Measuring Complexity of Domain Standard Specifications using XML Schema Entropy

George Feuerlicht[1,2,3], Vladimir Kovar[2], David Hartman[2], Marek Beranek[2], and Pavel Bory[2]

[1]University of Technology, Sydney, P.O. Box 123, Broadway, NSW 2007, Australia
george.feuerlicht@gmail.com
[2]Unicorn College, V Kapslovně 2767/2, 130 00 Prague 3, Czech Republic
david.hartman,pavel.bory,vladimir.kovar,marek.beranek}@unicorncollege.cz
[3]Prague University of Economics, W. Churchill Sqr. 4,
130 67 Prague 3, Czech Republic

Abstract. XML schemas are used extensively in e-commerce standardization initiatives. Such XML-based standards define the structure and the semantics of messages that are used to implement business transactions in a particular industry domain (e.g. travel). The design of the document structures that form the message payloads is of key importance as once the specification is published it is difficult to re-design the documents without impacting on existing applications. Furthermore, such domain standards need to be maintained and evolved over long time periods, typically decades, without unduly increasing the complexity of the specification. The concept of software entropy has been used in the literature to estimate complexity and to express decline in quality, maintainability and understandability of software though its lifetime. In this paper we propose a Message Software Entropy (MSE) metric that estimates the complexity of XML message structures and we use this metric to study the complexity of a subset of the Open Travel Alliance Specification as it evolves over time.

Keywords: complexity metrics, XML schema evolution, software entropy

1 Introduction

XML schemas specifications are used extensively in both vertical domain (e.g. Open Travel Alliance [1]) and horizontal domain (e.g. ebXML [2]) e-commerce standardization initiatives. Such XML standards define the structure and the semantics of messages that are used to implement business transactions in a specific domain of interest (e.g. travel). The messages are typically delivered using SOAP [3] or REST [4] web services, so that in effect the specification forms a basis for a large and complex SOA (Service Oriented Architecture) software system. The design of the document structures that form the message payloads is of key importance as once the specification is published it is difficult to re-design without impacting on existing applications. Furthermore, the longevity

of domain standards requires that the design of the standard documents allows *graceful* evolution over a very long time period, typically decades, without unduly increasing the complexity of the specification. Many existing standard specifications were designed to overcome the limitations of the Internet as it was at the end of the last century (i.e. high latency, poor reliability and slow response time) rather than with focus on software maintainability and evolution. This typically resulted in specifications consisting of large complex messages suited for stateless communication, but difficult to maintain and evolve. Standard XML specifications (e.g. OTA) typically contain hundreds of complex XML message schemas and thousands of schema elements. As these specifications evolve over time incorporating new requirements their complexity further increases. For example, the OTA message schema that defines the structure of the flight availability requests (OTA_AirAvailRQ) contains 428 elements with multiple levels of nesting. Design of such XML schemas typically follows the Document Engineering approach [5] or a similar methodology that produces XML documents by aggregating data elements based on pre-defined simple and complex types. For example, OTA message level schemas are constructed by aggregation of simple (OpenTravel Simple Types) and complex (OpenTravel Common Types, and Industry Common Types) schema elements [1]. This design approach while ensuring uniform structure and semantics of data elements can result in overlapping message schemas and excessive complexity, reducing the maintainability of the specification [6].

Evaluation of the quality of the design of XML schema has been the subject of recent research interest. In our previous work we have proposed data coupling metrics that evaluate interdependencies among of a set of XML message schemas by estimating the level of data coupling [7]. In this paper we focus on estimating the complexity of message schemas using the concept of Schema Entropy [8]. In the following section (Section 2) we review related work that addresses the problem of design quality of XML schemas. We then describe our proposal for the Message Schema Entropy (MSE) metric and an XSD Analyzer tool that we developed to compute the MSE metric (Section 3). In Section 4 we present experimental results obtained by analysing OTA Air (Airline) schemas. We note that the OTA message schemas were chosen as an example of open industry domain specifications, and that we do not imply any criticism of the OTA schema design in this work. Section 5 presents our conclusions and outlines directions for further work.

