# Measuring the Reusable Quality for XML Schema Documents

**Tinzar Thaw**

University of Computer Studies, Mandalay, P.O. Box: 73 22, Vientiane Lao PDR Patheingyi Township, Mandalay, Myanmar; e-mail: tinzar.t@gmail.com

**Sanjay Misra**

Department of Computer Engineering, Atilim University, Kizilcasar Mahallesi, 06836 Incek Golbasi, Ankara, Turkey; e-mail: smisra@atilim.edu.tr

*Abstract: eXtensible Markup Language (XML) based web applications are widely used for data describing and providing internet services. The design of XML schema document (XSD) needs to be quantified with software with the reusable nature of XSD. This nature of documents helps software developers to produce software at a lower software development cost. This paper proposes a metric Entropy Measure of Complexity (EMC), which is intended to measure the reusable quality of XML schema documents. A higher EMC value tends to more reusable quality, and as well, a higher EMC value implies that this schema document contains inheritance feature, elements and attributes. For empirical validation, the metric is applied on 70 WSDL schema files. A comparison with similar measures is also performed. The proposed EMC metric is also validated practically and theoretically. Empirical, theoretical and practical validation and a comparative study proves that the EMC metric is a valid metric and capable of measuring the reusable quality of XSD.*

*Keywords: XML; XSD; WSDL; Software Metrics; Entropy*

# 1 Introduction

The Web Services Description Language (WSDL) is an XML format for describing the functions of web services and network services and defining interfaces between these services and web based applications. A web service is a software system designed to support interoperable machine-to-machine interaction over a network. Within the web services development environments, developers use WSDL language to facilitate web services without understanding the details of network protocols. Any special data types used are embedded in the WSDL file in the form of XML Schema [1]. In the software development process, when considering a Web service design, XML Schema components should be carefully designed for easy reuse for the purpose of software maintainability, the usage of memory and controlling development cost. The inheritance feature of software has a significant impact on software reusable quality.

In object-oriented programming (OOP), for the XML schemas, inheritance is a way to represent in modules (compartmentalization) and reuse schema components by creating collections of structural schema components [2]. A class, a schema type as a collection of elements and attributes, not only inherits elements or attributes from parent elements, but also validates the contents of these components. This means less programming is required when adding functions to complex web applications. The ability to reuse the existing component collections is a major advantage of object-oriented technology [3]. In the World Wide Web Consortium (w3c) standard schema, using extending or restricting keywords in the simple or complex type definitions can provide inheritance features that elements and attributes inherited from parent elements [4].

Reusable quality is important to reduce software development cost. Many metrics help developers and development groups to assess software quality during the software development process. Although not too much effort has been made to develop XML schema quality metrics, entropy-based metrics have been developed for measuring the maintainability and complexity of XML schema documents. Entropy, in information theory, is used to measure the uncertainty associated with a random variable [5]. In the context of an XML schema document, it is difficult to determine that how many inheritance feature components affect the degree of the reusable quality of the XML schema document; the Entropy method is suitable and useful for measuring the complexity.

By considering all the above issues, the Entropy Measure of Complexity (EMC) was proposed and presented at a conference [6]. One of the authors of the present paper proposed a different metric for reusable and extensible quality [7] for XML Schema Documents. The authors have proposed a formula for estimating target quality of XML schema by utilising the extendible quality (EQ) and reusable quality (RQ).The present work is an extension of the entropy measure of Complexity [6]. This metric is based on entropy concept and measures how components of XML schema documents inherit to other schema components. We have extended the conference work and validated EMC through different perspectives which include empirical validation, practical and theoretical evaluation, and a comparison with a similar metric. A rigorous empirical validation is performed by applying EMC on 70 WSDL real files available on the web. A comparison is also performed by applying the metrics on the same 70 WSDL files considered for empirical validation.

The structure of the paper is as follows. The next section represents the related works and metrics applicable for XML schemas. The definition of the EMC metric is summarised in Section 3. The validation of EMC is performed in Section 4. Finally, the conclusion drawn on the work is in Section 5.

# 2 Related Works

The eXtensible Markup Language (XML) based web applications use XML standard schemas to display information and provide network services. Developing efficient XML web applications requires having good quality XML schema documents. Much research has been done to improve quality in different areas of the software development process and to explore the best practices for knowledge capturing and network services. In addition, many metrics have been proposed to measure the quality of software, but unfortunately, the majority of them are not adopted in industry because of improper empirical validation [8]. Although XML based web applications are important, metrics for XML schema document are scare and there has been very little research to create quality metrics for XML schema documents and thereby improve the web engineering process. Therefore, a mature process can produce high quality schema documents.

McDowell et al. [9] proposed the XML schema analyzer tool to measure two composite indices: the quality and complexity of XML schema documents. This tool was created based on the complexity metrics proposed by Klettke et al. [10]. To ensure the quality of the tool, the ISO 9126 quality model was focused on when developing the tool. Moreover, the tool was an open source tool, to which on could easily add new metrics and change their composite indices according to the requirements of a given application. They concluded that this tool was more important for the XML schema documents than working internal data format for applications. Their future work was the validation of the XML schema analyzer tool.

A schema metric was proposed by Basci and Misra [11] to measure the structure design complexity of the XML schema documents. Their metric was based on the internal structure design components of their schema documents. If their metric value increases, the complexity of the given schema document increases. On the other hand, if the complexity value increases, the quality of the given schema document decreases because of inefficient use of memory and time. They validated their complexity metric theoretically and empirically. To prove the usefulness, they applied well-known structure metrics to XML schema documents and their proposed metric compared with these applied structure metrics.

