Size Matters!? Measuring the Complexity of XML Schema Mapping Models

Christian Pichler, Michael Strommer
Inter-Organizational Systems
Research Studios Austria
Vienna, Austria
{*cpichler, mstrommer*}*@researchstudio.at*

Christian Huemer
Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria
huemer@big.tuwien.ac.at

***Abstract*—Exchanging structured business documents is inevitable for successful collaboration in electronic commerce. A prerequisite, for fostering the interoperability between business partners utilizing different business document standards, is a mapping between different standards. However, the effort involved in creating those mappings is hard to estimate. For example, the complexity of standardized formats is one crucial aspect affecting the effort of the mapping process.**

Therefore, a notion of complexity is desirable for both, manual as well as automatic mapping processes. For this reason we develop an initial set of metrics, based on well established metrics for XML Schema, allowing to analyze the complexity of business document standards. Having such metrics at hand allows estimating the complexity and hence the mapping effort of a business document standard, prior to the actual mapping process. We demonstrate the complexity metrics on three different business document standards from the electronic commerce domain.

***Keywords*-complexity metrics; business document models; business document standards; XML Schema metrics;**

I. Introduction

Exchanging structured business documents is inevitable for successful collaboration in electronic commerce. For exchanging information electronically, standardized formats are required as it is achieved through Standard Developing Organizations (SDOs). These standardized formats are typically created for a particular domain or industry. One example is the Universal Business Language (UBL) [?] which defines business documents such as an electronic invoice or an electronic purchase order. Moreover, different business document standards often co-exist for a particular domain. For instance, for the Austrian market, three business business document standards are predominant including: UBL, ebInterface [?], a local Austrian standard, as well as the United Nations Centre for Trade Facilitation and Electronic Commerce's (UN/CEFACT) Cross Industry Invoice (CII), which has recently been mandated for electronic invoicing within the European Union by the Expert Group on e-Invoicing. However, when adopting a particular business document standard, stakeholders will choose a standard fitting their requirements. The use of different standards anticipates interoperability inhibiting successful collaboration in electronic commerce. For enabling interoperability, it is necessary to create mappings between the business document standards. For example, a business partner utilizes the UBL electronic invoice whereas another business partner uses ebInterface. In case the two business partners engage in a business partnership, interoperability is inhibited. Therefore, to enable interoperability, it is necessary to perform a mapping between UBL's electronic invoice and ebInterface. These mappings are either done manually or through an automated mapping process.

Nevertheless, the effort involved in creating those mappings is hard to estimate due to a number of reasons. Examples include special naming conventions - or in more sophisticated cases the extent to which reuse-oriented concepts are used in business document standards. Likewise algorithms may be analyzed by O notation in order to estimate time complexity. The mapping process itself as being an algorithm may be analyzed using O by means of counting the necessary steps to completion. Similarly, a measure for estimating the effort involved in the mapping processes, be it manually or automatically, a priori would be desirable. Having such metrics at hand allows proper planning prior to accomplishing an actual mapping.

II. Related Work

In the literature one may find quite a few metrics for the analysis of XML Schema schemas, denoted as XML Schemas or schemas in the following. The work done by [?], [?], [?] provides an overview of what's state of the art in the field of schema metrics, which will be discussed in

The work of Research Studios Austria is funded by the Austrian Federal Ministry of Science and Research. Furthermore, this work has been carried out under the research grant Public Private Interoperability (No. 818639) of the Austrian Research Promotion Agency (FFG).

the following. A comprehensive introduction to schema metrics is given by Lämmel et al. [?]. Besides basic size metrics they develop also various complexity metrics and metrics for determining the XSD style most likely used.

However, the metrics described in [?] are agnostic about the structure of an XML Schema and focus on size. Visser [?] presents various metrics that take the structure of a schema into account by adopting well known measurement methods from graphs. As a prerequisite, a graph representation must be computed from a given XML Schema so to measure for example how closely the graph structure is related to a tree structure. Also, measures of recursiveness are identified.

Basci and Misra [?] apply a different method on measuring the complexity of XML related schemas. Although, their proposed metric has been evaluated on DTDs, their results may be easily computed for XML Schemas as well. Their basic idea is to adapt the concept of entropy as a measure of complexity for XML schemas. First, they extract a graph representation of DTDs, where elements and attributes are depicted as nodes and parent-child relationships are represented as edges. Second, they group elements and attributes by computing their similarity based on fan-in and fan-out numbers. At last, the entropy is computed as a discrete set of probabilities, where larger numbers indicate more complex schemas than smaller ones.