2 Related Work

Evaluation of the quality of design of XML schemas has been the subject of recent research interest [9-11]. Ensuring XML schema design quality for industry domain specifications presents a particularly difficult problem as the schemas are often developed in the absence of a domain data model [9]. Current work in this area includes research that focuses on identifying dependencies among schema elements and developing tools for automating the propagation of schema changes

to all dependent components. Necasky, et al. proposed a five-level XML evolution architecture with the top level Platform-Independent Model (PIM) that represents the data requirements for a particular domain of interest. PIM model is mapped into a Platform-Specific Model (PSM) that describes how parts of the PIM schema are represented in XML. PSM then maps into Schema, Operational and Extensional level models. Atomic operations (create, update, and remove) for editing schemas are defined on classes, attributes, and associations, and a mechanism for propagating these operations from PIM to PSM schema is proposed. Composite operations are constructed from atomic operations to implement complex schema changes [12-14]. Numerous XML schema quality metrics have been proposed primarily with the objective to measure various aspects of schema complexity. McDowell et al. proposed eleven metrics and two composite indexes to measure the quality and complexity of XML schemas. These metrics are based on counts of complex type declarations, derived complex types, number of global type declarations, number of simple types, element fanning (*fan-out* – number of child elements that an element has, and *fan-in* – number of times that an element is referenced by), etc. [9]. The authors formulate a *Quality Index* and a *Complexity Index* that estimate the quality and complexity of XML schemas based on a weighted values of the metrics. A metric analysis tool is provided for developers to verify the validity of the metrics in the context of specific projects. The concept of entropy [15] has been adapted to the measurement of complexity of software and was initially applied to procedural software [16] and later to object-oriented design [17,18]. Ruellan [19] used an entropy measure to assess the amount of information contained in XML documents (information density) with the objective to reduce the size of XML documents and to improve processing speed of XML messaging applications. Thaw et al. .[20] proposed entropy-based metrics to measure reusability, extensibility, understandability of XML schema documents. Basci et al. [11] proposed and validated XML schema complexity metric that evaluates the internal structure of XML documents taking into account various sources of complexity that include recursion and complexity arising from importing external schema elements. The authors used the concept of Schema Entropy (SE) to assess XML schema complexity. SE is evaluated based on the complexity of schema elements as measured by fan-in and fan-out, and the number of simple elements that constitute individual schema elements. The SE metric was empirically validated using publicly available XML schemas, and the authors conclude that the metric provides a useful feedback when comparing schemas with equal number of elements [8]. In [21] Tang et al. apply an entropy-based measure to assessing the structural uniformity (*structuredness)* of XML documents. Two metrics are defined: Path-Based Entropy and Subtree-Based Entropy that attempt to measure the level of diversity of a set of XML documents. Unlike Basci et al. [8,11], the authors base the entropy calculation on XML documents, rather than XML schemas. Pichler et al. [6] developed a set of metrics to analyse the complexity of business documents with the objective of estimating the effort involved in data element mapping between different business document standards.

Our proposal differs from the above approaches in two important respects. Firstly, we estimate schema entropy by adapting an entropy-based metric originally developed for object-oriented design [17], and secondly our focus is on the evaluation of the changes in complexity of domain specifications as the specifications evolves over time.

3 Proposed Message Schema Entropy (MSE) Metric

The concept of software entropy has been used in literature to express decline in quality, maintainability and understandability of software though its lifetime. In our formulation of the Message Schema Entropy (MSE) metric we adapt the Class Definition Entropy (CDE) metric for object-oriented design described in Bansiya et al. [17]. The CDE metric calculates the frequency of occurrence of name strings in a given class. Calculation of MSE metric involves computing the frequency of occurrence of complex schema elements (i.e. schema elements that contains other elements and attributes) using the formulae:

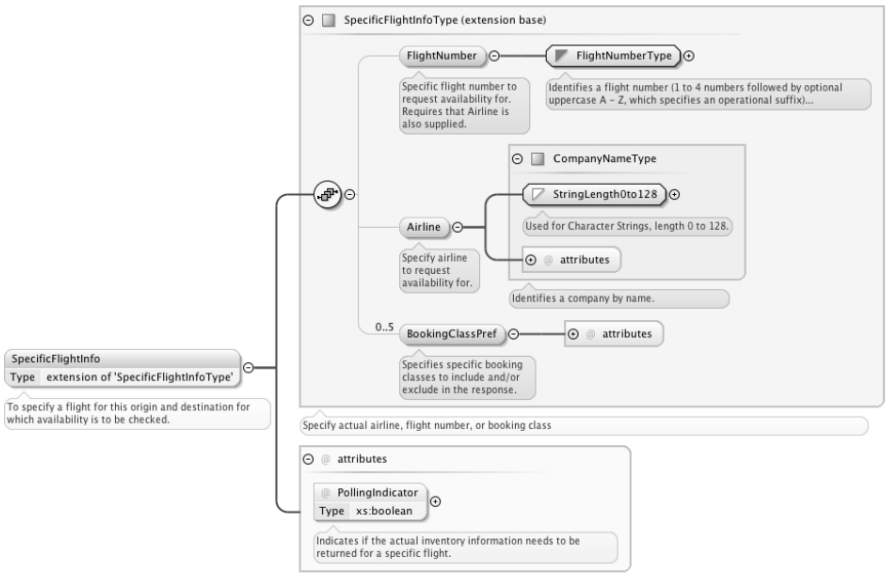

Fig. 1. Fragment of OTA AirAvailRQ schema

$$\text{MSE} = -\sum_{j=1}^{N} (P_i \log_2 P_i)$$

where:

N = total number of unique complex elements in the message schema

$\mathbf{n_i}$ = number of occurrences of the i^{th} complex element in the message schema

M = total number of non-unique complex elements in the message schema
$P_i = n_i/M$

Figure 1 shows a fragment of the schema of the OTA_AirAvailRQ message illustrating the SpecificFlightInfo element. The OTA_AirAvailRQ message is used to implement (web) service for flight availability inquiry and includes 428 elements with multiple levels of nesting. Elements are based on simple types (e.g. FlightNumber) or on complex types, as is the case with the SpecificFlightInfo element, which is based on the SpecificFlightInfo Type (extension of SpecificFlightInfo Type). MSE calculation is based on counting complex schema elements (i.e. elements based on complex types) and represents an approximation, as the internal complexity of individual elements is not taken into account.

3.1 XSD Analyzer Tool

We have developed an XSD Analyzer tool that calculates the values of the MSE. XSD Analyzer allows the selection of message schemas for analysis and produces an output that includes the total number of non-unique complex schema elements (N1), the number of unique (distinct) complex schema elements (N2), the value of MSE, and counts of occurrences of complex schema elements.

4 Experimental Results

The OTA Air messages are a subset of the OTA specification and are used to implement services that support various business functions related to airline travel such as checking flight availability, flight booking, etc. For example, the Search and Availability of flights business function is implemented using the Air_AvailabilityRQ/RS (request/response) message pair. OTA defines common data types (OTA_AirCommonTypes) for the airline messages that form a global type repository of XML Schema components (i.e. simple and complex type definitions) used in the construction OTA Air messages. OTA differentiates between *complex types* (types that contain multiple data elements) and *simple types* (types that contain a single data element). We use a subset of OTA Airline (Air) message schemas for our calculations. Open Travel Alliance typically publishes two (A and B) schema specifications each year. We have used air message schema specifications for our evaluation of schema complexity. Table 1 shows MSE values for 26 OTA Air message schemas for versions 2010B to 2013B. The total value of the MSE for the entire set of messages is also shown at the bottom of the tables. It is evident that MSE increases monotonically for all message schemas (with some minor exceptions) as new (enhanced) schema versions are released, indicating that the complexity of the OTA specification is increasing. The total MSE increases from a value of 143 for the 2010B OTA Specification to 154 for the 2013B OTA Specification, representing a 7% increase. This increase in MSE over the period of four years is probably not significant and indicates relative stability of the OTA messages schemas. The interpretation of the significance of the MSE values and the increase in MSE as new versions are

released requires further analysis. Some insight into the relative complexity of the message schemas can be gained by comparing the MSE of individual messages schema with the value of MSE computed for the global schema elements (OTA_AirCommonTypes) that include all the common elements shared across OTA Air messages. The MSE value for the OTA_AirCommonTypes schema for the 2013B version of the OTA Specification equals to 7.55. It can be noted that the MSE value for the some of the message schemas (e.g. OTA_AirAvailRQ, OTA_AirBookRQ, OTA_AirBookRS) exceeds this value, indicating that the complexity of these schemas is of similar magnitude as the complexity of the entire

