes representing the elements of XSD and can be defined as:

$$N = <N_1, N_2, ..., N_m>, m = 1,2...n \quad (1)$$

where *n* is the total number of element definitions in XSD; E is a set of edges that represent parent-child relationships between the elements of XSD. These elements that have no child elements are represented by leaf nodes, and the attributes that an element of XSD has are listed in square shapes and connected to their associated element nodes having circle shapes by straight lines in G(XSD). Any particular node $N_i$ in G(XSD) representing the $i^{th}$ element of XSD, can be identified according to the number of incoming edges [10] from the other nodes to the $N_i$ and the number of outgoing edges originated from $N_i$ to other nodes, fan-in and fan-out [24, 31]), and the number of attributes that an element *i* has. Accordingly, the identification of node $N_i$ in G(XSD) can be provided by the triple:

$$N_i = <fin_i, fout_i, s_i> \quad (2)$$

where $fin_i$ and $fout_i$ are the counts of its incoming and outgoing edges, and $s_i$ is the number of attributes of the element of XSD represented by $N_i$. Note that the total *fan-in* value is always equal to the total *fan-out* value of G(XSD). Any two elements *k* and *m* of XSD represented by the nodes $N_k$ and $N_m$ in G(XSD) are equal in structure and have the same complexity only if $fin_k = fin_m$ $fout_k = fout_m$, and $s_k = s_m$.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="books">
  <xsd:complexType>
   <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="book">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="author">
         <xsd:complexType>
          <xsd:sequence>
           <xsd:element name="firstname" type="xsd:string"/>
           <xsd:element name="lastname" type="xsd:string"/>
          </xsd:sequence>
         </xsd:complexType>
       </xsd:element>
       <xsd:element name="publisher">
         <xsd:complexType>
          <xsd:sequence>
           <xsd:element name="firstname" type="xsd:string"/>
           <xsd:element name="lastname" type="xsd:string"/>
          </xsd:sequence>
          <xsd:attribute name="email" type="xsd:string"/>
         </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="ISBN" type="xsd:positiveInteger"/>
      <xsd:attribute name="date" type="xsd:date"/>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

Figure 1. Listing 1 for the schema documents books.xsd.

As an example, consider the Schema document books. xsd shown in Figure 1 (Listing 1) and its directed graph depicted in Figure 2. From Figure 2, it can be observed that not all the nodes have the same structures- i.e., the number of their incoming and outgoing edges- and the number of attributes is not equal. For instance, the node representing the author element has been identified by <1, 2, 0> and the node representing the publisher element by <1, 2, 1>. Although both elements have equal *fan-in* and *fan-out* values, they do not have same number of attributes. Hence, these two nodes, i.e., the Schema element definitions- can be distinguished by the difference in their structures.

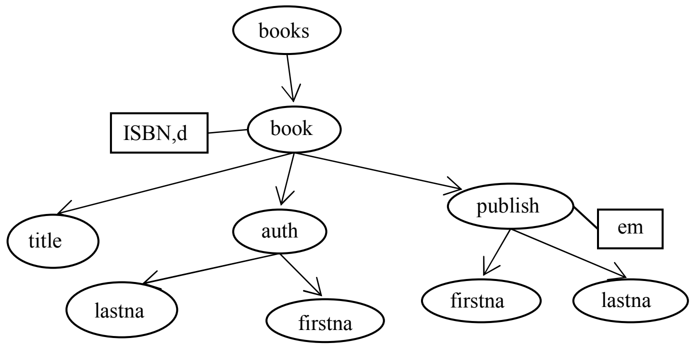

Figure 2. The directed graph representation of the schema books.xsd given in Listing 1.

```
<?xml version="1.0" encoding="UTF-8"?>
 <xs:schema attributeFormDefault="unqualified"
 elementFormDefault="qualified"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="projects">
   <xs:complexType>
     <xs:sequence>
      <xs:element name="project">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="scripts">
          <xs:complexType>
           <xs:sequence>
             <xs:element maxOccurs="unbounded" name="script">
              <xs:complexType>
               <xs:sequence>
                 <xs:element name="language" type="xs:string" />
               </xs:sequence>
               <xs:attribute name="scriptname" type="xs:string" />
              </xs:complexType>
             </xs:element>
           </xs:sequence>
          </xs:complexType>
         </xs:element>
        <xs:element name="namespaces">
         <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" name="namespace"
type="xs:string" />
           </xs:sequence>
           <xs:attribute name="url" type="xs:string" />
         </xs:complexType>
        </xs:element>
       <xs:element name="singletons">
         <xs:complexType>
          <xs:sequence>
           <xs:element maxOccurs="unbounded" name="singleton">
            <xs:complexType mixed="true">
             <xs:attribute name="createdBy type="xs:string" />
            </xs:complexType>
           </xs:element>
         </xs:sequence> </xs:complexType>
       </xs:element> </xs:sequence>
       </xs:complexType> </xs:element>
     </xs:sequence>   </xs:complexType>
     </xs:element>   </xs:schema>
```

Figure 3. Listing 2 for schema document projects.xsd.