In the fields of databases and schema mapping quite a few research approaches exist, covering the topics of complexity and benchmarking, see [?], [?], [?], which is closely related to our approach. The main focus lies on performance of queries for data manipulation and on formal foundations for computational complexity. An evaluation of mapping systems is provided in [?]. They present a benchmark that provides a standard set of test cases covering various problems and challenges in the course of schema mapping.

III. Metrics

A typical mapping scenario is shown in Figure 1, where Schema A is being mapped to Schema B by means of a mapping model. The complexities of each of those modeling artifacts shall be measured by proper metrics. In this section, we develop complexity measures to support the mapping process of schemas. A roadmap for the sections of interest for a specific artifact whose complexity is to be computed is shown in Figure 1.

A. Preliminaries

Schema Mapping. The task of schema mapping may be done manually or automatically, depending on size, structure, name similarity and complexity. In the case of automatic mapping, i.e., matching by means of applications such as Coma++ [?], the results may vary from schema to schema. In either case, the complexity and structure of two XML Schemas affect the resulting mapping model $map_{AB} = Schema_A \leftrightarrow Schema_B$. A mapping meta model relates model elements of XML Schema by means of correspondences with a source and target role. These roles may express cardinality constraints supporting various mapping scenarios.

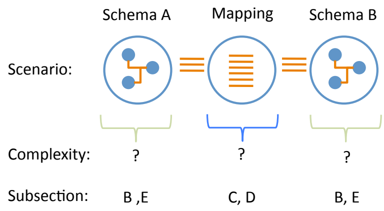


Figure 1. Mapping scenario involving schema complexity.

Standards Landscape. Business document standards may be distinguished into standards defined on the conceptual level and standards defined on the transfer syntax level. Defining a standard on the conceptual level means that a standard is defined using languages such as UML class diagrams. The conceptual representation is then used for generating the transfer syntax from the conceptual model. The transfer syntax may be represented through an XML Schema. An example for a standard defined on the conceptual level are the United Nations Centre for Trade Facilitation and eBusiness (UN/CEFACT) Core Components [?]. Furthermore, UN/CEFACT's XML Naming and Design Rules (NDR) [?] provide proper rules generating XML schemas from conceptual models. On the contrary, standards are also defined on the transfer syntax level using XML Schema withouth any conceptual model. A standard defined on the transfer syntax level is for example ebInterface [?]. In this paper we only address the complexity involved in mapping business document standards defined using XML Schema. The reason for doing so is that the structure of business documents exchanged in electronic commerce is mostly defined using XML Schema.

Complexity. For example, as Ken Holman pointed out in a mail to the UBL developers list [?], the current version of the UBL purchase order covers 830,338 different elements and 2,171,455 attributes when flattening the document structure and taking the combinatorial issues of qualified elements into account. Counting elements and attributes is one way of evaluating the complexity of a business document standard which may be used for quantifying a standard's size. However, the complexity of a business document standard expressed using counting metrics is not necessarily useful for someone interested in structural characteristics of a standard. This raises the question: what is the definition of complexity for a business document standard?

There is no definite answer since the understanding of complexity of a business document standard depends on the quantifications of interest. For example, in the context of mapping business document standards the number of different XML Schema concepts, such as `XSD Redefine` for extending or restricting certain types of another schema document, represent useful complexity information.

B. Basic XML Schema Metrics

In this subsection we develop a combined metric, which tries to capture schema complexity with respect of mapping two schemas. Although, a lot of research has been done so far, we contribute new aspects into the field of schema analysis. Note, that for each partial metric we define that 0 indicates no complexity, and 1 indicates a 100% complexity.

