XBench - A Family of Benchmarks for XML DBMSs

Benjamin Bin Yao and M. Tamer Özsu
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{bbyao,tozsu}@uwaterloo.ca

John Keenleyside
IBM Toronto Software Laboratory
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7
keenley@ca.ibm.com

Technical Report CS-2002-39
December 2002

Abstract

XML is beginning to be extensively used in various application domains, and as a result, large amounts of XML documents are being generated. Researchers in both industry and academia have proposed a number of approaches to efficiently store, manipulate, and retrieve XML documents. The individual performance characteristics of these approaches as well as the relative performance of various systems is an ongoing concern.

The range of XML application and the XML data that they manage are quite varied and no one database schema and workload can properly capture this variety. We propose a family of XML benchmarks, collectively call XBench, to measure and evaluate the performance of different approaches to deal with the management of XML documents. The family is defined according to a classification of applications, and each class has its own database and workload.

We discuss the general requirements for an XML DBMS benchmark, followed by a detailed explanation of the XBench, including the methodology of database generation, the workload, and the setup of test environment. A brief discussion of other existing XML benchmarks and comparison among them will be given as well.

Contents

Chapter 1

Introduction

XML (eXtensible Markup Language) [1], a subset of SGML (Standard Generalized Markup Language) [2], is a specification proposed by the World Wide Web Consortium (W3C) to complement HTML (Hypertext Markup Language) [3] for electronic data representation and exchange on the Web. Because it is self-describing, it is beginning to be extensively used in various application domains such as chemistry, biology, medicine and e-business. As a result, large amounts of XML documents are being generated, which has raised the demand for their efficient management.

Researchers in both industry and academia have been focusing on efficiently storing, manipulating, and retrieving XML documents. A number of approaches have been proposed including using flat file systems (e.g., Kweelt [4]), extending mature DBMS technologies such as relational DBMS (e.g., IBM DB2 XML Extender [5], Oracle, and Microsoft SQLS Server) or object-oriented DBMS (e.g., Ozone [6]), and building native XML repositories (e.g., Tamino [7], Natix [8], and Xyleme [9]).

The individual performance characteristics of these approaches as well as the relative performance of various systems is an ongoing concern. There are several major domain-specific database benchmarks from the Transaction Processing Performance Council (TPC) such as TPC-C for OLTP (on-line transaction processing), TPC-H and TPC-R for decision support, and TPC-W for web e-commerce. However, these benchmarks do not directly address the requirements of XML databases (e.g., nested document structures and path expression queries). Several application-level XML benchmarks have also been proposed: Xmach-1 [10], Xmark [11] and XOO7 [12]. Each of these benchmarks assumes a single application and defines the database schema and workload accordingly.

The range of XML application and the XML data that they manage are quite varied and no one database schema and workload can properly capture this variety. We propose a family of XML benchmarks, collectively called XBench, to measure and evaluate the performance of different approaches to deal with the management of XML documents. The family is defined according to a classification of applications.

The rest of this document is organized as follows. The next section is dedicated to a discussion of the general requirements for an XML DBMS benchmark family. Section 3 explains how to set up the XBench benchmark. The benchmark is described in Sections 4 and 5; Section 4 focuses on database generation while Section 5 defines

the workload. We will discuss three existing XML benchmarks with respect to our requirements and identify what requirements they do not satisfy, followed by a brief comparison between these three benchmarks, in Section 6. In the conclusion, we state our contributions and what needs to be done in the near future.

Chapter 2

XML Benchmark Desiderata

A useful domain-specific benchmark should be relevant, portable, scalable and simple [13]. Domain-specificity is accomplished in XBench by the definition of a specific database and workload design for a given application domain. Each of these are developed using an identical methodology, thus producing a family of benchmarks. In addition to these general principles, the following specific requirements guide the development of XBench benchmark.

2.1 Text and Non-text Data

XML data can come from text documents or formatted data that are converted to XML. The XML data generator should support both types of XML documents. Two key features of text documents that differentiate them from non-text documents are the importance of the order of elements, and the possibility of mixed content of elements (elements contain sub-elements and text). Typical instances of such text documents are digital libraries and Web pages.

In non-text documents, element types may contain only child elements or only character data. These type of documents can be further specialized as *schema-based documents* (also referred as *structured documents*) or *schema-less documents* (also referred as *semi-structured documents*). Schema-based data are similar to traditional database data, but include more complex structure (nested elements), such as e-commerce catalog data.

2.2 Single and Multiple Document Environments

Some XML databases contain a single large (and deep) XML document (e.g., e-commerce catalogs) while others contain many smaller and shallower documents (e.g., digital library). The benchmark should be able to test both of these scenarios.

2.3 XML Schema and DTD

The benchmark should deal with well-formed XML documents, with or without associated schemas or DTDs against which the documents can be validated. This is important since these schemas and DTDs can be exploited by query optimizers.

2.4 Scalability

The sizes of real XML documents vary. For instance, some e-commerce XML documents, such as catalog, can be very large (500MB and up) while in a digital library, there are large number (500,000 and up) of relatively small documents (40KB to 600KB with occasional larger ones). Therefore, the XML data generator should be able to simulate these extremes.

2.5 XML Documents

The structures of generated XML documents should include both balanced tree and unbalanced tree structures. Most XML documents have unbalanced tree structures, while some XML documents, have flat and balanced tree structures, especially those XML documents that are directly converted from relational data. Therefore, it is likely that conventional DBMSs (especially relational DBMSs) can performance well on XML documents with balanced tree structures, since they can be easily mapped to simple relations. Native XML repositories, on the other hand, may outperform other DBMSs when it comes to unbalanced tree structured XML documents.[1]

Besides basic XML features (elements and attributes), documents should exploit more XML features such as links, notations, entities (internal, external and parameter entities for DTDs), and namespaces.

2.6 Workload Composition

The performance of an XML DBMS will be measured by how much time it takes for different types of operations. The operations include queries, updates and bulk loads. Most benchmarks only consider queries, but the other workloads are also important. The semantics of the queries should be made precise by specifying them in a standard language such as XQuery [14].

Little effort has been made to efficiently update data in an XML DBMS. Most of the current XML DBMSs provide little or no support for updates. The reasons include the absence of a well-defined update semantics and the difficulty of implementation. However we believe that like other conventional DBMS, a XML DBMS should have full functionalities including updates.

[1]Nevertheless, storage mechanism, index techniques and many other important factors decide the overall performance of a DBMS. Ideally a DBMS should provide good performance in dealing with all kinds of XML documents.

Update operations consist of insertion, deletion and replacement. In the context of XML DBMS, update operations can be at the document level or at the element/attribute (type names and contents) level. An initial attempt at update semantics is given in [15]. The benchmark should capture those operations in anticipation.

Bulk loading XML documents into a database is not time critical but various systems may show significant differences in time depending on the approaches they use to address storage management. Even within the same approach like relational DBMSs, different mapping mechanisms may lead to different results. It would be appropriate to measure those facts in order to better understand the effects of different storage mechanisms.

The benchmark should define two kinds of bulk loading operations: load one large document into a XML database, and load a large number of small documents into a database.

2.7 Workload and XQuery

XQuery is the standard query language for XML. It provides a rich set of functionalities that should be exercised. These features include structure/content based searches, regular path expression, reconstruction of complex documents or document fragments, as well as conventional database functions like exact match, join, sort and aggregation. A workload should exercise all or at least a substantial part of XQuery functionality. The XML Query Use Cases [16] is a collection of examples of how end users utilize XQuery to access and manipulate XML documents. The workload of a benchmark should cover all the scenarios mentioned in the Use Cases.

2.8 Implementation Independence

The XML benchmark should be independent of a particular implementation of the XML DBMS. The benchmark should be able to evaluate any XML DBMS developed for different platforms.

In the first version of the benchmark, we additionally place the following restriction. We consider non-distributed XML databases to prevent any side effects caused by distributed environments, such as network overhead and communication costs.

Chapter 3

System Under Test (SUT) Description

As mentioned in the previous section, the benchmark is set up in a single machine environment. The architecture of the SUT is depicted in Figure 3.1.

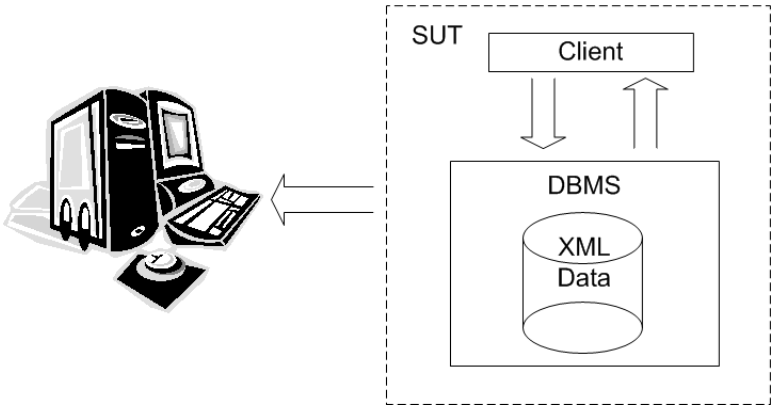

Figure 3.1: System Under Test

The components of the SUT are the following:

Supporting Hardware and Software: A stand alone machine with an operating system on which a DBMS is installed and on which the performance of the DBMS will be measured. The hardware configuration (e.g., CPU speed, main memory and storage media) should be sufficiently powerful to allow tested DBMSs to work properly in such settings and run all the queries[1] on all data sizes.

A DBMS: Any XML DBMS whose performance will be measured. The XML documents are loaded into the tested DBMS. Only one DBMS instance exists in the SUT at one time.

[1]It is possible a DBMS cannot execute some queries because of the implementation of the DBMS, but not because of the hardware configuration

A Client: A client issues workloads to the DBMS. The client can be a human being or a running program. The functions that the client can perform includes the following:

- Randomly picking executable queries from pre-defined query sets, and submitting them to the DBMS in sequence. Since not all DBMSs support XQuery, it's up to end users to implement queries for the specific DBMS, if necessary.
- Receiving the query result of each executable query from the DBMS.
- Measuring the query execution time of each query by keeping track of the time when the client sends a query and also the time when the query result is fully returned.
- Maintaining a log of which queries are executed and the execution time of those queries.

Only one client instance at a time exists in the SUT.