Basci and Misra [12] have proposed another complexity metric to measure the structural design complexity of the XML schema documents [12]. This metric was developed based on the Shannon entropy function [5], which was suitable for measuring XML schema documents due to having complex structural design of schema components. Their metric provided valuable information for software developers and development groups about the reliability and maintainability of XML schema design. Their proposed metric was analyzed with many examples and empirically validated with test cases [12]. Moreover, to prove the usefulness of their metric, the validation framework and the formal set of nine Weyuker properties were used to evaluate their entropy metric theoretically. The same group of authors has also developed metrics for DTD [13] and Web-services [14]. The authors have proposed to evaluate the structural complexity of the DTD [13] through entropy and estimated the complexity due to repetition of similar structures in schema. A suite of metrics [14] for XML-Web-services maintainability includes five metrics. These metrics evaluate different features of the XML Web-services.

Luo and Shinavier, [15] have proposed a metric to measure schema reuse according to the actor-concept-instance model. Their metric was formulated to calculate the entropy value of simple relationships among actors, concept and instance. In this model, a concept was any one to annotate or describe various data. An actor annotated an instance with a concept. For instance, all user-defined types and build-in types were concepts in a XML schema document and student, teacher and staff types were concepts in the education domain. For example, Rose was an instance of the student type.

The authors [15] used entropy to measure the uncertainty of concepts; the formula would be:

$$H(X) = -\sum_{i}^{n} p(c_i) \log p(c_i) \tag{1}$$

where $p(c_i)$ =Pr(X= $c_i$ ) = $\frac{|A_{c_i}|}{|A|}$ , where $\left|A_{c_i}\right|$ is the total number of annotations using concept $c_i$, $|A|$ is the total number of annotations and i is the total number of concepts. If the metric value was small, the degree of schema reuse was high. This mean that increasing the metric value tends to decrease the degree of schema reuse. Their metric was evaluated against well-known data sets from the well-known web sites. Their research provided knowledge for users about the usefulness of these data sets to create and reuse popular domains.

# 3 Entropy Measure of Complexity (EMC) Metric

To formulate the EMC metric, a directed graph is exploited to demonstrate the inheritance structure of XML schema components. Section 3.1 explains how to demonstrate the components of a given schema document into a directed graph representation. The EMC metric is defined and demonstrated with three schema examples in Section 3.2.

## 3.1 Graph Representation of XML Schema Document

The elements, attributes and types, which are the components of the XML schema, can be inherited from their parent components in the sense of inheritance features that are supported by using restriction or extension keyword implemented within the type definition. The directed graph representation can provide the ability to grasp the complex structure components of the XML schema documents with higher frequencies of occurrences [12]. Before calculating the proposed metric, the inheritance features elements and attributes are counted on the directed graph of a given schema document. For instance, graphs of three schema documents are shown in Figures 1, 2 and 3. In the figures, the root node is a schema element and other circles show either schema elements or attributes. Each Node represents two parts: name and their type with either inheritance features or simple and their name represents in the brackets. The notation of simple type, complex type, simple type with restriction feature, complex type with restriction feature and complex type with extension are ST, CT, STr, CTr and CTe, respectively. Figure 1 contains one element with name CNode and typ: complex type by restriction. It has 4 children: 3 simple type attributes and 1 simple type attribute by restriction.

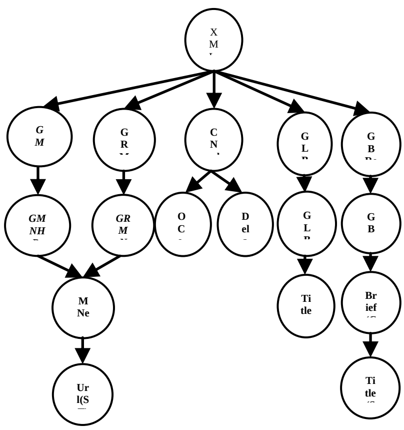


Figure 1
The inheritance feature representation of schema1.xsd

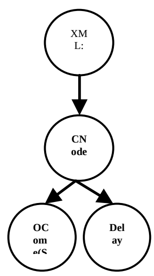


Figure 2
The inheritance feature representation schema2.xsd

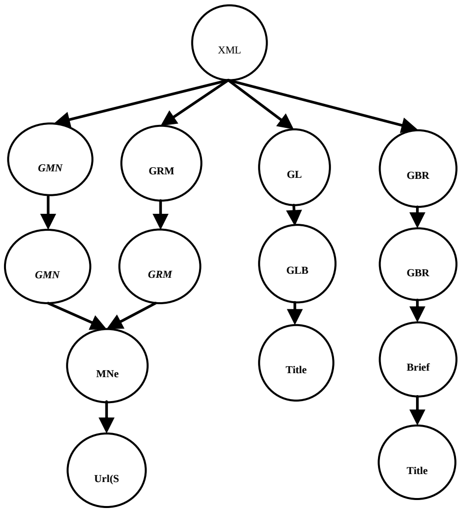


Figure 3
The inheritance feature representation schema3.xsd

## 3.2 The Metric

The Entropy Measure of Complexity (EMC) measures how XML schema components inherit to other schema components. Increasing the EMC values leads to increasing the reusable quality of XML schema documents. On the other hand, greater EMC values means that the given schema documents have many inheritance feature elements and attributes. The EMC metric is based on the entropy function and the used eight inheritance related metrics. The based eight inheritances related metrics are defined and counted over graph representation of a given XML schema document. These based metrics are shown in Table 1. The metric names are given in the first column and their notations are in the second column of Table.