Size-based. One of the first things we mention when we talk about code complexity is size. The same applies to schemas. With XML Schema we may count elements, attributes, lines of code, tags and so forth. An overview of possible counts, both, agnostic and aware of XML as well as XSD is provided by Lämmel et al. [**?**]. In order to easily understand a given schema certain sizes and thresholds should not be exceeded. If, for example, a specified threshold is exceeded this contributes to complexity. Otherwise, the complexity is not being affected so strongly. It is not important here, whether to count the number of lines of code, tags, element declarations or even bytes. It is more like having a method or class in programming not exceeding a certain amount of code to remain easy to understand. We may define a formula of size complexity in terms of

$$\Xi_{size} = 1 - e^{-\frac{\#Tags}{T}}, [0;1] \tag{1}$$

, where $\#Tags$ is the total number of tags (opening and closing) and T is a certain empirical threshold, which separates easy schemas from difficult ones. The ratio $\frac{\#Tags}{T}$ however shall reflect how many times a certain schema deviates from being easy to understand.

Concept-based. In [**?**] the authors point to the significance of XSD language concepts used with a schema. It is now straightforward to use the counts generated from concept usage for the construction of our combined metric. This way of counting concepts shall basically capture the human capability of XML Schema comprehension. The assumption is, that a schema is easier to understand, if only basic concept and features are used. The more advanced features are used the more complex a schema will most likely be. For this reason we count each individual concept occurrence within a schema and assign it a specific weight. It appears that a categorization of XSD features into core concepts, additional and advanced concepts is appropriate. We then may specify *XSD Elements, SimpleTypes, ComplexTypes, Sequences and Attributes* as core, that do not contribute to a schema's overall complexity, assigning core concepts zero weight. This also makes sense as it appears that B2B standards only use a limited set of schema concepts instead of the full expressiveness. Empirical evidence for this is provided by Schmitz et al. [**?**]. Table I maps each XSD concept to a specific weight w_i. Note, that we assigned a weight value of 0.5 to additional concepts like *XSD Choice* and a value of 1 to advanced concepts like the *XSD Any* feature, which fully contributes to complexity. This pragmatic weights may be adjusted if required in the given situation or empirical studies yield further insights. The formula for the metric just described may be defined as follows:

$$\Xi_{concept} = \frac{\sum_{i=1}^{k} w_i f(k)}{N}, [0;1] \tag{2}$$

and

$$f(k) = \begin{Bmatrix} k = 1 & \mid n_1 \\ k = 2 & \mid n_2 \\ k = 3 & \mid n_3 \\ ... & \\ k = m & \mid n_m \end{Bmatrix} \tag{3}$$

where N is the total number of XML Schema objects, n is the number of elements in a specific category, i.e., of a specific concept k, m is the number of different concepts, w_i is the weight measure of one category and f_k is a function determining the number of elements within a category.

Name-based. Typically, names of elements and attributes are not based on some well known ontology, which defines their semantic unambiguously. Also, names are often constructed from separate terms resulting in compound names with ambiguous meaning. For querying the meaning of a certain name one may use WordNet [**?**] or some other lexical resource. However, names like *Orderable Unit Factor Rate Type* or *Inhalation Toxicity Zone Code* from the UBL [**?**] schemas will most likely not be resolved. And if they are, several meanings for that term may exist. Another way to incorporate complexity into schemas is the use of acronyms, which are often only valid within a certain domain, and thus also have multiple meanings in practice. Acronyms in case of domain cross-communication may not be interpreted properly. As names are crucial in the understanding of XML Schemas, we propose a metric that takes the problems described above into consideration.

$$\Xi_{name} = (\#LN + \#WN + \#Acr * 0.05)\frac{1}{N}\frac{1}{3}, [0;1] \tag{4}$$

Concept k	Weight w_i	Weight m_i
XSD Element	0	1
XSD SimpleType	0	1
XSD ComplexType	0	1
XSD Sequence	0	1
XSD Attribute	0	1
XSD Restriction	0.5	1
XSD Extension	0.5	0.5
XSD AttributeGroup	0.5	0.5
XSD Choice	0.5	0.5
XSD Redefine	0.5	0.5
XSD SubstitutionGroup	0.5	0.5
XSD All	0.5	0.5
XSD Any	1	0.5
XSD AnyAttribute	1	0
XSD Group	1	0
XSD Key	1	0
XSD KeyRef	1	0
XSD Union	1	0
XSD List	1	0
XSD Unique	1	0

Table I
Weight assignment table.