Table 1. Values of MSE for OTA Air Message Schema (Versions 2010B–2013B)

OTA Air Message	**2010B**	**2011A**	**2011B**	**2012A**	**2012B**	**2013A**	**2013B**
OTA_AirAvailRQ	6.41	6.41	6.41	7.77	7.77	7.77	7.77
OTA_AirAvailRS	4.72	4.83	4.83	4.83	4.83	4.83	4.83
OTA_AirBookModifyRQ	7.31	7.31	7.32	7.40	7.38	7.35	7.35
OTA_AirBookRQ	7.27	7.27	7.27	7.61	7.60	7.56	7.56
OTA_AirBookRS	7.32	7.32	7.27	7.67	7.67	7.63	7.63
OTA_AirCheckIn	6.41	6.41	6.41	7.04	7.05	7.01	7.01
OTA_AirCheckInRQ	6.50	6.50	6.50	7.10	7.10	7.06	7.06
OTA_AirCheckInRS	6.56	6.56	6.56	7.13	7.14	7.09	7.09
OTA_AirDemandTicketRQ	5.49	5.49	6.56	7.17	7.20	7.18	7.18
OTA_AirDemandTicketRS	4.32	4.32	4.32	4.32	4.32	4.32	4.32
OTA_AirDetailsRQ	3.55	3.55	3.55	3.55	3.55	3.55	3.55
OTA_AirDetailsRS	3.95	4.12	4.12	4.12	4.12	4.12	4.12
OTA_AirFareDisplayRQ	5.50	5.50	5.50	6.87	6.87	6.87	6.87
OTA_AirFareDisplayRS	6.07	6.07	6.07	6.07	6.07	6.07	6.07
OTA_AirFlifoRQ	3.50	3.50	3.50	3.50	3.50	3.50	3.50
OTA_AirFlifoRS	3.91	3.91	3.91	3.91	3.91	3.91	3.91
OTA_AirLowFareSearchRQ	6.26	6.26	6.22	7.30	7.30	7.30	7.30
OTA_AirLowFareSearchRS	6.31	6.31	6.31	7.29	7.29	7.29	7.29
OTA_AirPriceRQ	6.36	6.36	6.36	7.15	7.15	7.14	7.14
OTA_AirPriceRS	6.27	6.27	6.27	6.27	6.27	6.28	6.28
OTA_AirRulesRQ	5.00	5.00	5.00	5.00	5.00	5.00	5.00
OTA_AirRulesRS	5.22	5.22	5.22	5.22	5.22	5.22	5.22
OTA_AirScheduleRQ	4.30	4.30	4.30	4.30	4.30	4.30	4.30
OTA_AirScheduleRS	4.48	4.61	4.61	4.61	4.61	4.61	4.61
OTA_AirSeatMapRQ	4.99	4.99	4.99	5.64	5.64	5.64	5.64
OTA_AirSeatMapRS	5.14	5.14	5.44	5.97	5.71	5.71	5.71
Total MSE	143.13	143.54	144.83	154.80	154.57	154.31	154.31

global schema. In addition to globally defined schema elements individual message schemas include locally defined elements that contribute to MSE, explaining this apparent inconsistency.

5 Conclusions and Further Work

Complexity of domain standard specifications is a major factor inhibiting the evolution of specifications and increasing the maintenance costs of domain applications. There is a need for reliable metrics that estimate the complexity of XML-based standard specification and can be used to identify excessively complex schema design early in the design cycle. In this paper we have proposed a Message Software Entropy (MSE) metric that estimates the complexity of XML schemas and we have used this metric to study the complexity of a subset of the Open Travel Alliance Specification as it evolves over a period of four years and seven version releases. The results indicate monotonic increase in schema complexity as measured by MSE, with the total MSE increasing from 143 for the 2010B OTA Specification to 154 for the 2013B OTA Specification, representing a 7% increase. This increase is probably not significant and indicates relative stability of the OTA specification. We have also noted that the MSE value for some of the message schemas is of similar magnitude as the value of MSE for the global schema elements OTA_AirCommonTypes (7.55), indicating that the complexity of these schemas (e.g. OTA_AirAvailRQ, OTA_AirBookRQ) is similar to the complexity of the entire OTA Air global schema. The current version of the MSE metric is purely based on complex element counts and does not take into account the complexity of the individual elements or the number of levels in the message schema. This makes it easy to interpret the metric, but it also reduces the accuracy of the estimates of schema complexity. MSE metric represents only a first approximation of the complexity of XML schemas, and we are working on refining the MSE metric to take account a range of XML schema structural features.