Table 1
The inheritance feature related metrics over the graph representation of XSD

| **Metric Name** | **Notation** |
|---|---|
| Total simple type nodes with restriction | $STNode_R$ |
| Total complex type nodes with restriction | $CTNode_R$ |
| Total complex type nodes with extension | $CTNode_E$ |
| Total simple type nodes without restriction | $STNode_{WR}$ |
| Total complex type nodes without restriction | $CTNode_{WR}$ |
| Total complex type nodes without extension | $CTNode_{WE}$ |
| Total complex type nodes | CTNodes |
| Total simple type nodes | STNodes |

In object-oriented programming (OOP), inheritance is a way of organizing and structuring reuse functions and components. In XML schema components, to get this inheritance feature and properties, derive by restriction and derive by extension are used to inheritance schema component structure. The knowledge of the reusable quality of xml schema helps software developers to save time and money in the XML based software system development process [4]. In Table 1, the first three metrics are inheritance feature nodes that support the reusability of schema document. The next three metrics without inheritance feature do not support reusable quality. The $CT_{Total}$ and $ST_{Total}$ are used to get the ratios of above six metrics for the whole document.

Table 2
The based metrics' values of three XML schema documents

| **Notation** | **Schema1.xml** | **Schema2.xml** | **Schema3.xml** |
|---|---|---|---|
| CTNodes | 12 | 1 | 11 |
| STNodes | 24 | 4 | 20 |
| $STNode_R$ | 1 | 1 | 0 |
| $CTNode_R$ | 1 | 1 | 0 |
| $CTNode_E$ | 4 | 0 | 4 |
| $STNode_{WR}$ | 23 | 3 | 20 |
| $CTNode_{WR}$ | 11 | 0 | 11 |
| $CTNode_{WE}$ | 8 | 1 | 7 |
| EMC | 0.208 | 0.104 | 0.237 |

$STNode_R$, $CTNode_R$ and $CTNode_E$ are inheritance variables, and $STNode_{WR}$, $CTNode_{WR}$ and $CTNode_{WE}$ are non-Inheritance variables. A component contains all elements and attributes. Based on the entropy definition [5], given a schema document (SD), the entropy of a given Schema document has *k* distinct variables ($V_k$) and k is the total number of inheritance type variables. Each variable contains positive and negative concepts.

To measure the EMC metric, each variable is defined as:

$$V_{STNode_R} \leftarrow [STNode_R, STNode_{WR}]$$

$$V_{CTNode_R} \leftarrow [CTNode_R, CTNode_{WR}]$$ and

$$V_{CTNode_E} \leftarrow [CTNode_E, CTNode_{WE}]$$

The entropy metric is formulated based on their relative inheritance probabilities of inheritance variables $P(V_k)$. If the XML document does not contain Inheritance features, its complexity will be computed based on its number of types. As a result, the values are negative. For this purpose, before calculating the entropy equation, an algorithm is used. This algorithm is defined as:

Check if a given schema document (SD) contains inheritance variables.

IF not,

For each inheritance variable:

1. multiply the negative value with minus one
2. replace the positive value of variable with the multiply result
3. decrease the total number of particular type nodes by one
4. replace its negative value with the total number of particular type nodes

Accordingly, the EMC metric is defined as:

$$EMC(SD) = -\sum_{k \in (STNode_R, CTNode_R, CTNode_E)} P(V_k) \log_2 P(V_k) \text{..........}(2)$$

For example, the EMC metric value for the schema document *schema1.xsd* (the listing of schema1.xsd is in Figure 7) is calculated by using Entropy Equation:

$$V_{STNode_R} \leftarrow [1,23]$$

$V_{CTNode_R} \leftarrow [1,11]$ and $V_{CTNode_E} \leftarrow [4,8]$

$$\begin{aligned}
EMC(schema1.xsd) &= -\sum_{k \in (STNode_R, CTNode_R, CTNode_E)} P(V_k) \log_2 P(V_k) \\
&= -P(V_{STNode_R}) \log_2 P(V_{STNode_{WR}}) \\
&\quad - P(V_{CTNode_R}) \log_2 P(V_{CTNode_{WR}}) \\
&\quad - P(V_{CTNode_E}) \log_2 P(V_{CTNode_{WE}}) \\
&= -\frac{1}{24}\log_2 \frac{23}{24} - \frac{1}{12}\log_2 \frac{11}{12} - \frac{4}{12}\log_2 \frac{8}{12} \\
&= 0.208007
\end{aligned}$$

As examples, all inheritance feature related metrics are counted on the graph representation of XML schema documents shown in Table 2. Figure 1, Figure 2 and Figure 3 contain 36, 5 and 31 components, respectively. Figure 3 has the highest ratio of inheritance type variables and the total number of components among them. Therefore, the EMC value also produces the greatest value. A greater EMC metric value means that this XML schema document has many inheritance features, elements and attributes and a high degree of reusable quality.

# 4 Validation of the Proposed Metric

In this section, the usefulness of the proposed metric will be proved by using the validation process. Software developers and development groups should use only validated metrics to assess product and process quality. The EMC metric is validated empirically and evaluated theoretically in Sections 4.1 and 4.2 respectively.

## 4.1 Empirical Validation

Empirical validation is the process of proving the practical usefulness of a new metric. To prove the utility of the EMC metric, 70 schema documents from the well-known WSDL files are analyzed, and the analyzed results of the new metric are shown in Appendix A. Figure 4 shows the numbers of nodes with simple types by restriction and with complex types by extension for each schema file. These files have not nodes with complex types by restriction. The comparative results between EMC and H metric values for analyzed schema documents with inheritance features are shown in Figure 5.

The EMC metric can better differentiate the schema files in terms of the inheritance type nodes relationships. Moreover, the two metric values are the ratio to total type nodes. The H value defines and measures the information entropy of actor-concept-instance relationships in a given schema document. According to this Figure, the higher the reuse quality, the higher the EMC values. Inheritance type nodes that contain all elements and attributes are directly related EMC values. The highest EMC value contains more inheritance simple and complex type nodes than others. It is clear that the schema reusable and quality will increase since it has more inheritance feature types of attributes and elements.