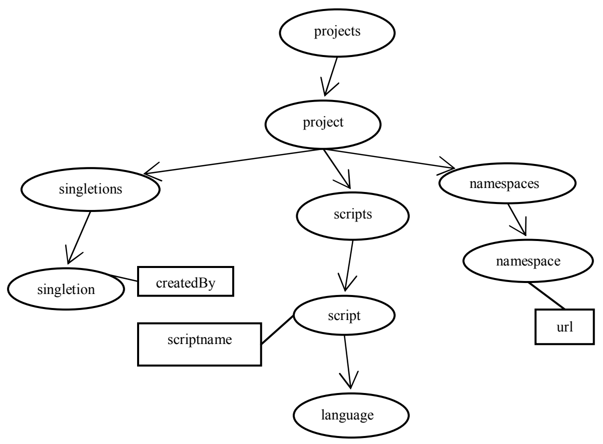

Figure 4. The directed graph representation of the Schema *projects.xsd* given in Figure 3.

The nodes in G(XSD*)* and the elements have been distinguished accordingly in terms of the difference in their structures; the nodes having similarly-structures are grouped into the same equivalence classes. More formally, an equivalence class C having m number of elements can be defined as a set of similarly-structured element nodes in G(XSD), i.e., C = <$N_0$, $N_1$,…$N_m$>, where the element nodes $N_0$=$N_1$ =$N_{2...}N_{m-1}$ =$N_m$. The number of equivalence classes reflects the number of unique element structures in XSD, and the member counts of each class reflects the number of occurrences of each class member. Each element that belongs to the same class has the same structure. That is to say that their *fan-in*, *fan-out,* and the number of attributes are equal but the elements belonging to the different classes have distinct structures. For instance, the equivalence classes of books.xsd given in Figure 1, Listing 1, and projects.xsd given in Figure 3 (Listing 2) have been extracted from their corresponding graphs (Figures 2 and 4) and are listed in Listing 3 and Listing 4, respectively. Note that both of the Schemas have an equal number of elements, which is nine, and while books.xsd has five distinct structured elements reflected by the number of its class, it is six for projects.xsd. In Listing 3, the class $C_3$ has five elements that are equal in structure and the remaining classes have only one distinct structured element. This means that one element structure appears five times in books.xsd and has a higher frequency among the other four distinct element structures. The frequencies of each distinct structured element of books.xsd are 1, 1, 5, 1, and 1. In Listing 4, the member counts of class $C_3$ is three and that of class $C_4$ is two, and all the remaining classes have only one distinct structured element. As a result the frequencies of each distinct structure of projects.xsd are 1, 1, 3, 2, and 1.

*Listing 3*: *$C_1$= {books}, C2= {book}, $C_3$= {title, firstname, lastname,firstname,lastname},*
*$C_4$ = {author}, C5 = {publisher}.*

*Listing 4: $C_1$= {projects}, $C_2$= {project}, $C_3$= {singletions, namespaces, scripts},*
*$C_4$ = {singletion, namespace}, C5 = {script}, C6 = {language}.*

## 4.2. Definition of the SE Metric

By following similar approach taken by Davis and LeBlanc [8] and based on their definitions we applied entropy metric to measure complexity of XSD files. In our case the declarations of the components of XSD can be seen as"chunks" since these components can be related with the other elements in the form of parent-child relationship that can be shown in a directed graph representation of Schema document G(XSD). In G(XSD) while *fan-in* [10] metric's value gives information about how many times an element node is referenced by the other element nodes, i.e. the number of parent nodes of a child node. The *fan-out* metric gives the number of element nodes that a particular element node depends on, i.e. number of child nodes. While low *fan-out* for any parent node can be interpreted as low dependency on a few child nodes, high *fan-in* for any child node can be interpreted that many parent nodes are dependent on that child node. It is clear that having high *fan-in* for a particular child node implies that that child node has impact on its parent nodes since these parent nodes will be affected by any modification made in the child node. On the other hand, for any particular node, zero *fan-in* and *fan-out* means that neither it has effect on any other nodes and nor is affected by the other nodes. Hence, both *fan-in* and *fan-out* metrics can measure the dependency between nodes and the nodes that have the equal *fan-in*, *fan-out* values and equal number of attributes can be treated as equal in terms of their structural complexities. Any two elements k and m of XSD represented by the nodes $N_k$ and $N_m$ in G(XSD) are equal in structure and have the same complexity only if $fin_k = fin_m$, $fout_k = fout_m$ and $s_k = s_m$ .