, with $\#LN$ the number of long names, $\#WN$ as the number of unanswered Word Net queries and $\#Acr$ as the number of acronyms. The number of long terms may be computed from all elements and attributes N as well as each name Na, with σ denoting the standard deviation, as follows:

$$\#LN = \sum_{i=1}^{N} f(Na_i) \tag{5}$$

$$f(Na) = \begin{Bmatrix} Na.length > 2\sigma & \mid 1 \\ else & \mid 0 \end{Bmatrix} \tag{6}$$

.We compute the number of acronyms by counting every sequence of at least two capital letters within an element, attribute or type name. As this may only be an indicator we multiply this result with an error rate of 0.05.

Combined. Adding up each of the formulas above and treating their contribution to complexity equally we derive a general complexity measure for XML Schemas:

$$\Xi = +\frac{1}{3}\Xi_{size} + \frac{1}{3}\Xi_{concept} + \frac{1}{3}\Xi_{name} \tag{7}$$

. This formula yields a complexity measure < 1, where 0 means nearly no complexity and numbers close to 1 indicate high complexity.

C. Mapping Metric

To measure complexity of a potential mapping task, we need to determine the complexity of the models, i.e., schemas and mappings, involved. Figure 2 visualizes the problem space for estimating the difficulty or even more the feasibility of a mapping task at hand. Obviously, there are two XML Schemas A and B whose complexity is given by α and β, respectively. These numbers may be obtained by applying Ξ on A and B, introduced in the previous section. More interestingly for us is to capture the mapping model's complexity, i.e., the effort of doing the mapping, by some quantitative measure ρ. This measure may not be observed directly but has to be predicted a priori. This fact is captured in Figure 2 by the dashed arrow. The mapping model itself is also influenced by the structure and concept defining Schema M, which is known and thus may be analyzed by applying our complexity measure Ξ. The intuition here is, that the more functionality a mapping language possesses, the harder it is to understand and apply for a modeler.

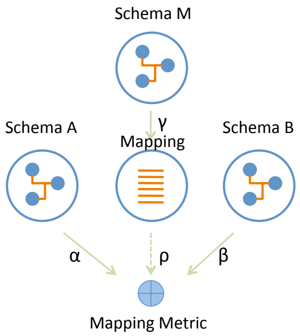


Figure 2. A composite mapping metric with $\alpha = \Xi(A)$, $\beta = \Xi(B)$ and $\gamma = \Xi(M)$.

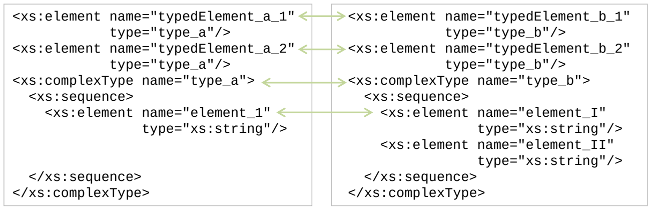
The complexity of a mapping task strongly depends on what may be done automatically and what must be done manually. Generally, a fully-automated match of two schemas may not occur in practice. Names may simply be so different and structures may vary dramatically leading to custom mapping functions and sophisticated transformation rules. If we omit structure and concentrate on names, we may sample from all mappings to be done, and let a matching engine such as Coma++ [?] compute similarity values. The number of all possible mappings is to be estimated from the smaller schema's element and attribute declarations, taking reuse into account. A fixed number of elements and attributes is then picked randomly from the given set of source elements (from the smaller schema) to be mapped. A name is said to be matched if similarity does not fall below a certain threshold. Therefore, we compute the mapping model's complexity ρ as follows:

$$f(Match) = \begin{Bmatrix} Match.similarity > 0.5 & \mid 1 \\ else & \mid 0 \end{Bmatrix} \tag{8}$$