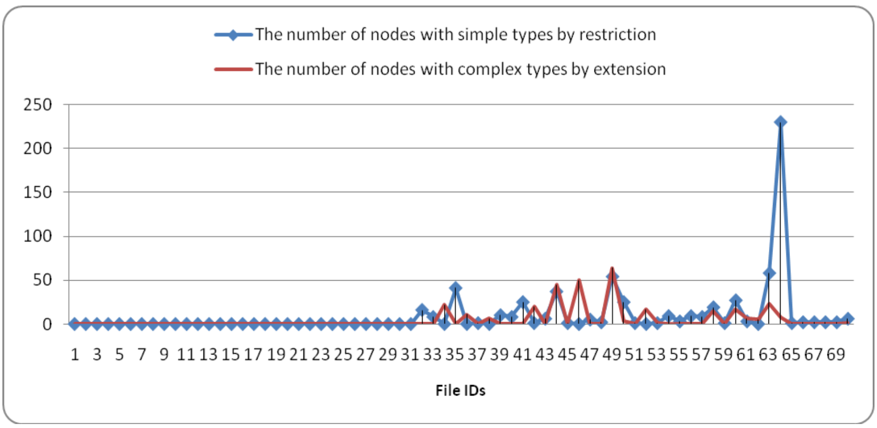


Figure 4
The number of nodes with simple types by restriction and the number of nodes with complex type by extension for each schema file

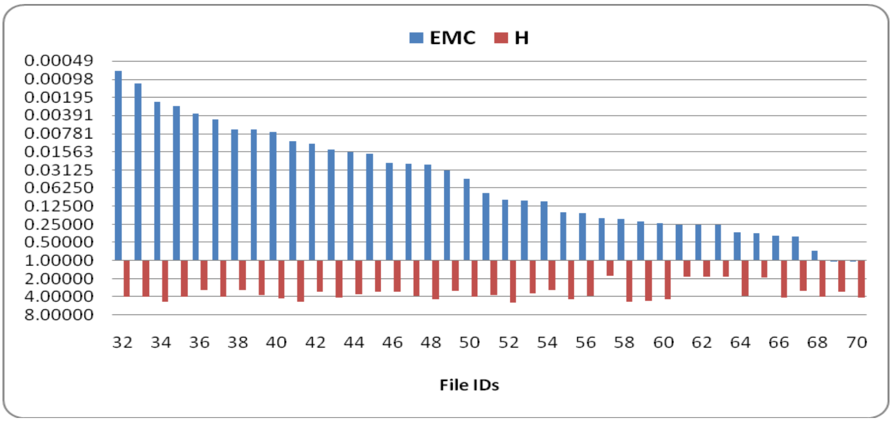


Figure 5
The comparative results between EMC and H values for analyzed WSDL files with inheritance features

Among these 70 schema files, the H metric measures and estimates the schema ID 1 as the most reuse quality at 1.25163 (in Figure 6) and arranges the file IDs 1, 2, 13, 22, 6, 57, 61, 62, 63, 65, 23 and so on in terms of the degree of schema reuse. The file IDs 1, 2, 13, 22, 6, 23 and others greater than ID 31 have not contained any inheritance feature type nodes. Therefore, the H metric does not consider inheritance feature type nodes. The EMC metric measures the 70 schema files and estimates the file ID 70 as having the highest reuse quality. Figure 6 illustrates the comparative results between EMC and H values for analyzed WSDL files without inheritance features. The EMC metric can calculate these files computed based on their simple and complex type nodes in the schema documents.

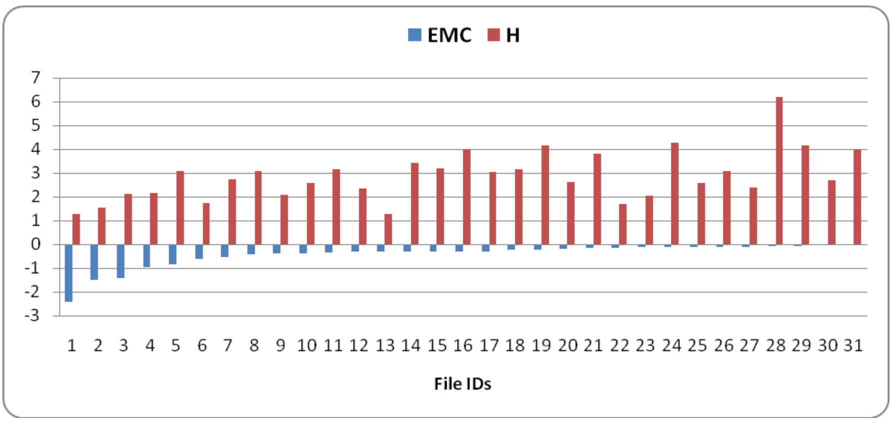


Figure 6
The comparative results between EMC and H values for analyzed WSDL files without inheritance features

## 4.2 Theoretical Validation of EMC Metric

The usefulness and quality of a new metric is also evaluated by using theoretical validation. In order to perform the validation of the presented metric, the section is organized as follows. EMC is evaluated against Kaner's evaluation framework [16]. Moreover, in section 4.2.2, EMC is also evaluated against the well-known Weyuker's properties [17] through a case study.

### 4.2.1 Evaluation through a Practical Framework

The practical success of the proposed metric is very important. The metric should be examined formally and practically for its proper validation. When we analyzed the EMC metric according to the practical framework given in [16], EMC is identified as an indirect metric because it depends on many attributes. The EMC is a measure of software reusability and flexibility based on the complexity of schema documents. In the following paragraphs, EMC is evaluated by Kaner's framework.

**The purpose of the measure:** The purpose of the EMC metric is to help software developers undertake private assessment and to improve their schema based software products.

**Scope of usage of the measure:** The proposed EMC metric is a reusable quality-measuring tool for software developers and development groups working especially on the XML based applications.

**Identified Attribute to measure:** The identified attribute measured by EMC is the reusable and flexible quality of the XML schema. A higher complexity value of the schema makes it more reusable and flexible.