Based on the entropy definition [8] the entropy of a given Schema document having *n* distinct class ($C_n$), of elements can be calculated using relative frequencies as unbiased estimates of their probabilities $P(C_i)$, i=1,2,...,n. The relative frequency of occurrence of the equivalence classes of the Schema document is the number of elements inside the equivalence class divided by the total number of elements in the Schema document. Accordingly, the SE metric is defined as:

$$SE = -\sum_{i=1}^{n} P(C_i)\log_2 P(C_i) \tag{3}$$

where *n* is the number of distinct classes.

As an example, for projects.xsd the relative frequency of occurrence of its equivalence class $C_3$= {singletions, namespaces, scripts} is $P(C_3)$ = 3/9. When all elements of XSD are placed in the same equivalence class the minimum entropy value is evaluated. Since there is only one class, i.e. n=1, and all elements are grouped in this class the relative frequency of occurrence of that class, P(C$_1$)=1, and *SE* value is:

$$SE = -\sum_{i=1}^{1} P(C_i)\log_2 P(C_i) = P(C_i)\log 2\, P(C_i) = 0 \qquad (4)$$

On the other hand the possible maximum entropy occurs when each element of XSD has different fan-in*,* fan-out values and number of attributes, i.e., each has different structure. The SE value of that schema, in this case is:

$$SE = -\sum_{i=1}^{n} P(C_i)\log_2 P(C_i) = -\sum_{i=1}^{1} (1/n)\log_2 P(C_i) = \log_2 n \qquad (5)$$

## 5. Experimentations and Validation

In order to demonstrate the *SE* metric we employ the same example of Schema documents given in Listing 1 and 2 as shown in Figures 1 and 3, and their directed graph representations in Figures 2 and 4.

The SE values for the schemas books.xsd, and projects.xsd, are calculated by considering their corresponding equivalence classes given in Listing 3 and 4. The counts of distinctly-structured elements appearing in the graph representations of these Schemas are equal to the number of their equivalence classes, and are found 5 and 6, respectively. Hence,

$$SE_{books.xsd} = -\sum_{i=1}^{5} P(C_i)\log_2 P(C_i) \qquad (6)$$

$$=(1/9)*\log_2(1/9)+(1/9)*\log_2(1/9)+(5/9)*\log_2(5/9)+(1/9)*\log_2(1/9)+(1/9)*\log_2(1/9) = 1.87996$$

$$SE_{projects.xsd} = --\sum_{i=1}^{6} P(C_i)\log_2 P(C_i) \qquad (7)$$

$$= (1/9)*\log_2(1/9) + (1/9)*\log_2(1/9) + (3/9)*\log_2(3/9) + (2/9)*\log_2(2/9)+(1/9)*\log_2(1/9)+(1/9)*\log_2(1/9) = 2.41938$$

As can be seen from the above example although both example XSDs have the same number of element definitions they do not have equal SE values. The schema documents with lower SE tend to be dominated by fewer distinct structured elements that have higher frequency of occurrences. In books.xsd the number of equivalence classes reflecting number of unique structures is five and, in projects.xsd it is six. These numbers imply that compared with books.xsd the Schema projects.xsd has more diversity in its elements' structures. Further, by looking the class C$_3$ of books.xsd it can be observed that five elements has equal in structure, that is the same structure appears five times. On the other hand, in projects.xsd maximum repetition of similar structured elements is three which is the member counts of class C$_3$. It will be more obvious that books.xsd has less diversity when we calculate the ratio of the number of distinct i.e. unique structured elements to total elements of XSD; we find it as 5/9 for books.xsd and 6/9 for projects.xsd. From these ratios we can guess which XSD will have lower *SE*. As larger XSDs in terms of elements' count generally have more elements, SE measure may tend to be lower. However, this reflects the notion that larger XSDs are able to contain more repetition, and so SE reflects this. It is likely that real world XSDs will not have *SE* measurements approaching minimum value which is 0 and maximum value log$_2$(n), where *n* is the total number of elements of XSD. As a consequence, *SE* metric can be useful in comparing XSDs having equal number of elements. As stated earlier the structural varieties of elements declared in Schema has not been captured by any existing Schema measures. Therefore, usefulness of SE metric can be verified by comparing it with the other XSD complexity measures such as element fanning [31, 36], number of elements [25]. The element fanning for a given Schema is calculated by Fanning = e/n where e is the number of edges i.e., totals fan-in or fan-out and n is the number of nodes in the directed graph of the Schema. The fan-in and fan-out measure total level of information flow between individual modules and the rest of the system. The higher fanning value for a Schema can be interpreted as elements are highly connected, i.e., dependent to each other thus modification made in any individual element will update the other element to which that individual element is connected. The #E [25] metric is a kind of size metric which measures the total number of local and global element definitions/declarations in XSD.