Chapter 4

Database Generation

Database applications are characterized along two dimensions: application characteristics, and document characteristics. Application characteristics indicate whether the database that the application uses is data-centric or text-centric. In *data-centric* (DC) applications, the database stores data that are captured in XML even though the original data may not be in XML. Examples include e-commerce catalog data or transactional data that is captured as XML. *Text-centric* (TC) applications manage actual text documents natively encoded as XML documents. Examples include dictionaries, book collections in a digital library, or news article archives.

In terms of document characteristics, two classes are identified: single document[1] (SD) and multiple document (MD). The *single document* case covers databases, such as an e-commerce catalog, that consist of a single document with complex structures (deep nested elements), and dictionaries, while the *multiple document* case covers those databases that contain a set of XML documents, such as an archive of news documents or transactional data. The result is a requirement for a database generator that can handle four cases: DC/SD, DC/MD, TC/SD, and TC/MD.

The following are some examples of XML documents from each of the four classes mentioned above.

DC/SD: online shopping catalogs and internet movie database (IMDB) [17];

DC/MD: e-commerce transactional data;

TC/SD: GCIDE dictionary [18] and Oxford English dictionary [19];

TC/MD: Reuters news corpus, Springer digital library, Shakespeare works [20], and DBLP data [21].

Many of the real XML documents in DC/SD and DC/MD classes are currently relational data that have been transformed into XML. This may change in the future.

[1] "Document", in this context, refers to an XML document, not to a document as defined in the previous paragraph.

4.1 XML Document Characterization

The following parameters are used to characterize and generate XML documents in each class.

Element types. The collection of all element types that appear in XML documents.

Tree structure of element types. The relationship of all element types in the collection, indicating the parent/child relationships of each pair of element types, if there is such relationship.

Distribution of children to elements. For each element type, the probability distribution of instance occurrences of all its child element (directly sub-element) types.

Distribution of element values to types. The probability distribution of values of each element type.

Attribute names. The collection all all attribute names in an XML documents.

Distribution of attribute values to names. The probability distribution of values of each attribute.

Distribution of attributes to elements. The probability distribution of the attributes to each element.

For each distribution parameter, the minimum and maximum values of that distribution are defined in order to generate finite documents. The extraction of these parameter values are described in the following section.

Although the above are the only parameters that are important for our database generation, we have analyzed actual databases with a number of other parameters. This was done to get a better sense of database characteristics. These are discussed in Appendix A.

4.2 Data Gathering Methodology

XBench family of benchmarks can accommodate the requirements of the four classes of applications identified earlier. We conducted extensive studies to characterize the databases in each category in terms of the parameters identified above. The methodology that was followed is the following:

1. Analyze real XML documents and extract statistical data;
2. Generalize the characterizations of XML documents in each category;
3. Create synthetic data to simulate the XML documents in each category.

4.2.1 Analysis

For the purposes of these benchmarks, synthetic data are generated with domain specific features. Real XML documents that best represent their particular classes are analyzed, considering at least two types of documents in each class. We analyzed relatively big data sets, preferably, larger than 50 MB, to get accurate statistics. All real XML documents that are considered for each classes are listed in Table 4.1.

	Text-Centric (TC)	**Data-Centric (DC)**
Single-Document (SD)	GCIDE Dictionary, Oxford English Dictionary (OED)	TPC-W Catalog Data
Multi-Document (MD)	Reuters News Corpus, Springer-Valag Digital Library	TPC-W Transactional Data

Table 4.1: Real Data in each class

Text-Centric Class

There are sufficient amount of text-centric XML documents. For the single-document class, two dictionaries were analyzed: GNU version of the Collaborative International Dictionary of English (GCIDE) [18] and the Oxford English Dictionary (OED) [19]. For multi-document class, a collection of XML documents that make up the Reuters new corpus was used along with a part of Springer-Valag digital library that was provided to us. The OED dictionary and the Springer data are in SGML and were converted to XML prior to analysis.

Statistical data are collected for these documents in terms of the parameters described in Section 4.1. For instance, given a set of XML documents with a similar tree structure as in Figure 4.1, the following information is gathered:

- the statistical data of occurrences of element type *chapter*,
- the statistical data of occurrences of element type *section* for each instance of element type *chapter*,
- the statistical data of occurrences of element type p for each instance of element type *chapter*,
- the statistical data of occurrences of element type p for each instance of element type *section*,
- the statistical data of lengths of contents of element type p.

Based on the statistics, frequency distributions are computed and standard probability distributions are fit to the data. The data generation tool that is used to

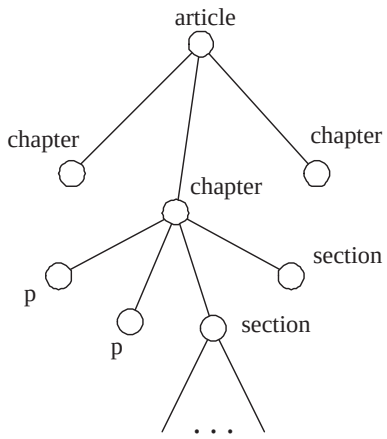

Figure 4.1: A XML Document

produce synthetic data requires that these distributions be one of the standard ones (e.g., exponential, geometric, lognormal, normal/gaussian, or uniform), requiring the elimination of some skews in the data. The tool allows the combination of different probability distributions for each parameter, so local optimizations are possible to fit a standard distribution to part of a frequency distribution.

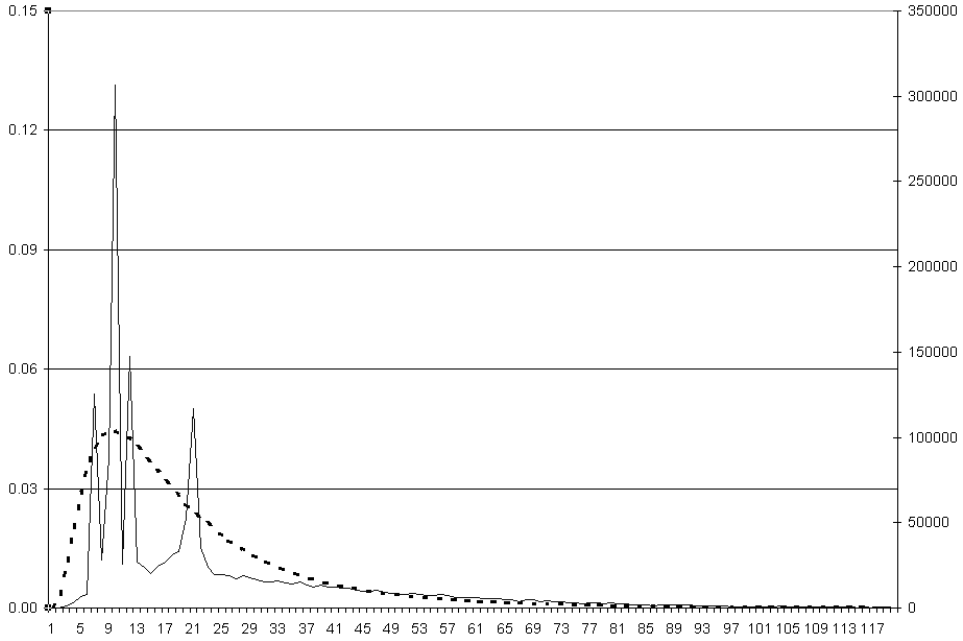

Figure 4.2: Frequency Graph and Distribution of Values of p

Consider the example in Figure 4.2 in which the X axis represents the length (in bytes) of content of p. The solid line (frequency polygon) shows the frequency distribution of the contents of element type p, and the Y axis on the right shows the frequency of each length in all documents. Based the frequency distribution, a standard probability distribution is fit to the frequency distribution.

The dotted line in Figure 4.2 is the approximate standard probability distribution. The Y axis on the left represents the standard distribution value of p. In this case, it is Log-normal distribution with parameters μ (2.7602) and σ (0.7463).

Notice that, in this example, the little spike is ignored in order to fit the frequency distribution into a standard distribution used for database generation. In some cases, if spikes (skewed data) are too big to ignore, a combination of more than one standard distribution is used to represent the statistics. If that is the case, the original data set is split into several sets each of which is considered separately. The final probability distribution is a weighted sum of probabilities of each of these sets.

An important consideration is to find local optimized values of the distribution. χ^2 (so called *Chi-square*) test is used to test the fitness of a distribution[2]. For the above example, given the value of length (x), the real probability of x ($P(x)$), and the standard theoretical probability of x ($f(x)$, based on the probability density function) with initial values of parameters μ and σ, values of these parameters are found that minimize the result of formula $\chi^2 = \sum \frac{[f(x)-P(x)]^2}{f(x)}$.

Data-Centric Class

For data-centric classes, the availability of real XML data for analysis is problematic. Although there are XML specification of transactional data (e.g., Electronic Catalog XML (eCX) [23], Commerce XML (cXML) [24], XMLPay [25], and XML Common Business Library (xCBL) [26]), they are not yet widely used and no substantial repository of XML documents can be found. The data that is available is too small to extract meaningful statistics. Most of the XML documents in the data-centric classes are currently relational that may be translated into XML for communication. Therefore, the schema of the TPC-W benchmark (see [27] for details) is used and is mapped to XML. TPC-W is a mature, and industry-accepted, benchmark for Web-based e-commerce systems that captures the database characteristics of the DC classes of documents.

Elements and attributes are two basic entities in XML documents, and respectively there are element-oriented and attribute-oriented mappings when it comes to relational to XML mapping. Element-oriented approach maps all or most of the data in relations into elements, while attribute-oriented method maps the data into attributes. Compared to attributes, elements have semantic orders and make it easier to handle multiple occurrences. Therefore, element oriented mapping is more commonly used.

Two types of schema mapping approaches have been proposed for single table mapping: *Flat Translation* (FT) [28, 29] and *Nesting-based Translation* (NeT) [29]. FT maps a relation into an element type. Each tuple in the relation is mapped into an instance of the element type, and all the columns are mapped into sub-elements (if it's attribute-oriented, all columns are mapped into attributes of the element). The resulting XML documents of this method are very flat. NeT can generate more hierarchically structured XML documents by applying time consuming aggregations

[2]There are other criteria of goodness of fit, but χ^2 is the most popular one. In [22], it explains why χ^2 makes sense in fit tests.

on duplicate data in the relation. Eventually, columns with higher duplications on value become elements that are closer to the top level. *Constraints-based Translation* (CoT) [30] approach is an extension of NeT that deals with mapping multiple tables related to each other. Although the resulting XML documents of NeT and CoT have low data redundancy, these two approaches are based on statistical aggregate information, ignoring the semantics and knowledge on application domains.