**Natural scale of the attribute:** The natural scale of the attribute is difficult to identify because quality has several definitions, and the reusable quality of XML schema can be measured by several methods.

**Definition of metric:** The definition of the EMC is given in Section 3.

**Measuring instrument to perform the measurement:** For inheritance feature metrics of a schema document, the developed oriented model (DOM) parser is used to parse components of this document, and then the system counts these particular components for the particular metric.

**Natural scale for the metric:** The EMC does not satisfy the additive property so it is not on ratio scale. The exact scale of metric is a task of future work.

**Relationship between the attribute to the metric value:** The EMC is intend to measure the reusable quality of XML schema, and therefore the metric is directly related to the quality attribute. The experimentations show that an increase in EMC reflects that the schema reusable and flexible quality will increase since it implies having more inheritance feature types of attributes and elements. EMC metric is not a unique indicator of schema reusable and flexible quality.

### 4.2.2 Evaluation through Weyuker's Properties

In this section, an evaluation of the EMC is also done against Weyuker's properties [17]. Several object oriented metrics are suitable only for the six Weyuker's properties, and other properties are not very useful [18] [19]. The EMC metric is evaluated against 9 properties by using a case study. The evaluations of EMC against the Weyuker's properties are as follows.

***Property 1:*** **($\exists$ P) ($\exists$ Q) ($|P| \neq |Q|$) where P and Q are the two different XML schema documents.**

There are different EMC values of all 70 schemas because these different schema documents have different inheritance feature arguments. Hence, the EMC metric satisfies this property.

***Property 2:*** **Let c be a non-negative number, and then there are only finite numbers of schema documents of complexity c.**

All schema documents consist of only a finite number of inheritance feature based metrics and the EMC metric highly depends on these based metrics. This means that there are only a finite number of XML schema documents of the same complexity if the complexity is a non-negative number. Therefore, EMC satisfies this property.

***Property 3:*** **There are distinct classes P and Q such that $|P| = |Q|$.**

This property states that there exist many schema documents of the same complexity value. One can find the same EMC values, if different schema documents have the same inheritance feature arguments. Thus, the EMC metric satisfies this property.

***Property 4:*** **($\exists$ P) ($\exists$ Q) ($P \equiv Q \,\&\, |P| \neq |Q|$)**

If P and Q are different schema documents having the same functionality, their EMC values can be different because of different implementation. As the EMC metric is based on the internal structure of schema documents, it satisfies this property.

***Property 5:*** **($\forall$ P) ($\forall$ Q) ($|P| \leq |P;Q| \,\&\, |Q| \leq |P;Q|$)**

This property states that if the combined schema is constructed from schema P and schema Q, the value of the combined schema document is larger than the value of schema P or schema Q. In Table 3, although Figure 1 is the combination of Figure 2 and Figure 3, the value of Figure 3 is larger than those of the Figure 1. According to this result, the proposed metric cannot satisfy this property.

***Property 6:*** $(\exists P)(\exists Q)(\exists R)(|P| = |Q|) \,\&\, (|P;R| \neq |Q;R|)$

This property states that if a new schema document is appended to two schema documents of the same EMC value, the values of the appended documents can be different. For instance, we have two schema documents P, Q and R. These schemas have inheritance feature documents:

P:{ $V_{STNode_R} \leftarrow [2,4]$,

$V_{CTNode_R} \leftarrow [2,14]$,

$V_{CTNode_E} \leftarrow [2,14]$},

Q:{ $V_{STNode_R} \leftarrow [3,6]$,

$V_{CTNode_R} \leftarrow [1,7]$,

$V_{CTNode_E} \leftarrow [1,7]$} and

R:{ $V_{STNode_R} \leftarrow [1,13]$,

$V_{CTNode_R} \leftarrow [1,9]$,

$V_{CTNode_E} \leftarrow [2,7]$}.

P and Q have the same EMC values of 0.243149 and R produces the value of 0.125752. The R schema is then appended to the P and Q schema documents.

(P;Q) :{ $V_{STNode_R} \leftarrow [3,17]$,

$V_{CTNode_R} \leftarrow [3,23]$,

$V_{CTNode_E} \leftarrow [4,22]$} and

(Q;R):{ $V_{STNode_R} \leftarrow [4,19]$,

$V_{CTNode_R} \leftarrow [2,16]$,

$V_{CTNode_E} \leftarrow [3,15]$}.

We can observe that the values of the appended P and Q schema documents are different with 0.92657 and 0.110656, respectively. Therefore, this property is also satisfied by the proposed metric.

***Property 7:*** **There are two schema P and Q such that Q is formed by permuting the structure components of P, and** $|P| \neq |Q|$.

The presented metric highly depends on the internal inheritance structure of schema documents and so the EMC metric satisfies this property.

***Property 8:*** **If P is renaming of Q, then** $|P| = |Q|$.

The proposed metric satisfies this property because the value of a given schema is not changed even if the names of the schema and inheritance feature components in this schema are changed.

***Property 9:*** $(\exists P)(\exists Q)(|P| + |Q|) < (|P;Q|)$

According to Table 2, Example1 is the combination schema of P (schema2.xsd) and Q (schema3.xsd). The EMC values of P and Q are 0.104 and 0.207, respectively. The value of the combination schema document is 0.204, and this value is less than the sum of the P and Q values. Therefore, EMC does not satisfy property 9. Further, if two schema are combined, then the complexity of the combined schema will either be less than the sum of the individual ones (due to fact if some modules/elements are in common) or equal (if all modules/elements are different), but in no case will the complexity of the combined schema be less than complexity of the individuals.

In this section, the proposed metric is validated against 9 Weyuker's properties. The EMC metric satisfies 7 properties. It is important to note that it is not necessary to satisfy all the Weyuker's properties [18]. From this point of view, EMC's satisfying seven Weyuker's properties shows that it is a robust measure.