From Table 1, it can be observed that the values of complexity measures, the *fanning*, *#E* and C(XSD) are consistent to each others. However, SE metric has the different values for the two Schemas as a measure of their complexities. It is due to the fact; the *SE* metric considers the diversity in the structures of each element appearing in the graph representation of Schema document and their frequencies. The Schema documents that exhibit greater variety in structures of elements with less frequency of occurrences have the greater value of SE than the Schema documents that exhibit less variety in the structures of elements with high frequency of occurrences, i.e., more repetitions of similar structured elements. Since the high frequency of similar structured elements makes the developer more familiar to the Schema structure overall understandability of Schema document becomes much easier. This was neglected by the other compared complexity metrics.

The SE metric is also useful for comparing XSDs that conforms the same resulting XML documents. Consider for example the Schema document books2.xsd (Figure 5, listing 6) which is the modified version of books.xsd and, its graph representation is depicted in Figure 6. In this modified version we defined a reusable global elements group names. By giving reference to this global elements group we defined the firstname and lastname elements' within author and publisher elements. The equivalence classes of modified version books2.xsd are:

*Listing5:* $C_1=\{books\}$, $C_2=\{book\}$, $C_3=\{firstname, lastname\}$, $C_4=\{author\}$, $C_5=\{publisher\}$, $C_6=\{title\}$

Accordingly, the SE value for books2.xsd is:

$$SE_{\text{books2.xsd}} = -\sum_{i=1}^{6} P(C_i)\log_2 P(C_i) \qquad (8)$$

$$=(1/7)*\log_2(1/7)+(1/7)*\log_2(1/7)+(2/7)*\log_2(2/7)$$
$$+(1/7)*\log_2(1/7)+(1/7)*\log_2(1/7)+(1/7)*\log_2(1/7)$$
$$= 2.80735$$