These approaches target general dynamic schema mapping, and depending on the particular schema, one can end up with an XML document that has totally bizarre structure. For instance, a product catalog relation needs to be mapped to XML. Each tuple of the relation represents the information on an item including *item id* (primary key), *item name*, *price*, *quantity in stock*, and *manufacture*, etc. The desired XML file should have a root element which consists of many sub-elements each of which describes the information on an item. One of the possible resulting XML documents should look like the following:

```
...
<root>
  ...
  <item>
    <id>...</id>
    <name>...</name>
    <price>...</price>
    ...
  </item>
  <item>
    ...
  </item>
  ...
</root>
...
```

It's possible that the prices of different items happen to be the same. If previous approaches are applied, because of the duplication of prices (while other information such as item id is unique), the price information gets closer to the root than other information. The resulting XML document could look like the following:

```
...
<root>
  ...
  <price>
    <item>
      <id>...</id>
      <name>...</name>
      ...
    </item>
    ...
```

```
    <item>
      <id>...</id>
      <name>...</name>
      ...
    </item>
    ...
  </price>
  <price>
    ...
  </price>
  ...
</root>
...
```

It is not normal to have a product catalog which is grouped by price information.

In mapping TPC-W schema to XML, a new approach is used that takes into consideration the domain specific semantics. One relation (one table or a join of several tables) is mapped to one XML document. For single tables, tuples are represented as instances of an element type whose sub-elements are element types mapped from columns in that table, as in the FT approach.

	Column1	**Column2**	**Column3**
Tuple1	Value11	Value12	Value13
Tuple2	Value21	Value22	Value23

Table 4.2: A simple table

For example, given a table T_a with 3 columns $Column1$, $Column2$ and $Column3$, and 2 tuples (see Table 4.2), one mapping into XML would look like the following:

```
<?xml version="1.0">
<TableA>
    <Tuple>
        <Column1>Value11</Column1>
        <Column2>Value12</Column2>
        <Column3>Value13</Column3>
    </Tuple>
    <Tuple>
        <Column1>Value21</Column1>
        <Column2>Value22</Column2>
        <Column3>Value23</Column3>
    </Tuple>
</TableA>
```

Sometimes columns in a table are semantically classified into different groups, and both columns and groups are mapped into XML. In the previous example, if

*Column*1 and *Column*2 belong to *Group*1 and *Column*3 belongs to *Group*2, the mapping would look like the following:

```
<?xml version="1.0">
<TableA>
    <Tuple>
        <Group1>
            <Column1>Value11</Column1>
            <Column2>Value12</Column2>
        </Group1>
        <Group2>
            <Column3>Value13</Column3>
        </Group2>
    </Tuple>
    <Tuple>
        <Group1>
            <Column1>Value21</Column1>
            <Column2>Value22</Column2>
        </Group1>
        <Group2>
            <Column3>Value23</Column3>
        </Group2>
    </Tuple>
</TableA>
```

The resulting XML document of this mapping has more depth than the original method. Considering the example of product catalog, *item id* and *item name* could be grouped as primary item information, and *price* and *quantity in stock* could be grouped as pricing information, etc.

One relation is not necessarily mapped into only one XML document. Alternatively, each tuple of a relation can be mapped into a single XML document in some cases. In e-commerce environment, for instance, given a table that contains customer order information, each tuple of the table represents one order. By applying this approach, each resulting XML document holds information of only one order.

The mapping of a join of several tables is done by picking one main table and mapping it to XML first using the above described method. All matching tuples in the second table are inserted as sub-elements of the corresponding tuples in the first table based on foreign key references. The process is repeated recursively for other tables. Accordingly, as the number of jointed tables increases, the mapped XML document gets deeper. For instance, if there are two more tables T_b and T_c besides table T_a in the previous example, and their join is mapped to XML, the resulting XML document would look like the following:

```
<?xml version="1.0">
<TableA>
    <Tuple>
```

```
        <Column1>Value11</Column1>
        <Column2>Value12</Column2>
        <Column3>Value13</Column3>
        <TableB>
            <Tuple>
                <Column1>...</Column1>
                <Column2>...</Column2>
                ...
                <TableC>
                ...
                </TableC>
            </Tuple>
            ...
            <Tuple>
                <Column1>...</Column1>
                <Column2>...</Column2>
                ...
            </Tuple>
        </TableB>
    </Tuple>
    <Tuple>
        <Column1>Value21</Column1>
        <Column2>Value22</Column2>
        <Column3>Value23</Column3>
        <TableB>
            <Tuple>
                <Column1>...</Column1>
                <Column2>...</Column2>
                ...
            </Tuple>
            ...
        </TableB>
    </Tuple>
</TableA>
```

Sections 4.3.3 and 4.3.4 explain, in detail, how to do the mappings from TPC-W to XML. Thus, for DC/SD and DC/MD classes, there is no need to perform abstraction and generalization. Synthetic databases are created based on the characteristics of the adapted TPC-W data model in XML version.

4.2.2 Generalization

Since two XML documents are analyzed in each of the TC classes, it is necessary to generalize the common characteristics of that class. We define a XML document structure for each class that is simpler than those of the original XML documents,

but that incorporates the basic features of those documents.

The process of generalization also involves combining statistical data of two or more semantically same element types (from same or different XML documents), producing a probability distribution or a combination of several distributions. For example, Table 4.3 gives two semantically identical element types <*Lastname*> and <*Surname*> and the statistics of the length of each element type value (for the simplicity of this example, each element type only has a limited number of statistical data).

Lastname	**Lastname**		**Surname**	**Surname**
Length of Value	Frequency		Length of Value	Frequency
10	3		11	4
11	5		12	8
13	8		13	9
14	9		15	10
15	7		16	9
18	3		20	5

Table 4.3: Statistical Data of <*Lastname*> and <*Surname*>

If both of these element types come from the same XML document source, they can simply be merged by summarizing the frequency data together by each length. For instance, for the frequency of the length of value (10), the sum of both element types is $3 + 0 = 3$ (3 from <*Lastname*> and 0 from <*Surname*>), and for the frequency of the length (11), the sum is $5 + 4 = 11$ (5 from <*Lastname*> and 4 from <*Surname*>).

If, however, these two element types from different data sources, the statistical data of these two element types are merged by scaling them with respect to the data size. If, for example, <*Lastname*> comes from data of size 500 MB and <*Surname*> comes from data of size 100 MB, the combined frequency of length (10) is $3 \times 5 + 0 \times 1 = 15$. This is done based on the following two assumptions:

- All the data sources analyzed are equally important. Therefore, if the data size of each data source is different, we scale the statistical data by the data size, in order to get the frequencies of *equally sized* data sources.
- The frequencies change proportionally when the size of a data source changes. For example, if the size changes from 100 MB to 500 MB, the frequency of (10) changes from 10 to 50, as well.

The same rules apply if more than two element types are merged. After the merge, a standard probability distribution or combination of distributions is obtained, based on the merged statistical data using the method described in Section 4.2.1.

4.2.3 Creation

For actual data generation, we use ToXgene [31], which is a template-based tool facilitating the generation of synthetic XML documents. A ToXgene template is created for each of the four classes.

ToXgene templates are written in the ToXgene Template Specification Language (TSL). The syntax of TSL is similar to that of XML Schema [32]. Once users write a template that specifies the attributes of desired XML documents, ToXgene will generate those documents.

ToXgene uses XML Schema as its basis. A *type* can be either a *simpleType* (CDATA literals) or a *complexType* (elements, CDATA literals, or both). TSL uses a *gene* to define an element (so called, an element gene) or an attribute (referred as an attribute gene). One of ToXgene's advantages is that users can specify probability distributions to determine the number of occurrences for elements as well as the contents of elements. Another advantage is that ToXgene provide users tools to share and re-use elements.

The following are some simple examples taken from [31]. The first example shows how to define a simpleType:

```
<simpleType name = "isbn_type">
    <restriction base = "string">
        <pattern value = "[0-9]{10}"/>
    </restriction>
</simpleType>
```

The second example describes how to declare an exponential probability distribution, and later on use this distribution when defining another simpleType:

```
<tox-distribution name = "c1"
    type = "exponential" minInclusive = "5"
    maxInclusive = "100" mean = "35"/>
...
<simpleType name = "my_float">
    <restriction base = "float">
        <tox-number tox-distribution = "c1"/>
    </restriction>
</simpleType>
```

Based on the generalized distribution and abstract structure of real XML documents in an application domain, ToXgene templates are generated to simulate real XML documents. Templates for the database classes are given in Appendix B.

4.3 Databases

Considering the scalability of the benchmarks, four types of database size are defined for each of the database classes: small (10 MB), normal (100 MB), large (1 GB) and huge (10 GB). The default database is normal.

4.3.1 TC/SD

For text-centric/single document category, we analyze dictionaries GCIDE (The GNU version of The Collaborative International Dictionary of English) and OED (Oxford English Dictionary). Table 4.4 summarizes some basic information of these data sets.

Sources	**Num of Files**	**File Size (in MB)**	**Data Size (in MB)**
GCIDE	1	56	56
OED	1	548	548

Table 4.4: Brief Information on Real XML Data in class TC/SD

The common features of XML documents in this class are a big text-dominated document with repeated similar entries, deep nesting and possible references between entries. The generated XML document is a single big XML document (*dictionary.xml*) with numerous word entries. The size of database is controlled by a parameter called *entry_num*. The default value of entry_num is 7333 and file size is about 100 MB. Other important parameters that characterize documents in this class are described in Appendix C.1.

Figure 4.3 gives a visual representation of the schema of XML documents in this class. The complete schema and DTD files for this class are listed in Appendix D.1 and Appendix E.1.

This diagram is generated using XML Spy, a popular XML editor. It clearly illustrates the parent/child tree structure of element types in a XML documents. The rectangles refer to element types. Whatever is on the right side of an element type are its sub-element types. Solid rectangles mean element types are mandatory while dotted ones mean they may or may not exist in a given document. By default, only one instance of a particular element type can appear in real documents, unless otherwise specified under rectangles. $(0..\infty)$ means element types are not mandatory and can appear multiple times, and similarly $(1..\infty)$ means element types are mandatory and can appear multiple time. If an element has attributes, they will appear on the top right corner of the rectangle where the element is located.