We have followed almost all the steps suggested for the evaluation and validation of software complexity measures [20], except that we have adopted Weyuker's properties in the place of principles of measurement theory for theoretical validation. According to the measurement theory criteria for software complexity measures, a metric should be on ratio scale but is not applicable in majority of object oriented metrics [21]. Our metric is also found not on ratio scale. It is proved via Weyuker's property 9 that EMC is not an additive measure.

## Conclusion

The reusable nature of XML schema documents allows developers and software development groups to have the capability of increasing productivity and decreasing development cost of the XML based applications. Increased flexibility and reusability in XML schema documents results in an increased number of inheritance feature elements and attributes in these documents. The EMC metric is developed to achieve these goals. The EMC metric is based on the entropy concept and inheritance feature elements and attributes of XML schema documents. The EMC metric is passed through a rigorous validation process.

EMC is practically evaluated against Kaner's framework. Theoretical evaluation has been done against nine Weyuker's properties. EMC satisfies seven of Weyuker's properties. The practical evaluation and theoretical validation of EMC proves that the metric is developed on scientific principles. The empirical validation is done by applying the metric on 70 real WSDL files. The results and a comparison with the H metric proved the worth and usefulness of the metric. It is found that measuring the reusable quality of XML schema document with the EMC metric will be more useful than via other related metrics. As future work, we aim to explore other factors that are responsible for increasing the complexity of XML schemas. Fixing the threshold values for the EMC metric is also a task of future work.

## References


[1] http://www.w3.org/TR/wsdl (Last retrieved on June 2011)

[2] Bieman, J. M., Kang B. K. (1995) Cohesion and Reuse in an Object-Oriented System, In Proceedings of the 1995 Symposium on Software reusability, pp. 259-262

[3] http://java.boot.by/scea5-guide/ch01.html (Last retrieved on June 2011)

[4] http://www.w3c.org/XML/Schema (Last retrieved on June 2011)

[5] Cover T. M., Thomas J. A., Elements of Information Theory, Second Edition (2006)

[6] Thaw T. Z. Khin M. M. (2011) Entropy Measure of XML Schema Document Complexity Metric, Proc. International Conference on Computer Research and Development - ICCRD, pp. 476-479

[7] Thaw T. Z. (2011) Measuring and Evaluation of Reusable Quality and Extensible Quality for XML Schema Documents, In Proc. IEEE Student Conference on Research and Development, pp. 473-478

[8] Misra S. (2011) An Approach for the Empirical Validation of Software Complexity Measures, Acta Polytechnica Hungarica, 8(2), pp. 141-160

[9] McDowell A., Schmidt C., Bun K. Y. (2004) Analysis and Metrics of XML Schema, In Proceedings of Intl Conference on Software Engineering Research and Practice, pp. 538-544

[10] Klettke M., Schneider L., Heuer A. (2002) Metrics for XML Document Collections, In Proc. of XMLDM Workshop, Czech Republic, pp. 162-176

[11] Basci D., Misra S. (2009) Measuring and Evaluating a Design Complexity Metric for XML Schema Documents' Journal of Information Science and Engineering, 25(5) pp.1405-1425

[12] Basci D., Misra S. (2010) Entropy as a Measure of Quality of XML Schema Document, International Arab Journal of Information Technology, 8(1) pp. 16-24

[13] Basci D., Misra S. (2011) Document Type Definition (DTD) Metrics, Romanian Journal of Information Science and Technology, 14(1) pp. 31-50

[14] Basci D., Misra S. (2011) Metrics Suite for Maintainability of XML Web-Services, IET Softwar, 5(3), pp. 320-341

[15] Luo X., Shinavier J. (2009) Entropy-based Metrics for Evaluating Schema Reuse, In Proceedings of the 4th Asian Conference on The Semantic Web (Springer-Verlag Berlin Heidelberg) pp. 321-331

[16] Kaner C. (2004) Software Engineering Metrics: What Do They Measure and How Do We Know? In Proceedings of the 10th International Software Metrics Symposium, 2004, pp. 1-10

[17] Weyuker E. J. (1988) Evaluation Software Complexity Measures, IEEE Transactions on Software Engineering, 14, pp. 1357-1365

[18] Misra S. (2011) Evaluation Criteria for Object-oriented Metric, Acta Polytechnica Hungarica, 8(5) pp. 109-136

[19] Misra S., Akman I. (2008) Applicability of Weyuker's Properties on OO Metrics: Some Misunderstandings, Journal of Computer and Information Sciences, 5(1) pp. 17-24

[20] Misra S., Akman I., Palacios R. C. (2012) A Framework for Evaluation and Validation of Software Complexity Measures, IET Software, 6(4), pp. 323-334

[21] Zuse, H (1996) Foundations of Object-oriented Software Measures. In Proceedings of the 3rd International Symposium on Software Metrics: From Measurement to Empirical Results (METRICS '96) IEEE Computer Society, Washington, DC, USA, pp. 75-84

## APPENDIX A

Table 3

The results of EMC metrics for the analyzed schema files. Files are arranged according to the EMC values with ascending.