$$\#Matches = \sum_{i=1}^{N} f(Match_i) \tag{9}$$

$$\rho = \frac{\#Matches}{N}, [0;1] \tag{10}$$

, where N is the total of names in the sample.



```
<xs:element name="typedElement_a_1"
            type="type_a"/>
<xs:element name="typedElement_a_2"
            type="type_a"/>
<xs:complexType name="type_a">
  <xs:sequence>
    <xs:element name="element_1"
                type="xs:string"/>


  </xs:sequence>
</xs:complexType>
```

```
<xs:element name="typedElement_b_1"
            type="type_b"/>
<xs:element name="typedElement_b_2"
            type="type_b"/>
<xs:complexType name="type_b">
  <xs:sequence>
    <xs:element name="element_I"
                type="xs:string"/>
    <xs:element name="element_II"
                type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Figure 3. Example Mapping Scenario.

Combined. Considering Ξ for the schemas A, B, and M, we may construct now an overall mapping metric given by:

$$\bigoplus = \frac{1}{3}\alpha + \frac{1}{3}\beta + \frac{1}{3}(\frac{\rho + \delta}{2}), [0;1] \qquad (11)$$

.

D. Enhancing the XML Schema Metrics

Another manner affecting the mapping effort is the extent to which the reuse-oriented concepts of XML Schema are utilized in business document standards. Following the manual mapping process described in [?], it is necessary to map complex types as well as elements, exemplified in Figure 3. Based on the mapping example illustrated, the following rule is established:

$$f(Element) = \begin{Bmatrix} Element.type = Complex & \mid 1 \\ else & \mid 0 \end{Bmatrix} \qquad (12)$$

and

$$\#ComplexElements = \sum_{i=1}^{N} f(Element_i) \qquad (13)$$

and

$$reuse = \frac{\#CT}{\#ComplexElements}, [0;1] \qquad (14)$$

, where $\#CT$ is the total number of complex types, and $\#ComplexElements$ is the number of elements typified through a complex type. As a result, the ratio *reuse* quantifies the reuse within a business document standard, defined using XML Schema.

A low ratio indicates high usage of XML Schema reuse concepts. For instance, in an XML schema which contains one complex type definition as well as ten element declarations typified through the same complex type, the mapping effort remains low since the complex type itself only needs to be mapped once. On the other hand, a high ratio indicates low usage of XML Schema reuse concepts. For example, in an XML Schema which contains then element declarations whereas each element is typified through a different complex type, each complex type needs to be mapped separately.

E. Special Case: XSD 2 CCTS Mapping

The complexity of a specific mapping scenario is strongly dependent on the meta languages used for modeling the data structures. In the previous section we focused on XML Schema as a language for data modeling. However, there exist other languages for data modeling as well, which differ from XML Schema. In such cases the mapping task of schemas becomes a meta mapping task at first. How such meta mappings and hence model mappings may be implemented is shown in [?]. As meta language other than XML Schema, the conceptual UN/CEFACT's Core Components Technical Specification [?] is used. Consequently, as a first step forward engineering produced XML Schemas from the conceptual CCTS models to foster schema to schema mapping. In a reverse engineering step the XML Schema based format may be represented as conceptual CCTS model. For large and concept-rich XML Schemas a metric reflecting mappability may be beneficial before it comes to the actual mapping. Again, the proposed metric uses the concepts used in the XML Schema as predictors. Table I lists the concepts under consideration and the corresponding weight m_i in the third column. As greater numbers are preferable in this case, weights have been adjusted in comparison to our complexity reflecting metrics. We also take CCTS mapping specifics [?] into account.

Idea. If an XML Schema concept such as *XSD Element* is fully mappable to a CCTS equivalent it receives a weight measure of 1. If workarounds and traces as described in [?] have to be introduced a weight of 0.5 is applied. Otherwise, if the mapping is currently not possible at all for some concepts of the XML Schema language we apply a weight of 0.0. The mapping measure Φ may then be computed as

$$\Phi = \frac{\sum_{i=1}^{k} m_i f(k)}{N}, [0;1] \qquad (15)$$

, where N is the total number of XML Schema elements measured by our analyzer tool, n is the number of elements in a specific category, i.e. of a specific concept, k, m_i is the weight measure of one category and f_k is a function determining the number of elements within a category.

IV. Evaluation

The metrics introduced, support estimating the effort involved in creating mappings between different business document standards. The evaluation, which we are also currently working on, assesses the mapping metrics introduced. In the following, we elaborate on the evaluation of the size-based metric introduced in Section III-B, the metric for the special case of XSD to CCTS mapping introduced in section III-E, as well as the