4.3.2 TC/MD

We have studied the Reuters news corpus and part of the Springer digital library for the text-centric/multiple document class. Table 4.5 lists some basic information about those data sources.

The features of XML documents in this class are numerous relatively small text-centric XML documents with references between documents, looseness of schema and possibly recursive elements. The target XML documents are a set of XML articles with sizes ranging from several kilobytes to several hundred kilobytes. The size of this database is controlled by *article_num* with default value of 266, and respectively,

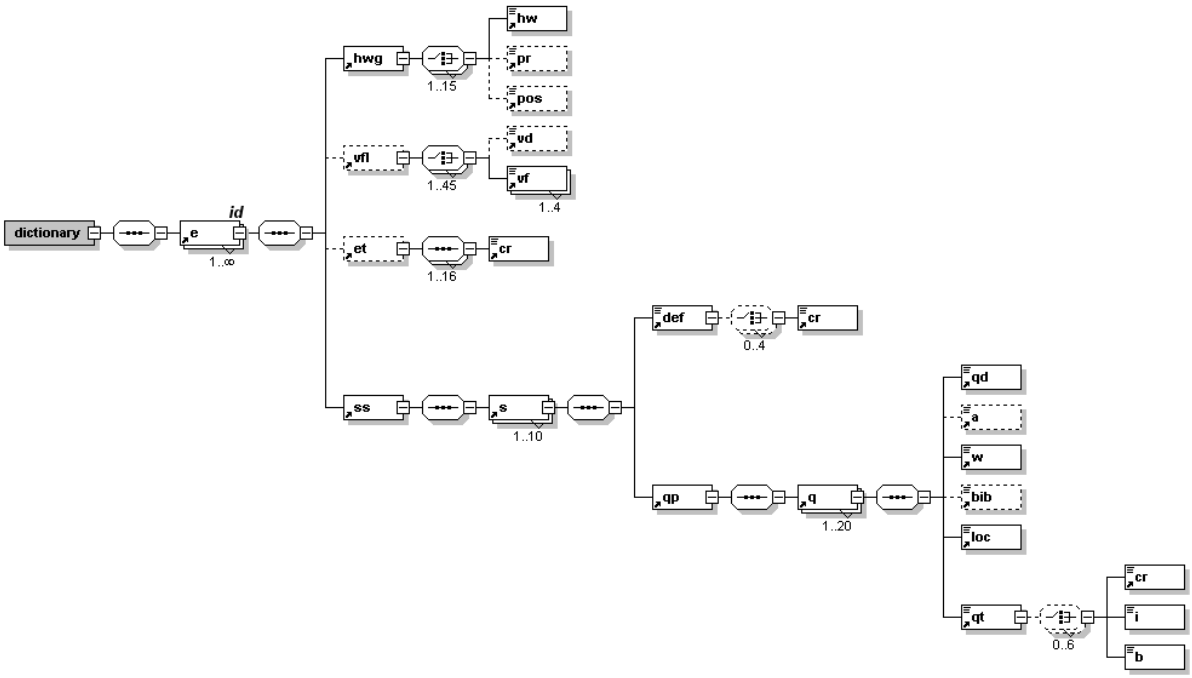

Figure 4.3: Schema Diagram of TC/SD (Dictionary)

Sources	**Num of Files**	File Size (in KB)	**Data Size (in MB)**
Reuters	807,000	[1, 59]	2,484
Springer	196,000	[1, 613]	1,343

Table 4.5: Brief Information on Real XML Data in Class TC/MD

the default data size is around 100 MB. Other important document parameters are listed in Appendix C.2.

Figure 4.4 illustrates the schema information of XML documents in this class. The figure depicts the irregularity of this class of documents. Appendix D.2 and E.2 show detailed schema and DTD information.

4.3.3 DC/MD

As indicated earlier, due to the nature of data-centric documents, it's very difficult to get *real* XML documents, since most of e-commerce and transactional data are still stored in relational databases. TPC-W [27] benchmark schema proves a good starting point for modelling this type of data. Therefore, TPC-W schema is used as the basis of these two classes.

There are eight basic individual tables (relations) in the TPC-W database.

ORDERS – purchase order information such as order date, price and status;

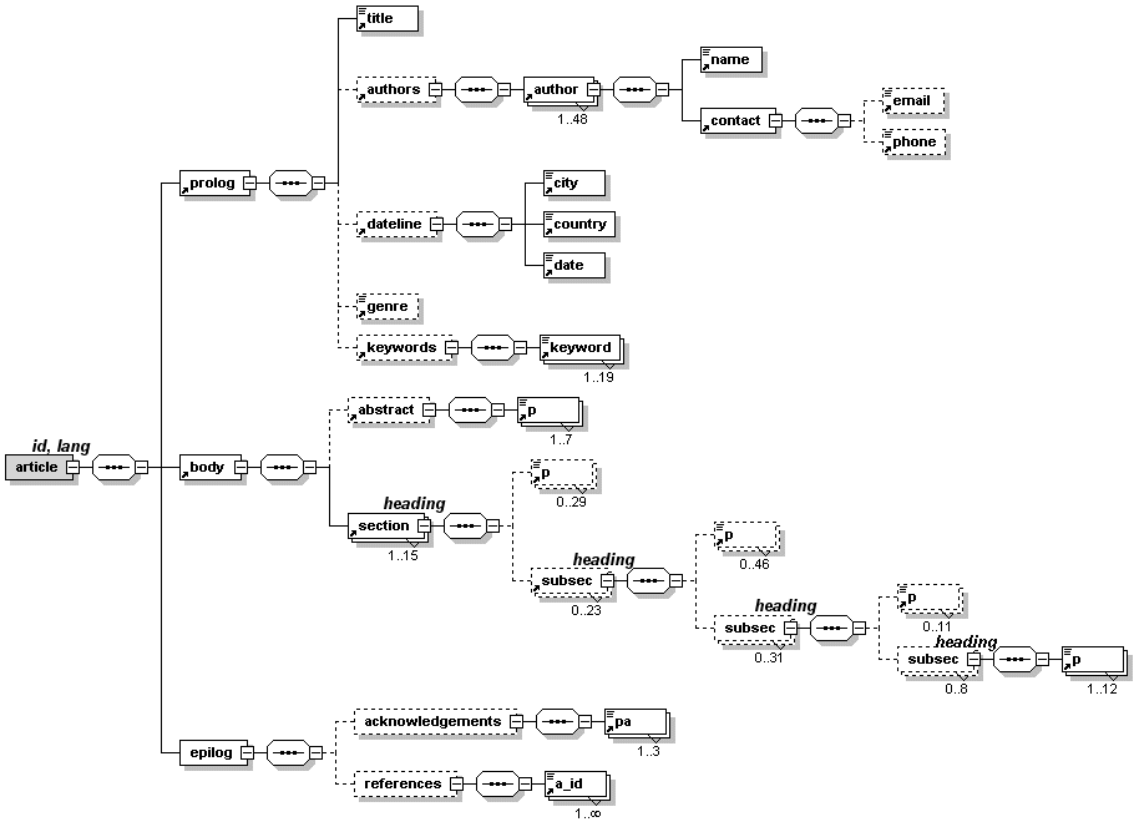

Figure 4.4: Schema Diagram of TC/MD (ArticleXXX)

CC_XACTS – information on credit card transaction for each order such as credit number, name, and authorization id;

ORDER_LINE – information on all detailed items of each order, such as list of ordered items, quantity and discount;

CUSTOMER – customer information such as name, contact information and account balance;

ITEM – item (book) information, such as item title, subject, publisher, and price;

AUTHOR – author information, such as author name and contact information;

ADDRESS – address information for customer;

COUNTRY – country information including name, currency and exchange rate.

The relationship of all these tables is depicted in Figure 4.5. The arrows point in the direction of one-to-many relationship between tables. Dotted lines indicate one-to-one relationship.

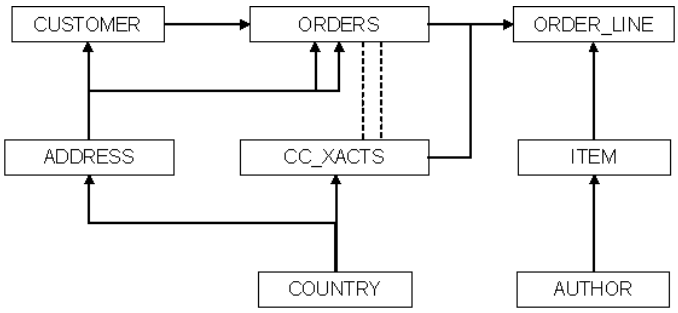

Figure 4.5: The Simplified TPC-W Data Model

Data-centric multiple documents are transactional and are primarily used for data exchange. Thus, the tags are more descriptive and contain less text content. Usually, the structure is more restricted (in terms of irregularity) and flat (less depth) since most of the data came originates in relational databases.

Due to the nature of these documents, we use the FT approach to map five of the TPC-W basic tables (CUSTOMER, ITEM, AUTHOR, ADDRESS and COUNTRY) to separate XML documents, called *Customer, Item, Author, Address,* and *Country*, respectively. Tables ORDERS, ORDER_LINE and CC_XACTS (tables ORDERS and ORDER_LINE have one-to-many relationship and tables ORDERS and CC_XAVTS have one-to-one relationship), are joined together into one big table and mapped into multiple XML documents called *OrderXXX* (XXX represents the number of a particular document). Each order XML document contains exactly one order information which come from all these three tables. The mapping method for multiple tables, described in Section 4.2.1, is used in order to increase some levels of depth. All columns are mapped into elements except primary keys, which are mapped into attributes. The primary key CX_O_ID of table CC_XACTS and the primary key OL_O_ID of

table ORDER_LINE ignored since it has been merged with table ORDERS. The primary key CX_O_ID and OL_O_ID cease to exist. The declaration of element/attribute types are transformed from the TPC-W accordingly. The parameters for document characterization in this class are in Appendix C.4.

Figures 4.6-4.11 describe the structures of these XML documents, and the XML schemas and DTDs for class DC/MD are described in detail in Appendix D.4 and E.4, respectively.