| ID | WEB LINK | EMC | H |
|---|---|---|---|
| 1 | http://www.webservicex.net/CreditCard.asmx?WSDL | -2.41504 | 1.25163 |
| 2 | http://www.thomas-bayer.com/axis2/services/BLZService?wsdl | -1.49185 | 1.25163 |
| 3 | http://www.elguille.info/NET/WebServices/HolaMundoWebS.asmx?WSDL | -1.41504 | 2.12809 |
| 4 | http://service.ecocoma.com/geo/cityzip.asmx?WSDL | -0.96578 | 2.16096 |
| 5 | http://www.yazgelistir.com/YGServices/ArticleService.asmx?wsdl | -0.85982 | 3.09580 |
| 6 | http://service.ecocoma.com/geo/distance.asmx?WSDL | -0.63298 | 1.73136 |
| 7 | http://www.webservicex.net/*BibleWebservice.asmx*?wsdl | -0.52279 | 2.73451 |
| 8 | http://rangiroa.essi.fr:8080/dotnet/evaluation-cours/EvaluationWS.asmx?WSDL | -0.42954 | 3.06365 |
| 9 | http://services.nirvanix.com/ws/Authentication.asmx?WSDL | -0.38201 | 2.07486 |

| | | | |
|---|---|---|---|
| 10 | http://services.argosoft.com/AddressValidation/AddressVerifier.asmx?WSDL | -0.38201 | 2.56131 |
| 11 | http://service.ecocoma.com/convert/chinese.asmx?WSDL | -0.34417 | 3.16608 |
| 12 | http://www.geoservicios.com/V2.0/sgeo/sgeo.asmx?WSDL | -0.31348 | 2.35019 |
| 13 | http://service.ecocoma.com/shipping/fedex.asmx?WSDL | -0.31063 | 1.26903 |
| 14 | http://services.test.musiccue.net/rapidcueapplication/WorkManager.asmx?WSDL | -0.30694 | 3.41259 |
| 15 | http://ws.cdyne.com/emailverify/Emailvernotestemail.asmx?wsdl | -0.30134 | 3.20987 |
| 16 | http://www.mathertel.de/AJAXEngine/S02_AJAXCoreSamples/CalcService.asmx?WSDL | -0.29183 | 3.99255 |
| 17 | http://quisque.com/fr/chasses/blasons/search.asmx?WSDL | -0.29170 | 3.05216 |
| 18 | http://quiksilver.ws.eto.fr/Connexion.asmx?WSDL | -0.22151 | 3.14242 |
| 19 | http://webservice.webxml.com.cn/webservices/ChinaTVprogramWebService.asmx?WSDL | -0.21270 | 4.16272 |
| 20 | http://demo.soapam.com/services/FedEpayDirectory/FedEpayDirectoryService?WSDL | -0.17715 | 2.60746 |
| 21 | http://www.oorsprong.org/websamples.arendsoog/ArendsoogbooksService.wso?WSDL | -0.16379 | 3.82050 |
| 22 | http://trial.serviceobjects.com/pa/phoneappend.asmx?WSDL | -0.15110 | 1.70601 |
| 23 | http://ws2.serviceobjects.net/ev/EmailValidate.asmx?WSDL | -0.13412 | 2.03487 |
| 24 | http://api.legiomedia.com/Content.asmx?WSDL | -0.13019 | 4.28320 |
| 25 | http://secure.adpay.com/affiliate/affiliates.asmx?wsdl | -0.12088 | 2.57051 |
| 26 | http://hooch.cis.gsu.edu/bgates/MathStuff/Mathservice.asmx?WSDL | -0.11651 | 3.08831 |
| 27 | http://www.sipeaa.it/wset/ServiceET.asmx?WSDL | -0.11507 | 2.37127 |
| 28 | http://developer.factiva.com/2.0/wsdl/FDKParsers.wsdl | -0.07789 | 6.19208 |
| 29 | http://www.banguat.gob.gt/variables/ws/BDEF.asmx?WSDL | -0.06640 | 4.13503 |
| 30 | http://omnovastage.crowechizekasp.com/attributes.asmx?wsdl | -0.04449 | 2.68170 |
| 31 | http://ws.eoddata.com/data.asmx?wsdl | -0.00333 | 4.00160 |
| 32 | http://www.chemspider.com/MassSpecAPI.asmx?WSDL | 0.00072 | 3.96796 |
| 33 | http://ws.interfax.net/dfs.asmx?WSDL | 0.00117 | 3.95824 |
| 34 | http://demo.turtletech.com/latest/webAPI/metering.asmx?WSDL | 0.00234 | 4.77941 |
| 35 | http://ssl.9squared.com/catalog/catalog.asmx?WSDL | 0.00275 | 3.96266 |
| 36 | http://www.imagine-r.com/services/WsImagineR.asmx?WSDL | 0.00370 | 3.11048 |
| 37 | https://api.wildwestdomains.com/wswwdapi/wapi.asmx?wsdl | 0.00458 | 3.95678 |
| 38 | https://demo.docusign.net/API/3.0/Credential.asmx?WSDL | 0.00664 | 3.08703 |
| 39 | http://service.thefamousgroup.com/ProjectService.asmx?wsdl | 0.00671 | 3.71067 |
| 40 | http://www.esendex.com/secure/messenger/soap/ContactService.asmx?WSDL | 0.00742 | 4.28139 |
| 41 | http://msrmaps.com/TerraService2.asmx?WSDL | 0.01034 | 4.78186 |
| 42 | http://www.multispeak.org/interface/30j/10_OA_EA.asmx?WSDL | 0.01141 | 3.32276 |
| 43 | http://terraserver-usa.com/LandmarkService.asmx?WSDL | 0.01446 | 4.11318 |
| 44 | http://svc.exaphoto.com/eXaPhoto/CollectionServices.asmx?WSDL | 0.01601 | 3.59425 |
| 45 | http://services.nirvanix.com/ws/Accounting.asmx?WSDL | 0.01669 | 3.28918 |
| 46 | http://www.phdcc.com/findinsite/SearchService.asmx?wsdl | 0.02408 | 3.22759 |
| 47 | http://www.partenairedejeu.fr/WebServices/RelationManager.asmx?WSDL | 0.02488 | 3.89019 |
| 48 | https://www.devcallnow.com/WebService/OneCallNow.asmx?wsdl | 0.02583 | 4.32031 |
| 49 | https://api.channeladvisor.com/ChannelAdvisorAPI/v5/AdminService.asmx?WSDL | 0.03177 | 3.17985 |
| 50 | http://114-svc.elong.com/NorthBoundService/V1.1/ NorthBoundAPIService.asmx?WSDL | 0.04398 | 3.91752 |
| 51 | http://www.hitslink.com/reportws.asmx?WSDL | 0.07633 | 3.71378 |