	ebInterface	UBL	CII
Size-based	0.63	0.99	1.00
Special case (XSD 2 CCTS)	0.08	0.16	0.04
Reuse-based	0.44	0.39	0.16

Table II
Evaluation Results.

enhanced mapping metric introduced in Section III-D. The metrics are applied to three business document standards including ebInterface, UBL, as well as the CII, which have been introduced earlier in this paper. The results of the evaluation are illustrated in Table II. Looking at the results of ebInterface, one may observe that all metrics are located between 0.08 and 0.63. The size-based metric having the value 0.63, indicates that the size of the XML schema adds complexity. However, the special case metric reflects high mappability of the business document standard, thus reducing the mapping complexity. Also, the reuse-based metric indicates high reuse within the XML schema reducing the mapping complexity as well.

Similar results may be observed for UBL where the metrics indicate high reuse within the XML schema as well as proper mappability of the business document standard. For UBL, the size-based metric converges towards 1.0, indicating that the size of the XML schema is complex. On the contrary, the special case metric having the value 0.16, as well as the reuse-based metric of 0.39, indicate propper mappability of the business document standard. The CII shows comparable characteristics as UBL. Generally, the size-based metric inidicates high complexity of the standard. However, the special case metric as well as the reuse-based metric, both having values less than 0.2, indicate high mappability of standard.

V. Conclusion and Future Work

The contribution of this paper is twofold. First, we have developed a set of new metrics for the analysis of XML Schema and more interestingly for the process of schema mapping. Second, we provide first evaluation results of a subset of these metrics for the domain of business document standards. For future work we strive for further enhancement of our metrics, especially our metric considering reuse may be improved several directions. Our mapping metric may also incorporate a correlation between the two schemas, in order to reflect time saving effects for mapping same concepts. Also, we plan to look at various standards which are candidates for mapping in practice. This will lead to further insights in the viability of the our qualitative measures. All of the presented metrics shall be implemented within an open source tool to test and evaluate them in a broader way.

References

[1] OASIS, "Universal Business Langauge v2.0," www.oasis-open.org/committees/ubl/, Last Visit: March 2010, 2006.

[2] AustriaPRO, "ebInterface 3.0, Der österreichische Rechnungsstandard," http://www.ebinterface.at/, Last Visit: March 2010, 2009.

[3] J. Visser, "Structure Metrics for XML Schema," in *Proceedings of the XATA 06, XML: Aplicacoes e Tecnologias Associadas, Portalegre, Portugal*, February 2006.

[4] R. Lämmel, S. Kitsis, and D. Remy, "Analysis of XML Schema Usage," in *Proceedings of the XML 2005 Conference, Atlanta, Georgia*, November 2005.

[5] D. Basci and S. Misra, "Entropy Metric for XML DTD Documents," *SIGSOFT Software Engineering Notes*, vol. 33, no. 4, pp. 1–6, 2008.

[6] B. Alexe, W.-C. Tan, and Y. Velegrakis, "STBenchmark: Towards a Benchmark for Mapping Systems," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 230–244, 2008.

[7] A. Schmidt, F. Waas, M. Kersten, M. J. Carey, I. Manolescu, and R. Busse, "XMark: A Benchmark for XML Data Management," in *Proceedings of the VLDB 2002 28th intern. conf., Hong Kong, China*, 2002.

[8] G. Gottlob and P. Senellart, "Schema Mapping Discovery from Data Instances," *J. ACM*, vol. 57, no. 2, pp. 1–37, 2010.

[9] D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm, "Schema and Ontology Matching with COMA++," in *Proceedings of the ACM SIGMOD, Baltimore, Maryland, USA, June 14-16*, 2005.

[10] UN/CEFACT, *Core Components Technical Specification 3.0*, http://www.unece.org/cefact/rsm/rsm_index.htm, Last Visit: March 2010, 2009.

[11] UN/CEFACT, *XML Naming and Design Rules 3.0, ODP5*, 2008.

[12] K. Holman, "UBL Catalogue analysis," http://markmail.org/message/o3ra6fffiw6mu7jw, Last Visit: March 2010, 2008.

[13] V. Schmitz, J. Leukel, and F. dieter Dorloff, "Does B2B Data Exchange Tap the Full Potential of XML Schema Languages," in *Proceedings of the 16th Bled Electronic Commerce Conference, June 2003, Bled, Slovenia*, 2003.

[14] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, pp. 39–41, 1995.

[15] M. Strommer, C. Pichler, and P. Liegl, "On Mapping Business Document Models to Core Components," in *Proceedings of the HICSS 2010, 5-8 January 2010, Koloa, Kauai, HI, USA*, 2010.