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="books">
  <xsd:complexType>
   <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="book">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="title" type="xsd:string"/>
       <xsd:element name="author">
        <xsd:complexType>
         <xsd:sequence>
          <xsd:group ref="names"/>
         </xsd:sequence>
        </xsd:complexType>
       </xsd:element>
       <xsd:element name="publisher">
        <xsd:complexType>
         <xsd:sequence>
          <xsd:group ref="names"/>
         </xsd:sequence>
         <xsd:attribute name="email" type="xsd:string"/>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="ISBN" type="xsd:positiveInteger"/>
      <xsd:attribute name="date" type="xsd:date"/>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
<xsd:group name="names">
         <xsd:sequence>
          <xsd:element name="firstname" type="xsd:string"/>
          <xsd:element name="lastname" type="xsd:string"/>
         </xsd:sequence>
</xsd:group>
</xsd:schema>
```

Figure 5. Listing 6 for schema document books2.xsd, a modified version of books.xsd as shown in Figure 1 Listing.1.

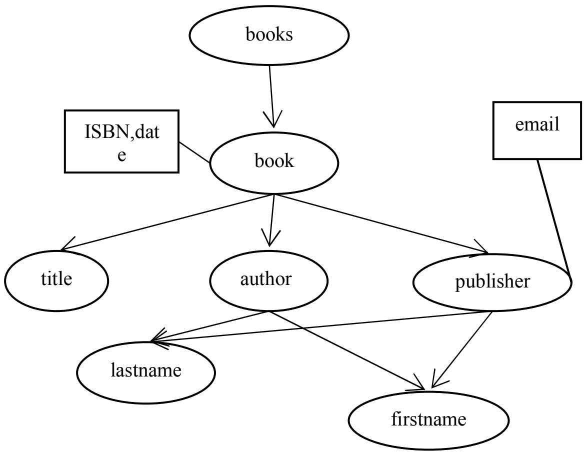

Figure 6. The directed graph representation of books2.xsd shown in Listing 6.

The values of complexity measures for this new design of Schema document books2.xsd are shown in the last line of table 1. When the ratio of distinct structured elements to the total number of elements appearing in the graph representation of books2.xsd is calculated it is found as 6/7 which is greater than 5/9, the ratio calculated for books.xsd, thus it is expected that the complexity of books2.xsd will increase since the number of distinct structured elements increases, i.e., diversity in structures is higher. Note that the ratio does not consider the repetition of similar structured elements. As can be observed from this table, by adding reusable component to the modified version of books.xsd its complexity value is increased since repetition of similar structured elements is decreased and variety in structures is increased. The side effect of increasing number of reuse of same components is that increasing number of affected components that use the same reusable component. That is, when we modify the definitions of firstname and lastname elements and make them global, both author and publisher elements of books2.xsd will be affected since the number of reuse for firstname and lastname will be two in this case. On the contrary, when firstname and lastname elements are declared locally inside author and publisher elements as declared in books.xsd, the number of their use is only one; hence any modification made on one of these pair only affects one of the associated parents. As a consequence, we may suggest that increasing reusability in XSD components may result in increasing complexity due to increasing number of affected components.

Table 1. The values of complexity measures developed for XSDs.

| .Schema | Listig | Graph Fig. No | Fanng | #E | C(XSD) | SE |
|---|---|---|---|---|---|---|
| **Books.xsd** | 1 | 2 | 8/9 | 9 | 12 | 1.8796 |
| **Projects.d** | 2 | 4 | 8/9 | 9 | 12 | 2.4198 |
| **Books2.xd** | 6 | 6 | 8/7 | 7 | 12 | 2.8075 |

## 5.1. Empirical Validation of the SE Metric

Empirical validation is the only way through which the academician and scientist can assist industry in selecting new technology. In addition, there are several evaluation criteria [1, 23] which evaluate software metrics theoretically, but either they are under criticism [6] or they do not fulfil the requirement of real validation. Only empirical validation is the way to prove the practical usefulness of the metric. For empirical validation of the SE metric, we have analyzed forty actual Schemas.

Most of the analyzed Schemas were extracted from the Web Service Description Language (WSDL) [20, 32, 35] documents and downloaded from well known web sites, such as xmethods.com, webservicex.net which provides links to Web services. The statistics we have collected after analyzing these Schemas to evaluate the SE measure are shown in Table 2. We have also calculated the values of the #E, and the Fanning metrics for compared it with SE metric.