| | | | |
|---|---|---|---|
| 52 | https://api.channeladvisor.com/ChannelAdvisorAPI/v3/OrderService.asmx?WSDL | 0.09683 | 5.00673 |
| 53 | http://b3.caspio.com/ws/api.asmx?wsdl | 0.09958 | 3.47567 |
| 54 | http://gw1.aql.com/soap/sendsmsservice.php?wsdl | 0.10376 | 3.02392 |
| 55 | http://labs.bandwidth.com/api/public/voip/v1_1/NumberManagementService.asmx?wsdl | 0.15874 | 4.37224 |
| 56 | http://www.xignite.com/xNews.asmx?WSDL | 0.16501 | 3.89726 |
| 57 | http://www.webservicex.net/TranslateService.asmx?wsdl | 0.19499 | 1.79248 |
| 58 | http://www.xignite.com/xCalendar.asmx?WSDL | 0.20097 | 4.83966 |
| 59 | http://water.sdsc.edu/wateroneflow/EPA/cuahsi_1_0.asmx?WSDL | 0.22382 | 4.71068 |
| 60 | http://ws.strikeiron.com/MidnightTraderFinancialNews?WSDL | 0.24208 | 4.31742 |
| 61 | http://www.webservicex.net/ConvertTemperature.asmx?WSDL | 0.25428 | 1.84237 |
| 62 | http://www.webservicex.net/ConverPower.asmx?WSDL | 0.25428 | 1.84237 |
| 63 | http://www.webservicex.net/CovertPressure.asmx?WSDL | 0.25428 | 1.84237 |
| 64 | http://www.xignite.com/xwatchlists.asmx?WSDL | 0.33584 | 3.78410 |
| 65 | http://www.webservicex.com/CurrencyConvertor.asmx?wsdl | 0.34601 | 1.91830 |
| 66 | http://www.xignite.com/xQuotes.asmx?WSDL | 0.38205 | 4.14231 |
| 67 | http://www.xignite.com/xDataSet.asmx?wsdl | 0.39984 | 3.13490 |
| 68 | http://event.peoplenet.dk/_vti_bin/dspsts.asmx?wsdl | 0.69499 | 4.00041 |
| 69 | http://www.xignite.com/xNASDAQLastSale.asmx?WSDL | 1.00640 | 3.30242 |
| 70 | http://www.xignite.com/xMetals.asmx?WSDL | 1.02051 | 4.13493 |

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<s:schema elementFormDefault="qualified"  targetNamespace="http://www.xignite.com/services/">
<s:element name="GetBriefings">
   <s:complexType />
</s:element>
<s:element name="GBResp">
   <s:complexType>
      <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="GBResu"
              type="tns:ArrayOfBriefing" />
      </s:sequence>
   </s:complexType>
</s:element>
<s:complexType name="ArrayOfBriefing">
    <s:sequence>
      <s:element minOccurs="0" maxOccurs="unbounded" name="Brief" nillable="true"
          type="tns:Briefing" />
    </s:sequence>
</s:complexType>
<s:complexType name="Briefing">
   <s:complexContent mixed="false">
   <s:extension base="tns:Common">
```

```
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="Title" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Time" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Text" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="Html" type="s:string" />
      </s:sequence>
  </s:extension>
  </s:complexContent>
</s:complexType>
<s:element name="CNode">
<s:complexType name="Common">
<s:restriction base="xsd:anyType">
    <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="OCome" type="tns:OutcomeTypes" />
   <s:element minOccurs="0" maxOccurs="1" name="Message" type="s:string" />
   <s:element minOccurs="0" maxOccurs="1" name="Identity" type="s:string" />
   <s:element minOccurs="1" maxOccurs="1" name="Delay" type="s:double" />
  </s:sequence>
</s:restriction>
  </s:complexType>
</s:element>
<s:simpleType name="OutcomeTypes">
  <s:restriction base="s:string">
   <s:enumeration value="Success" />
   <s:enumeration value="SystemError" />
   <s:enumeration value="RequestError" />
   <s:enumeration value="RegistrationError" />
     </s:restriction>
  </s:simpleType>
<s:element name="GLBR">
   <s:complexType>
       <s:sequence>
           <s:element minOccurs="0" maxOccurs="1" name="GLBRe"  type="tns:Briefing" />
       </s:sequence>
   </s:complexType>
 </s:element>
<s:element name="GMNHRp">
   <s:complexType>
      <s:sequence>
         <s:element minOccurs="0" maxOccurs="1" name="GMNHRs"
             type="tns:ArrayOfMarketNews" />
      </s:sequence>
   </s:complexType>
 </s:element>
```

```
<s:complexType name="ArrayOfMarketNews">
    <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="MNews"  nillable="true"
type="tns:MarketNews" />
    </s:sequence>
</s:complexType>
<s:complexType name="MarketNews">
   <s:complexContent mixed="false">
   <s:extension base="tns:Common">
      <s:sequence>
         <s:element minOccurs="0" maxOccurs="1" name="Headline" type="s:string" />
         <s:element minOccurs="0" maxOccurs="1" name="Time" type="s:string" />
         <s:element minOccurs="0" maxOccurs="1" name="Source" type="s:string" />
         <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="OriginalUrl" type="s:string" />
            <s:element minOccurs="0" maxOccurs="1" name="Summary" type="s:string" />
       </s:sequence> </s:extension> </s:complexContent>
 </s:complexType>
<s:element name="GRMNHRp">
   <s:complexType> <s:sequence>
      <s:element minOccurs="0" maxOccurs="1"  name="GRMNHRs" type="tns:ArrayOfMarketNews" />
      </s:sequence> </s:complexType>
</s:element>
</s:schema>
```

Figure 7

The list of the schema documents schema1.xsd