Acknowledgments. George Feuerlicht wishes to acknowledge the support of Research Centre for Human Centered Technology Design at the Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia.

References

1. OTA. OTA Specifications. (2010) Available from: `http://www.opentravel.org/Specifications/Default.aspx`
2. ebXML. ebXML - Enabling A Global Electronic Market. (2007) Available from: `http://www.ebxml.org/`
3. WC3. SOAP Specifications. (2007) Available from: `http://www.w3.org/TR/soap/`
4. Pautasso, C., Zimmermann, O., Leymann, F.: Restful web services vs. big'web services: making the right architectural decision. in 17th international conference on World Wide Web. Beijing, China: ACM (2008)

5. Glushko, R.J., McGrath, T.: Document Engineering for e-Business. in Proceedings of the 2002 ACM symposium on Document Engineering (DocEng'02). McLean, Virginia, USA: ACM Press New York, NY, USA (2002)
6. Pichler, C., Strommer, M., Huemer, C.: Size matters!? measuring the complexity of xml schema mapping models. in Services (SERVICES-1), 2010 6th World Congress on, IEEE (2010)
7. Feuerlicht, G.: Evaluation of Quality of Design for Document-Centric Software Services. in Service-Oriented Computing-ICSOC 2012 Workshops. Springer (2013)
8. Basci, D., Misra, S.: Entropy as a Measure of Quality of XML Schema Document. The International Arab Journal of Information Technology, 8(1): pp. 75–83 (2011)
9. McDowell, A., Schmidt, C., Yue, K.B.: Analysis and metrics of XML schema (2004)
10. Visser, J.: Structure metrics for XML Schema. Proceedings of XATA (2006)
11. Basci, D., Misra, S.: Measuring and evaluating a design complexity metric for XML schema documents. Journal of Information Science and Engineering, 25(5): pp. 1405–1425 (2009)
12. Nečaský, M.: Conceptual Modeling for XML, ser. Dissertations in Database and Information Systems Series, IOS Press/AKA Verlag (2009)
13. Nečaský, M., Mlýnková, I.: A Framework for Efficient Design, Maintaining, and Evolution of a System of XML Applications. Proceedings of the Databases, Texts, Specifications, and Objects, DATESO. 10: pp. 38–49
14. Nečaský, M., Mlýnková, I.: Five-Level Multi-Application Schema Evolution. Proceedings of the Databases, Texts, Specifications, and Objects, DATESO 9: pp. 213–217
15. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review, 5(1): pp. 3–55 (2001)
16. Mohanty, S.N.: Entropy metrics for software design evaluation. Journal of systems and software, 1981. 2(1): pp. 39–46
17. Bansiya, J., Davis, C., Etzkorn, L.: An entropy-based complexity measure for object-oriented designs. Theory and Practice of Object Systems, 5(2): pp. 111–118 (1999)
18. Olague, H.M., Etzkorn, L.H., Cox, G.W.: An Entropy-Based Approach to Assessing Object-Oriented Software Maintainability and Degradation-A Method and Case Study. in Software Engineering Research and Practice. Citeseer (2006)
19. Ruellan, H.: XML Entropy Study. in Balisage: The Markup Conference (2012)
20. Thaw, T.Z., Khin, M.M.: Measuring Qualities of XML Schema Documents. Journal of Software Engineering and Applications 6: p. 458 (2013)
21. Tang, R., Wu, H., Bressan, S.: Measuring XML structured-ness with entropy, in Web-Age Information Management. Springer. pp. 113–123 (2012)