Table 2. The analyzed Schema documents and measures.

| ID | #E | SE | FANNING | WEB LINK |
|---|---|---|---|---|
| 1 | *8* | 1.299 | 0.750 | http://www.thomas-bayer.com/axis2/services/BLZService?wsdl |
| 2 | 8 | 1.436 | 0.375 | http://www.elguille.info/NET/WebServices/HolaMundoWebS.asmx?WSDL |
| 3 | 12 | 1.252 | 0.667 | http://in2test.lsi.uniovi.es/sqlmutationws?WSDL |
| 4 | *17* | 1.902 | 0.588 | http://soap.einsteinware.com/nascar/nascardataservice.asmx?WSDL |
| 5 | *17* | 2.278 | 0.765 | http://ws.netedgesoftware.com/wsenabler/1.0/StockInfoCS.asmx?WSDL |
| 6 | *18* | 1.352 | 0.556 | http://webservices.daelab.net/temperatureconversions/TemperatureConversions.wso?WSDL |
| 7 | 18 | 1.723 | 0.556 | http://www.devhood.com/services/timelog/timelog-service.asmx?WSDL |
| 8 | *19* | 1.578 | 0.579 | http://rangiroa.essi.fr:8080/dotnet/evaluation-cours/EvaluationWS.asmx?WSDL |
| 9 | 19 | 2.715 | 0.579 | http://www.multispeak.org/interface/30j/10_OA_EA.asmx?WSDL |
| 10 | 20 | 1.522 | 0.400 | http://www.wubingstudy.com/WebService/Messages.asmx |
| 11 | *22* | 1.730 | 0.636 | *http://services.nirvanix.com/ws/Authentication.asmx?WSDL* |
| 12 | *25* | 1.940 | 0.720 | http://www.golemproject.com/Apps/96/Generator.asmx?WSDL |
| 13 | *29* | 1.760 | 0.586 | http://services.argosoft.com/AddressValidation/AddressVerifier.asmx?WSDL |
| 14 | *30* | 1.273 | 0.533 | http://www.mathertel.de/AJAXEngine/S02_AJAXCoreSamples/CalcService.asmx?WSDL |
| 15 | 30 | 1.711 | 0.533 | http://services.test.musiccue.net/rapidcueapplication/WorkManager.asmx?WSDL |
| 16 | *37* | 2.667 | 1.000 | *http://ws.strikeiron.com/MidnightTraderFinancialNews?WSDL* |
| 17 | 37 | 1.363 | 0.730 | http://webservices.freshegg.com/resources/service1.asmx?WSDL |
| 18 | 37 | 1.724 | 0.757 | http://www.cts.com.pl/webservices/rt_info.asmx?WSDL |
| 19 | *41* | 2.160 | 0.512 | http://gemlca.cpc.wmin.ac.uk/GLCProcess?WSDL |
| 20 | 42 | 1.462 | 0.881 | http://quisque.com/fr/chasses/blasons/search.asmx?WSDL |
| 21 | *43* | 2.700 | 0.953 | http://www.esendex.co.uk/secure/messenger/soap/InboxService.asmx?WSDL |
| 22 | *44* | 2.289 | 0.727 | http://www.oorsprong.org/websamples.arendsoog/ArendsoogbooksService.wso?WSDL |
| 23 | 45 | 1.267 | 0.933 | http://service.ecocoma.com/shipping/fedex.asmx?WSDL |
| 24 | 45 | 3.082 | 0.844 | ttp://pc218.cgk.affrc.go.jp/PMTypeService/MainEntry.asmx?WSDL |
| 25 | *46* | 2.189 | 1.370 | http://trial.serviceobjects.com/pa/phoneappend.asmx?WSDL |
| 26 | *47* | 1.742 | 0.660 | http://del.eterio.us/blog/editposts.asmx?WSDL |
| 27 | *48* | 2.123 | 0.917 | http://ws.cisa.ca/WehireWS/JobsWs.asmx?WSDL |
| 28 | *52* | 1.513 | 0.846 | http://www.geoservicios.com/V2.0/sgeo/sgeo.asmx?WSDL |
| 29 | *54* | 1.769 | 0.630 | http://api.legiomedia.com/Content.asmx?WSDL |
| 30 | 60 | 1.919 | 0.800 | http://svc.exaphoto.com/eXaPhoto/CollectionServices.asmx?WSDL |
| 31 | 60 | 2.579 | 1.083 | http://itplaza.jeju.go.kr/rpt_ws/Rpt_Ws_FD.asmx?WSDL |
| 32 | *63* | 2.634 | 1.159 | http://ws.strikeiron.com/ReverseResidentialLookup?WSDL |
| 33 | *65* | 2.693 | 1.077 | http://www.banguat.gob.gt/variables/ws/BDEF.asmx?WSDL |
| 34 | 68 | 1.858 | 0.882 | http://demo.soapam.com/services/FedEpayDirectory/FedEpayDirectoryService?WSDL |
| 35 | 83 | 1.873 | 0.867 | http://www.sipeaa.it/wset/ServiceET.asmx?WSDL |
| 36 | 83 | 1.474 | 0.590 | http://hooch.cis.gsu.edu/bgates/MathStuff/Mathservice.asmx?WSDL |
| 37 | *84* | 2.845 | 1.536 | http://www.esendex.com/secure/messenger/soap/ContactService.asmx?WSDL |
| 38 | *84* | 3.053 | 1.369 | http://www.xignite.com/xNews.asmx?WSDL |
| 39 | *111* | 2.936 | 1.820 | http://www.xignite.com/xwatchlists.asmx?WSDL |
| 40 | *153* | 1.447 | 0.902 | http://service.test.cdream.com.cn/CernetForSP/CernetInterfaceForSP.asmx?WSDL |