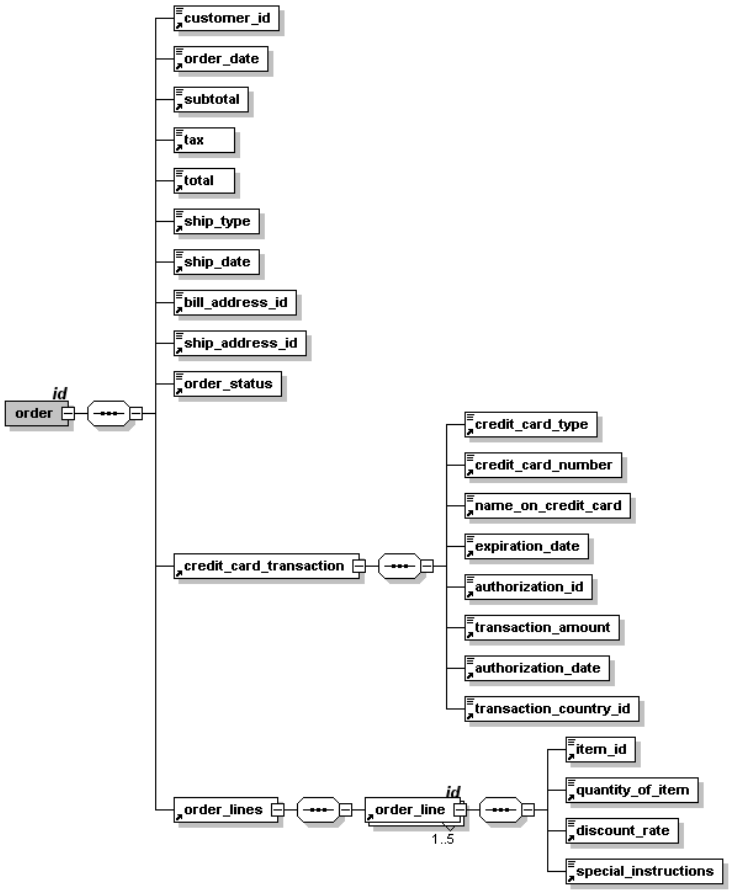

Figure 4.6: Schema Diagram of DC/MD (OrderXXX)

4.3.4 DC/SD

XML documents belonging to the class of data-centric/single document are similar to TC/SD in terms of structure but with less text content. However, the schemas of these data tend to be more strict in the sense that there is less irregularity in DC/SD than in TC/SD, since most of the XML documents in DC/SD are translated directly from relations.

In order to create a catalog data structure from TPC-W relational data model, table ITEM is picked as base, along with other three basic tables (AUTHOR, ADDRESS and COUNTRY). Two more tables are also created that do not exist in

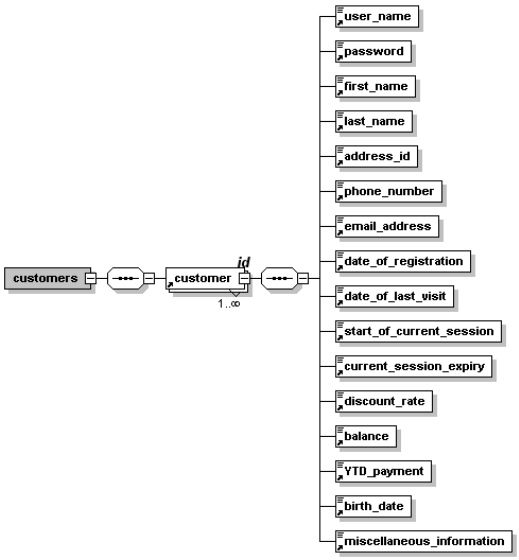

Figure 4.7: Schema Diagram of DC/MD (Customer)

TPC_W.

AUTHOR_2 includes other author information such as: mailing address, phone and email information;

PUBLISHER consists of name, FAX, phone and email address.

Figure 4.12 describes the contents of these two tables, and the relationships among all six tables. All columns in these two tables are either taken from the other four tables or from CUSTOMER table. The definition of column FAX is exactly the same as the definition of column C_PHONE in relation CUSTOMER.

Again, the arrows point in the direction of one-to-many relationship between tables. As shown, table ITEM has one-to-many relationship with table AUTHOR, which means that one item can have more than one author. A cardinality of (1:4) is adapted from the relationship between parts and suppliers in TPC-H[3].

All these six tables are joined together and mapped to an XML document called *Catalog*, using the method described in Section 4.2.1. The visual presentation of the tree structure of all element types are shown in Figure 4.13.

By joining all these tables, more depth is added to Catalog. Some elements are classified and grouped as well. For instance, elements *thumbnail* and *image* are grouped as *media*; *mailing_address, phone_number* and *email_address* are classified as *contact_information*. Element *size_of_book* is split into there sub-elements (*length,*

[3]The TPC-H benchmark, another benchmark proposed by TPC, is a benchmark to measure decision support systems in relational DBMS world. It models a general business environment, in which sellers manage and sell products worldwide. The benchmark also consists of a suite of queries designed to simulate transactions in such scenario over a standard database. The database contains information on parts, suppliers, customers and order transactions. Although the data model of TPC-H does not fit in the requirements of XML benchmark, the cardinality between parts and suppliers is a complement to TPC-W in designing the database for DC/SD class.

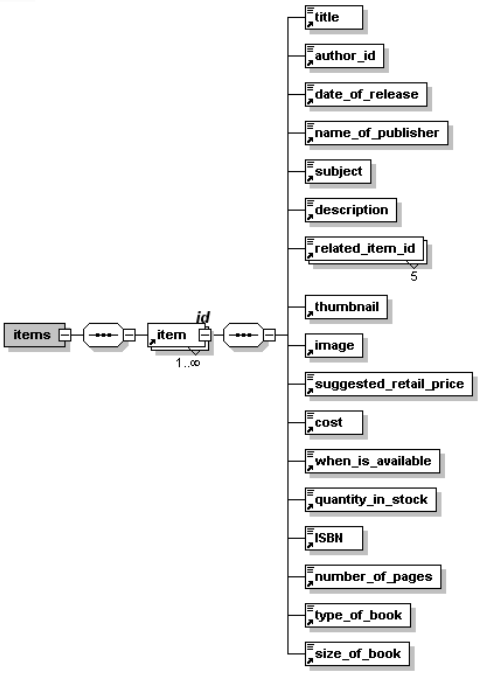

Figure 4.8: Schema Diagram of DC/MD (Item)

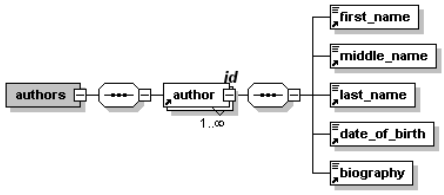

Figure 4.9: Schema Diagram of DC/MD (Author)

width, and height). The primary key of table ITEM is mapped as an attribute (ID type)[4] of element *item* and accordingly, the type of related *item_id* is declared as IDREF.

Other irregularities include:

- Second to fourth *authors* are optional;
- The second instance of element type *street_address* is optional;
- All instances of element type *related_item* are optional.

The detailed specifications for contents/values of element types/attributes are adapted from the TPC-W benchmark. The parameters of document for this class is shown in Appendix C.3. The schema and DTD for DC/SD class are in Appendix D.3 and E.3.

[4]According to the definition of an ID type, the first digit of ID type instance must be a character. Given that the type of the primary key at table ITEM is number, the value of the ID type is a combination of a character and the original value from the table.

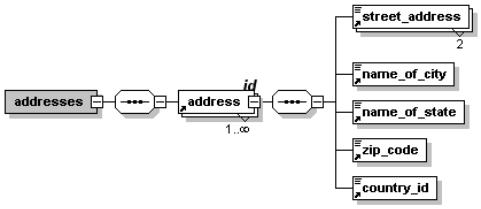

Figure 4.10: Schema Diagram of DC/MD (Address)

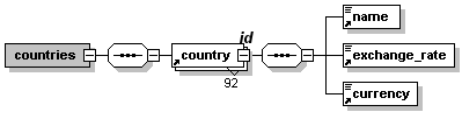

Figure 4.11: Schema Diagram of DC/MD (Country)

4.3.5 Summary

Table 4.6 describes the generated XML documents for classes DC/SD and DC/MD as well as TC/SD and TC/MD as comparison.

Sources	**Size Pars**	**# Files**	**Small**	**Normal**	**Large**	**Huge**
TC/SD (dictionary.xml)	entry_num	1	733	7333	73333	733333
TC/MD (articleXXX.xml)	article_num	[26,26666]	26	266	2666	26666
DC/SD (catalog.xml)	num_items	1	2500	25000	250000	2500000
DC/MD (orderXXX.xml)	num_orders	[2592, 2592000]	2592	25920	259200	2592000
DC/MD (customer.xml)	num_customers	1	2880	28800	288000	2880000
DC/MD (item.xml)	num_items	1	1000	10000	100000	1000000
DC/MD (author.xml)	num_authors	1	250	2500	25000	250000
DC/MD (address.xml)	num_addresses	1	5760	57600	576000	5760000
DC/MD (country.xml)	fixed (92)	1	-	-	-	-

Table 4.6: Brief Information on Synthetic XML Data

TC/SD has only one XML document *dictionary.xml* with the default 7333 entries. The size of the document is adjusted by changing the value of *entry_num* in the template file.

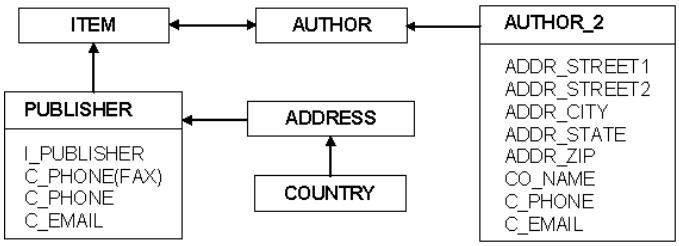

Figure 4.12: Catalog Relational Data Model

TC/MD has default 266 files, from *article1.xml* to *article266.xml*. The size of each individual file ranges from 2 KB to 1500 KB. Again, the number of files is scalable.

DC/SD has *catalog.xml*, one single catalog document with the default entry of 25000 and the resulting database is 100 MB. The number of items is changeable.

DC/MD has *customer.xml, item.xml, author.xml, address.xml*, and *country.xml* five files and multiple *orderXXX.xml* files depending on the number of items specified (from 2592 to 2592000). The file sizes of first four are scalable, but the *country.xml* is fixed. The size of each *orderXXX.xml* ranges from 1 KB to 3 KB. All files simulate TPC-W transactional data. For a default sized database of 100 MB, the number of emulate browser is 100 and the number of items is 1000.

4.4 Database Generation Templates

As indicated above, we use ToXgene to generate the actual database. Thus, templates are written for each class of XML database. These templates are given in Appendix B.

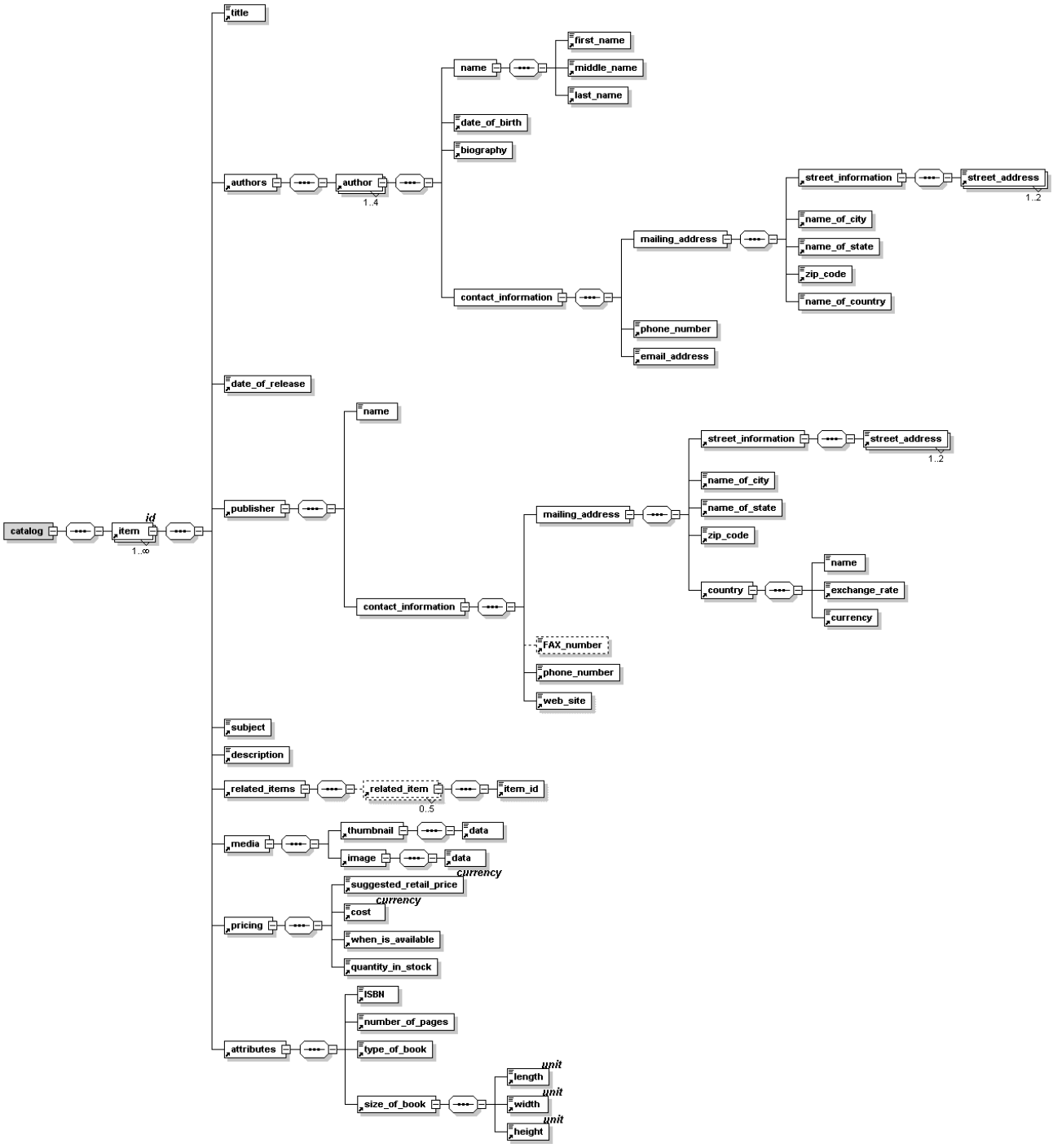

Figure 4.13: Schema Diagram of DC/SD (Catalog)

Chapter 5

Workload

As mentioned in Section 2, the workload should include three parts: queries, updates and bulk loads. In this first version of XBench, we only focus on queries that challenge a DBMS with XML-specific features as well as conventional DBMS functionalities. In the following the workload is discussed along two dimensions: by functionality and by document class. All queries are expressed in natural language, followed by XQuery expression. The workload subsumes all of XQuery's functionality, as captured by the XML Query Use Cases [16].

5.1 Queries by Functionality

5.1.1 Exact Match

This type of queries require string exact match with a specified and possibly long path expressions, depending on the levels of predicates being queried in XML documents. The following queries test the capability of the engines in dealing with predicates at different levels.

Shallow Queries. These are queries that match only at the top level of XML document trees, and include queries ending with $Q1$[1]. For instance, TC/MD_Q1[2], TC/SD_Q1, DC/MD_Q1, and DC/SD_Q1.

Deep Queries. Similarly, these are queries that match the nested structure of XML document tree, and include queries ending with $Q2$. For instance, TC/MD_Q2, TC/SD_Q2, DC/MD_Q2, and DC/SD_Q2.

5.1.2 Function Application

The queries ending with $Q3$ challenge the system with aggregate functions such as *count*, *avg*, *max*, *min* and *sum*.

[1] In the rest part of this section, we will use this labelling convention to refer to the queries.

[2] TC/MD_Qi: Queries for the class of Text-Centric Multiple Document, TC/SD_Qi: Queries for the class of Text-Centric Single Document, DC/MD_Qi: Queries for the class of Data-Centric Multiple Document, and DC/SD_Qi: Queries for the class of Data-Centric Single Document.

5.1.3 Ordered Access

In XML documents, ordering of elements is important. The system should be able to preserve the orders.

Relative Order. These queries are used to search preceding or following data based on the current matching position, including queries ending with $Q4$.

Absolute Order. These queries return data based on its order in a document, including queries ending with $Q5$.

5.1.4 Quantifier

The following queries test for the existence of some elements that satisfies a condition, or whether all the elements in the same collection satisfy a condition.

Existential Quantifier. These include queries ending with $Q6$.

Universal Quantifier. These includes queries ending with $Q7$.

5.1.5 Regular Path Expressions

Regular path expressions are inherent in many XML languages including XPath, XQuery, and XSLT. A good query processor should be able to optimize path expressions.

Unknown Element Name. These queries contain path expressions where one element name in the path is unknown. They include queries ending with $Q8$.

Unknown Subpaths. Similarly, more than one consecutive element names are unknown in the path expression. They include queries ending with $Q9$.

5.1.6 Sorting

Even though the generic data type of element content in XML documents is string, users may cast the string type to other types. Therefore the system should be able to efficiently sort values both in string and in non-string data types.

By String Types. These include queries ending with $Q10$.

By Non String Types. These include queries ending with $Q11$.

5.1.7 Document Construction

One of the big problems experienced by some relational DBMSs in storing XML documents is their poor performance on document reconstruction; some systems cannot even preserve the document's original structure. However, structure is very important to XML documents, especially to text documents. These queries test this feature.

Structure Preserving. These queries retrieve fragments of original documents with original structures, including queries ending with $Q12$.

Structure Transforming. These queries construct document fragments with new structures, which include queries ending with $Q13$.

5.1.8 Irregular Data

XML databases may not have strict schemas as their relational counterpart does. Their schemas are more flexible and may have a number of irregularities. The queries in this group test the ability of query processors in handling these irregularities.

Missing Elements. They include queries ending with $Q14$.

Empty (Null) Values. They include queries ending with $Q15$.

5.1.9 Retrieve Individual Documents

It is an essential function of an XML DBMS to retrieve individual XML documents efficiently while preserving the contents of those documents. The queries ending with $Q16$ fall in this category.

5.1.10 Text Search

Text search plays a very important part in XML document systems. The integration of information retrieval (IR) technologies with database querying is an emerging area of study. The systems should at least be able to handle the following cases efficiently.

Uni-gram Search. These query documents contain one particular word, including queries ending with $Q17$.

Bi-gram and N-gram Search. Similarly, these query documents contain multiple words, including queries ending with $Q18$.

5.1.11 References and Joins

Data-centric documents usually have references to identify the relationship between related data, even among different XML documents. Sometimes users want to combine separate information together using join by values. The queries ending with $Q19$ test this feature.

5.1.12 Datatype Cast

As mentioned before, the element values in XML documents are String type, but sometimes they need to be cast into other data types. The queries ending with $Q20$ measure this scenario.

Index

Table 5.1 lists all queries with respect to the features that they are supposed to test.

5.2 Queries by Document Class

5.2.1 Text-Centric Multiple Document

The following queries are based on an implicit (unnamed) input data set, which is a collection of article documents ("articleXXX.xml").