The graphs depicted in Figure 7 and 8 show the comparison results between #E and SE metrics and, between fanning and SE, respectively. We have assigned each Schema document with id numbers for the sake of clarity and the links for these documents are also provided in the Table 2. It can be observed from Table 2 and Figure 7 the Schema documents that have equal #E metric values can be differentiated by SE metric in terms of their complexities. As #E metric does not reflect variety in elements structures the Fanning metric evaluates different measures for those Schemas having equal number of elements hence they can also be ordered according to this metric. However, the graphs of those Schemas that have lower SE measure tend to exhibit more regularity due to higher frequencies of similar structured elements, thus they are easy to comprehend since the developer gains familiarity. On the other hand while some of the element fanning measures in Table 2 is equal or very close to each other for the Schema documents (for example Schemas with id numbers 6, 7, 8, 9 and 14, 15) the SE metric evaluates different values for them.

The graph depicted in Figure 8 shows the comparison result between element fanning and SE metrics.

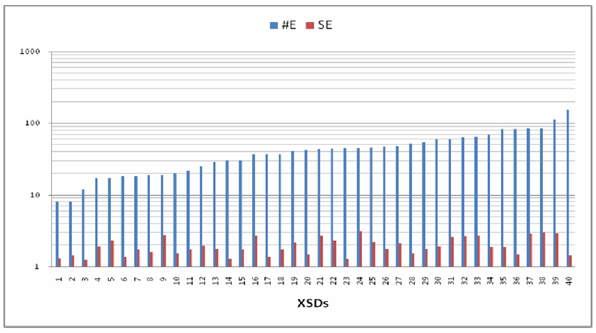

Figure 7. SE vs. #E, the data is ordered by #E measures.

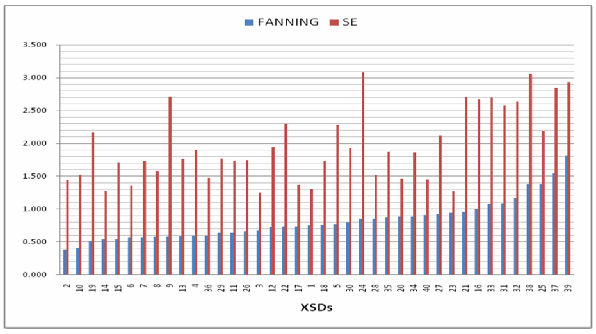

Figure 8. The element fanning vs. the DSERS. The data is ordered by element fanning measures.

## 6. Conclusions

With the successful design of the schemas, the developers can have the capability of increasing productivity, improving software reliability, minimizing development time, and decreasing time-to-market [4]. The SE metric proposed in this paper is an attempt to achieve these goals. This complexity metric is based on the entropy concept from Shannon's information theory for measuring the physiological complexity of a given Schema document due to the diversity in its elements' structure. It was found that measuring the complexity of the schema documents by entropy metric, SE, can provide a useful feedback when comparing Schemas documents that have equal number of elements. The graph representation of the Schema documents that have more similarly-structured elements with higher frequencies of occurrences exhibit more regularity, thus are easy to grasp because of gained familiarity. An analogy with this situation can also be observed when we intended to investigate a binary tree and an irregular tree. It is obvious that understanding the structure and the relation between the nodes of a binary tree is easier than that of an irregular tree. In this regard, the SE metric provides more information about the understandability of the Schema documents. It is obvious that less understandable Schemas require more maintenance efforts. Therefore, the SE provides valuable information about maintainability. The SE metric may also be used to reflect the reusability of Schema components. One may also use entropy concept to measure diversity in type definitions in Schema documents, in order to compare Schemas that have equal number of complex type definitions.

## References

[1] Briand L., Morasca S., and Basily V., "Property Based Software Engineering Measurement," *Computer Journal of IEEE Transactions on Software Engineering,* vol. 22, no. 1, pp. 68-86, 1996.

[2] Boxall M. and Araban S., "Interface Metrics for Reusability Analysis of Components," *in Proceedings of the 2004 Australian Software Engineering Conference*, pp. 1-40, Australia, 2004.

[3] Basci D. and Misra S., "Measuring and Evaluating Design Complexity Metric for XML Schema Document," *Computer Journal of Information Science and Engineering*, vol. 25, no. 5, pp.1-21, 2009.

[4] Binstock C., Peterson D., Smith M., Wooding M., Dix C., and Galtenberg C., *The XML Schema Complete Reference*, Addison Wesley Professional Publishers, 2002.

[5] Choi B., "A Few Tips for Good XML Design," *Technical Report*, University of Pennsylvania, http://db.cis.upenn.edu/˜kkchoi/DTDI2/,2000

[6] Cherniavsky J. and Smith C., "On Weyuker's Axioms for Software Complexity Measures," *Computer Journal of IEEE Transactions on Software Engineering*, vol. 17, no. 6, pp. 636-638, 1991.

[7] Chidamber S. and Kemerer C., "A Metric Suite for Object Oriented Design," *Computer Journal of IEEE Transactions on Software Engineering*, vol. 20, no. 6, pp. 476-493, 1994.

[8] Davis J. and LeBlanc R., "A Study of the Applicability of Complexity Measures," *Computer Journal of IEEE Transactions on Software Engineering*, vol. 14, no. 2, pp. 366-372, 1988.

[9] Fenton N., *Software Metrics: A Rigorous Approach*, Chapman and Hall, 1991.

[10] Henry S. and Kafura K., "Software Structure Metrics Based on Information Flow," *Computer Journal of IEEE Transactions on Software Engineering*, vol. 7, no. 5, pp. 510-518, 1981.