- **TC/MD_Q1** *Return the title of the article that has matching id attribute value (1).*

```
for $art in input()/article[@id="1"]
return
    $art/prolog/title
```

- **TC/MD_Q2** *Find the title of the article authored by (Ben Yang).*

```
for $prolog in input()/article/prolog
where
    $prolog/authors/author/name="Ben Yang"
return
    $prolog/title
```

- **TC/MD_Q3** *Group articles by date and calculate the total number of articles in each group.*

```
for $a in distinct-values (input()/article/prolog/dateline/date)
let $b:=input()/article/prolog/dateline[date=$a]
return
    <Output>
        <Date>{$a/text()}</Date>
        <NumberOfArticles>{count($b)}</NumberOfArticles>
    </Output>
```

Queries	Functionality	TC or DC
...Q1	Top level exact match	Both
...Q2	Deep level exact match	Both
...Q3	Function application	Both
...Q4	Relative ordered access	Both
...Q5	Absolute ordered access	Both
...Q6	Existential quantifier	Both
...Q7	Universal quantifier	Both
...Q8	Regular path expressions (unknown element name)	Both
...Q9	Regular path expressions (unknown subpaths)	Both
...Q10	Sorting by string types	Both
...Q11	Sorting by non string types	Both
...Q12	Document structure preserving	Both
...Q13	Document structure transforming	Both
...Q14	Missing elements	Both
...Q15	Empty (null) values	Both
...Q16	Retrieve individual docs	Both
...Q17	Uni-gram search	TC
...Q18	N-gram search	TC
...Q19	References and joins	Both
...Q20	Datatype Cast	Both

Table 5.1: Query List

- **TC/MD_Q4** *Find the heading of the section following the section entitled "Introduction" in articles written by (Ben Yang).*

```
for $a in input()/article/body/section,
    $b in input()/article/body/section
where $a/@heading="Introduction" and
    $a/../../prolog/authors/author/name="Ben Yang" and
    $a/position() = $b/position()-1
return
    <Output>
       {$b/@heading}
    </Output>
```

- **TC/MD_Q5** *Return the headings of the first section of articles authored by (Ben Yang).*

```
for $a in input()/article
where
    $a/prolog/authors/author/name="Ben Yang"
return
    <HeadingOfSection>
        {$a/body/section[1]/@heading}
    </HeadingOfSection>
```

- **TC/MD_Q6** *Find titles of articles where two keywords ("Life" and "Science")are mentioned in the same paragraph.*

```
for $a in input()/article
where some $b in $a//p satifies
    (contains($b, "Science") and contains($b, "Life"))
return
    $a/prolog/title
```

- **TC/MD_Q7** *Find titles of articles where a keyword ("Science") is mentioned in every paragraph.*

```
for $a in input()/article
where every $b in $a//p satisfies
    (contains($b, "Science")
return
    $a/prolog/title
```

- **TC/MD_Q8** *Return the names of all authors of the article with matching id attribute value (2).*

```
for $art in input()/article[@id="2"]
return
    $art/prolog/*/author/name
```

- **TC/MD_Q9** *Return all author names of the article with matching id attribute value (3).*

```
for $art in input()/article[@id="3"]
return
    $art//author/name
```

- **TC/MD_Q10** *List the titles of articles sorted by country, dated within a certain time period (from 1990-01-01 to 1991-01-01).*

```
for $a in input()/article/prolog
where $a/dateline/date gt "1990-01-01"
      and $a/dateline/date lt "1991-01-01"
order by $a/country
return
    <Output>
        {$a/title}
        {$a/country}
    </Output>
```

- **TC/MD_Q11** *List the titles of articles that have a matching country element type (Canada), sorted by date*

```
for $a in input()/article/prolog
where $a/dateline/country="Canada"
order by $a/dateline/date
return
    <Output>
        {$a/title}
        {$a/dateline/date}
    </Output>
```

- **TC/MD_Q12** *Retrieve the body of the article that has a matching id attribute value (4).*

```
for $a in input()/article[@id="4"]
return
    <Article>
        {$a/body}
    </Article>
```

- **TC/MD_Q13** *Construct a brief information on the article that has a matching id attribute value (5), including title, the name of first author, date and abstract.*

```
for $a in input()/article[@id="5"]
return
    <Output>
        {$a/prolog/title}
        {$a/prolog/authors/author[1]/name}
        {$a/prolog/dateline/date}
        {$a/body/abstract}
    </Output>
```

- **TC/MD_Q14** *List article title that doesn't have genre element and published within a certain time period (from 1995-01-01 to 1996-01-01).*

```
for $a in input()/article/prolog
where empty ($a/genre) and
    $a/dateline/date gt "1995-01-01" and
    $a/dateline/date lt "1996-01-01"
return
    <NoGenre>
        {$a/title}
    </NoGenre>
```

- **TC/MD_Q15** *List author names whose contact elements are empty in articles published within a certain time period (from 1995-01-01 to 1996-01-01).*

```
for $a in input()/article/prolog/authors/author
where empty($a/contact/text()) and
    $a/dateline/date gt "1995-01-01" and
    $a/dateline/date lt "1996-01-01"
return
    <NoContact>
        {$a/name}
    </NoContact>
```

- **TC/MD_Q16** *Get the article by its id attribute value (6).*

```
for $a in input()/article[@id="6"]
return
    $a
```

- **TC/MD_Q17** *Return the titles of articles which contain a certain word ("Science").*

```
for $a in input()/article
where contains ($a//p, "Science")
return
    $a/prolog/title
```

- **TC/MD_Q18** *List the titles and abstracts of articles which contain a given phrase ("Life Science").*

```
for $a in input()/article
where some $b in $a/body satisfies
    contains($b, "Life Science")
return
    <Output>
        {$a/prolog/title}
        {$a/body/abstract}
    </Output>
```

- **TC/MD_Q19** *List the names of articles cited by an article with a certain id attribute value (7).*

```
for $a in input()/article[@id='7'],
    $b in input()/article
where $a/epilog/refereces/a_id = $b/@id
return
    <Output>
        {$b/prolog/title}
    </Output>
```

5.2.2 Text-Centric Single Document

The following queries are based on an implicit (unnamed) input data set, which is "dictionary.xml".

- **TC/SD_Q1** *Return the entry that has matching headword ("the").*

```
for $ent in input()/dictionary/e
where $ent/hwg/hw="the"
return
    $ent
```

- **TC/SD_Q2** *Find the headword of the entry which has matching quotation author (Ben Yang).*

```
for $ent in input()/dictionary/e
where $ent/ss/s/qp/q/a="Ben Yang"
return
    $ent/hwg/hw
```

- **TC/SD_Q3** *Group entries by quotation location and calculate the total number entries in each group.*

```
for $a in distinct-values (input()/dictionary/e/ss/s/qp/q/loc)
let $b := input()/dictionary/e/ss/s/qp/q/[loc=$a]
return
    <Output>
        <Date>{$a/text()}</Date>
        <NumberOfEntries>{count($b)}</NumberOfEntries>
    </Output>
```

- **TC/SD_Q4** *List the headword of the previous entry of a matching headword ("you").*

```
for $ent in input()/dictionary/e,
    $prevEnt in input()/dictionary/e
where $ent/hwg/hw="you" and
    $prevEnt/position() = $ent/position()-1
return
    <Output>
        <CurrentEntry>{$ent/hwg/hw/text()}</CurrentEntry>
        <PreviousEntry>{$prevEnt/hwg/hw/text()}</PreviousEntry>
    </Output>
```

- **TC/SD_Q5** *Return the first sense of a matching headword ("that").*

```
for $a in input()/dictionary/e
where $a/hwg/hw="that"
return
    $a/ss[1]
```

- **TC/SD_Q6** *Return all the words which were quoted by a certain author (Ben Yang).*

```
for $word in input()/dictionary/e
where some $item in $word/ss
    satisfies $item//author eq "Ben Yang"
return
    $word
```

- **TC/SD_Q8** *Return Quotation Text of a word ("and").*

```
for $ent in input()/dictionary/e
where $ent/*/hw = "and"
return
    $ent/ss/s/qp/*/qt
```

- **TC/SD_Q9** *Return Quotation Text of a word ("and").*

```
for $ent in input()/dictionary/e
where $ent//hw = "and"
return
    $ent//qt
```

- **TC/SD_Q10** *List the words and their pronunciation, alphabetically, quoted by someone (Ben Yang).*

```
for $a in input()/dictionary/e
where $a/ss/s/qp/q/author="Ben Yang"
order by $a/hwg/hw
return
    <Output>
        {$a/hwg/hw}
        {$a/hwg/pr}
    </Output>
```

- **TC/SD_Q11** *List the quotation authors and quotation dates, sorted by date, for a word ("word").*

```
for $a in input()/dictionary/e
where $a/hwg/hw="word"
order by $a/ss/s/qp/q/qd
return
    <Output>
        {$a/ss/s/qp/q/a}
        {$a/ss/s/qp/q/qd}
    </Output>
```

- **TC/SD_Q12** *Retrieve the senses of a word ("his").*

```
for $a in input()/dictionary/e
where $a/hwg/hw="his"
return
    <Entry>
        {$a/ss}
    </Entry>
```

- **TC/SD_Q13** *Construct a brief information on a word ("his"), including: headword, pronunciation, part_of_speech, first etymology and first sense definition.*

```
for $a in input()/dictionary/e
where $a/hwg/hw="his"
return
    <Output>
        {$a/hwg/hw}
        {$a/hwg/pr}
        {$a/hwg/pos}
        {$a/etymology/cr[1]}
        {$a/ss/s[1]/def}
    </Output>
```

- **TC/SD_Q17** *Return the headwords of the entries which contain a certain word ("Science").*

```
for $a in input()/dictionary/e
where contains ($a, "Science")
return
    $a/hwg/hw
```

- **TC/SD_Q18** *List the headwords of entries which contain a given phrase ("Life Science").*

```
for $a in input()/dictionary
where some $b in $a/e satisfies
    contains($b, "Life Science")
return
    $a/e/hwg/hw
```

5.2.3 Data-Centric Multiple Document

The following queries are based on an implicit (unnamed) input data set, which is a collection of documents ("orderXXX.xml", "customer.xml", "item.xml", "author.xml", "address.xml" and "country.xml").