[11] http://www.w3.org/XML/, Last Visited 2008.

[12] http:// ivs. cs. uni- magdeburg. de/sw- eng/ us/ metclas /index.shtml, Last Visited 2008.

[13] http://www.xml.gr.jp/relax, Last Visited 2008.

[14] http://www.w3. org/TR /2004/REC- xmlschema-1-20041028/, Last Visited 2008.

[15] http://www.w3. org/TR /2001/PR- xmlschema-0-20010330/, Last Visited 2008.

[16] http://www.w3. org/TR /2004/REC- xmlschema-2-20041028/, Last Visited 2008.

[17] http://www.w3.org/TR/1998/REC-xml-19980210 , Last Visited 2008.

[18] http://www.xfront.com/GlobalVersusLocal. html, Last Visited 2008.

[19] http://www.oreillynet.com/xml/blog/2006/05/metrics_for_xml_projects_1_ele.html, Last Visited 2008.

[20] http://www.w3.org/TR/wsdl, Last Visited 2008.

[21] ISO/IEC: Information Technology – Text and Office Systems – Regular Language, Description for XML (RELAX) – Part 1: RELAX Core,.D TR 22250-1, 2000.

[22] Kan S., *Metrics and Models in Software Quality Engineering*, Addison-Wesley, 2003.

[23] Kaner C., "Software Engineering Metrics: What Do They Measure and How Do We know," *in Proceedings 10th International Software Metrics Symposium*, pp. 1-10, USA, 2004.

[24] Klettke M., Scneider L., and Heuer A., "Metrics for XML Document Collections," *in Proceedings XMLDM Workshop*, Czech Republic, pp. 162-176, 2002.

[25] Lammel R., Kitsis S., and Remy D., "Analysis of XML Schema Usage," *in Proceedings XML*, Atlanta, pp. 1-39, 2005.

[26] Lee D. and Chu W., "Comparative Analysis of Six XML Schema Language," *ACM SIGMOD Record*, vol. 29, no. 3, pp. 1-12, 2000.

[27] Lin Z., He B., and Choi B., "A Quantitative Summary of XML Structures," *Computer Journal of Lecturer Notes in Computer Science*, vol. 4215, no. 4, pp. 228-240, 2006.

[28] Misra S. and Akman I., "Weighted Class Complexity: A Measure of Complexity for Object Oriented Systems," *Computer Journal of Information Science and Engineering*, vol. 24, no. 2, pp. 1689-1708, 2008.

[29] Misra S. and Basci D., "Entropy Metric for XML DTD Documents," *in Proceedings of ACM SIGSOFT Software Engineering Notes,* pp. 1-6, vol. 33, 2008.

[30] Mustafa Q. and Smadzadeh M., "Determining the Complexity of XML Documents," *in Proceedings The International Conference on Information Technology: Coding and Computing (ITCC'05)*, USA, pp. 416-421, 2005.

[31] McDowell A., Schmidt C., and Yue K., "Analysis and Metrics of XML Schema," *in Proceedings the International Conference on Software Engineering Research and Practice*, USA, pp. 538-544, 2004.

[32] Newcomer E. and Lomow, G., *Understanding SOA with Web Services*, Addison Wesley Professional, 2004.

[33] Shannon C. and Weaver W., *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, 1949.

[34] Sahuguet A., "Everythink You Ever Wanted to Know About DTDs, but Were Afraid to Ask," *Lecture Notes in Computer Science*, vol. 1997, pp. 171-183, 2000.

[35] Thomas E., *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, Prentice Hall Publishers, 2004.

[36] Visser J., "Structure Metrics for XML Schema," *in Proceedings of XATA*, Portugal, pp. 1-12, 2006.

**Dilek Basci** completed her bachelor and master degrees in computer engineering in 2004 and 2008 from Atilim University Ankara, Turkey, respectively. Presently, she is working as software specialist in BILGIGIS, a software company in Ankara. Her area of interest are object oriented design and programming, API design, database management, geographic information systems, internet programming, web technologies, GML, XML technologies and SOA. She is a motivated researcher and has produced some very good results in recent research papers.

**Sanjay Misra** is professor in FUT, Minna. Presently, he is working in the area of software engineering, especially on software quality estimation through software metrics. His area of interests are software measurement, verification and validation techniques, object oriented technologies, XML, web services, and cognitive informatics. He published more than 40 research papers in these areas. Presently, he is chief editor of International Journal of Computer Science and Software Technology (IJCSST). He is also reviewer of several international journals and chaired several workshops and special sessions of international conferences.