- **DC/MD_Q1** *Return the customer id of the order that has matching id attribute value (1).*

```
for $order in input()/order[@id="1"]
return
    $order/customer_id
```

- **DC/MD_Q3** *Group orders by customer id and calculate the total number of each group.*

```
for $a in distinct-values (input()/order/customer_id)
let $b:=input()/order[customer_id=$a]
return
    <Output>
        <CustKey>{$a/text()}</CustKey>
        <NumberOfOrders>{count($b)}</NumberOfOrders>
    </Output>
```

- **DC/MD_Q4** *List the order id of the previous order of a matching order on certain date (1990-01-01).*

```
for $ord in input()/order[order_date="1990-01-01"],
    $prevOrd in input()/order
where $prevOrd/position() = $ord/position()-1
return
    <Output>
        <CurrentOrder>{$ord/@id}</CurrencyOrder>
        <PreviousOrder>{$preOrd/@id}</PreviousOrder>
    </Output>
```

- **DC/MD_Q5** *Return the first order line item of a certain order with id attribute value (2).*

```
for $a in input()/order[@id="2"]
return
    $a/order_lines/order_line[1]
```

- **DC/MD_Q6** *Return invoice where some discount rates of sub-line items are higher than a certain number (0.075).*

```
for $ord in input()/order
where some $item in $ord/order_lines
satifies $item/order_line/discount_rate gt 0.075
return
    $ord
```

- **DC/MD_Q7** *Return invoice where all discount rates of sub-line items are higher than a certain number (0.075).*

```
for $ord in input()/order
where every $item in $ord/order_lines
satisfies $item/order_line/discount_rate gt 0.075
return
    $ord
```

- **DC/MD_Q8** *Return the order status of an order with an attribute value (3).*

```
for $a in input()/order[@id="3"]
return
    $a/*/order_status
```

- **DC/MD_Q9** *Return the order status of an order with id attribute value (4).*

```
for $a in input()/order[@id="4"]
return
    $a//order_status
```

- **DC/MD_Q10** *List the orders (order id, order date and ship type), sorted by ship type, ordered within a certain period of time (from 1995-01-01 to 1996-01-01).*

```
for $a in input()/order
where $a/order_date gt "1995-01-01" and
    $a/order_date lt "1996-01-01"
order by $a/ship_type
return
    <Output>
        {$a/@id}
        {$a/order_date}
        {$a/ship_type}
    </Output>
```

- **DC/MD_Q11** *List the orders (order id, order date and order total), sorted by order total, ordered in a certain period of time (from 1995-01-01 to 1996-01-01).*

```
for $a in input()/order
where $a/order_date gt "1995-01-01" and
    $a/order_date lt "1996-01-01"
order by $a/order_total
return
    <Output>
        {$a/@id}
        {$a/order_date}
        {$a/order_total}
    </Output>
```

- **DC/MD_Q12** *List all order lines of a certain order with id attribute value (5).*

```
for $a in input()/order[@id="5"]
return
    <Output>
        {$a/order_lines}
    </Output>
```

- **DC/MD_Q16** *Retrieve one whole order document with certain id attribute value (6).*

```
for $a input()/order[@id="6"]
return
    $a
```

- **DC/MD_19** *For a particular order with id attribute value (7), get its customer name and phone, and its order status.*

```
for $order in input()/order,
    $cust in document("customer.xml")/customers/customer
where $order/customer_id = $cust/@id
    and $order/@id = "7"
return
    <Output>
        {$order/order_id}
        {$order/ordre_status}
        {$cust/first_name}
        {$cust/last_name}
        {$cust/phone_number}
    </Output>
```

5.2.4 Data-Centric Single Document

The following queries are based on an implicit (unnamed) input data set, which is "catalog.xml".

- **DC/SD_Q1** *Return the item that has matching item id attribute value (1).*

```
for $item in input()/catalog/item[@id="1"]
return
    $item
```

- **DC/SD_Q2** *Find the title of the item which has matching author first name (Ben).*

```
for $item in input()/catalog/item
where $item/authors/author/name/first_name = "Ben"
return
    $item/title
```

- **DC/SD_Q3** *Group items by publisher and calculate the total number of items for each group.*

```
for $a in distinct-values (input()/catalog/item/publisher)
let $b := input()/catalog/item[publisher=$a]
return
    <Output>
        <Publisher>{$a/text()}</Publisher>
        <NumberOfItems>{count($b)}</NumberOfItems>
    </Output>
```

- **DC/SD_Q4** *List the item id of the previous item of a matching item with id attribute value (2).*

```
for $item in input()/catalog/item[@id="2"],
    $prevItem input()/catalog/item
where $prevItem/position()= $item/position()-1
return
    <Output>
        <CurrentItem>{$item/@id}</CurrencyItem>
        <PreviousItem>{$prevItem/@id}</PreviousItem>
    </Output>
```

- **DC/SD_Q5** *Return the information about the first author of item with a matching id attribute value (3).*

```
for $a input()/catalog/item[@id="3"]
return
    $a/authors/author[1]
```

- **DC/SD_Q6** *Return item information where some authors are from certain country (Canada).*

```
for $item in input()/catalog/item
where some $auth in
    $item/authors/author/contact_information/mailing_address
satifies $auth/name_of_country = "Canada"
return
    $item
```

- **DC/SD_Q7** *Return item information where all its authors are from certain country (Canada).*

```
for $item in input()/catalog/item
where every $add in
    $item/authors/author/contact_information/mailing_address
satisfies $add/name_of_country = "Canada"
return
    $item
```

- **DC/SD_Q8** *Return the publisher of an item with id attribute value (4).*

```
for $a in input()/catalog/*[@id="4"]
return
    $a/publisher
```

- **DC/SD_Q9** *Return the ISBN of an item with id attribute value (5).*

```
for $a in input()/catalog/item
where $a/@id="5"
return
    $a//ISBN/text()
```

- **DC/SD_Q10** *List the item titles sorted by publisher name with release date within a certain time period (from 1990-01-01 to 1995-01-01).*

```
for $a in input()/catalog/item
where $a/date_of_release gt "1990-01-01" and
    $a/date_of_release lt "1995-01-01"
order by $a/publisher/name
return
    <Output>
        {$a/title}
        {$a/publisher}
    </Output>
```

- **DC/SD_Q11** *List the item titles sorted by date of release with date of release within a certain time range (from 1990-01-01 to 1995-01-01.*

```
for $a in input()/catalog/item
where $a/date_of_release gt "1990-01-01" and
    $a/date_of_release lt "1995-01-01"
order by $a/data_of_release
```

```
return
    <Output>
        {$a/title}
        {$a/date_of_release}
    </Output>
```

- **DC/SD_Q12** *Get the mailing address of the first author of certain item with id attribute value (6).*

```
for $a in input()/catalog/item[$id="6"]
return
    <Output>
        {$a/authors/author[1]/contact_information/mailing_addres}
    </Output>
```

- **DC/SD_Q14** *Return the names of publishers who publish books between a period of time (from 1990-01-01 to 1991-01-01) but do not have FAX number.*

```
for $a input()/catalog/item
where $a/date_of_release gt "1990-01-01" and
           $a/date_of_release lt "1991-01-01" and
           empty($a/publisher/contact_information/FAX_number)
return
      <Output>
        {$a/publisher/name}
      </Output>
```

- **DC/SD_Q19** *Retrieve the item titles related by certain item with id attribute value (7).*

```
for $item in input()/catalog/item[@id="7"],
    $related in input()/catalog/item
where $item/related_items/related_item/item_id = $related/@id
return
    <Output>
        {$related/title}
    </Output>
```

- **DC/SD_Q20** *Retrieve the item title whose the size is bigger than certain number (1000).*

```
for $item in input()/catalog/item
let $size := $item/attribtes/size_of_book
where $size/length*$size/width*$size/height gt 1000
```

```
return
    <Output>
        {$item/title}
    </Output>
```

Chapter 6

Related Work

XML benchmarks come in two groups: application benchmarks and micro benchmarks. The Michigan Benchmark [33] developed at the University Michigan belongs to the category of micro benchmarks. It is designed to isolate problems, measure and thus, improve a particular part of a XML system, the query processing engine, for instance.

Application benchmarks, on the other hand, are proposed to measure the overall performance of an XML DBMS. XBench is an application benchmark, and so are XMach-1, XMark, and XOO7.

XMach-1 [10] is a scalable multi-user benchmark. It is based on a web application and considers text documents and catalog data. It only defines a small number of XML queries that cover multiple functions and update operations for which system performance is determined. It provides support for DTD only and does not consider XML Schema for optimization.

XMark [11] is a single-user benchmark. The database model is based on an Internet auction site, and therefore, its database contains one big XML document with text and non-text data. Compared to XMach-1, it provides a concise and comprehensive set of queries. However, it has no support for XML Schema.

XOO7 [12] was derived from OO7, which was designed to test the efficiency of object-oriented DBMS. The database model of XOO7 is mapped from the relational data model of OO7. Besides mapping the original queries of OO7 into XML, XOO7 adds some XML specific queries. It supports DTD only.

These benchmarks (XMach-1, XMark and XOO7) cover only a subset of our requirements. For example:

- They consider a DBMS containing only XML documents — no data integration.
- They only address basic features of XML — no “advanced” features like entities and namespaces.
- They ignore XML schemas and overlook DTDs — though one of them provide multiple DTDs but the structures of DTDs are similar and the same.
- They ignore or overlook update operations.

	XMach-1	XMark	XOO7	XBench
Application Domain	E-Commerce	E-Commerce	Library	Various
Users	Multi-user	Single-user	Single-user	Single-user
Document Environment	Multiple Documents	Single Document	Multiple Documents	Both
Scalability in no. of documents	From 10,000 to 10,000,000 documents	1	Unlimited	Various depending on domains
Scalability in size of documents	10KB	From 10 MB to 10 GB	Unknown	Various depending on domains
Data Heterogeneity	XML documents only	XML documents only	XML documents only	XML documents only
DB Physical Distribution	Distributed	Centralized	Centralized	Centralized
XML Schema	No	No	No	Yes
DTD	Multiple but same structure	One	One	Multiple
Advanced XML Features	Links	Links	Links	Links
Query operations	cover most of them	cover most of them	most of them	complete XQuery
Update Operations	Few	None	None	None
Multi-function Queries	Yes	Not Applicable	No	Yes

Table 6.1: Features

- Queries defined in these benchmarks do not completely cover our requirements.

A brief comparison of key features on these three XML benchmarks against XBench is depicted in Table 6.1.

Table 6.2 groups queries of each benchmark by query functionality.

Providing four different classes of databases, XBench gives end users choices to pick the application domains they can test XML DBMSs. XBench workload covers the main XQuery functionality. Furthermore, it provides opportunities for systems to optimize queries using meta-data in the form of DTDs and XML Schema specifications.

XBench workload covers the main XQuery functionality. Furthermore, it provides opportunities for systems to optimize queries using meta-data in the form of DTDs and XML Schema specifications.

Query Functionality	XMach-1	XMark	XOO7	XBench
Exact Math				
– **Shallow**	-	-	Q1	Q1
– **Deep**	Q1	Q1	-	Q2
Function Application	Q3, Q7	Q18, Q20	Q3, Q7, Q15	Q3
Ordered Access				
– **Relative**	Q3	Q2-Q3	Q2	Q4
– **Absolute**	Q3	Q4	Q17-Q18	Q5
Quantifier				
– **Existential**	Q8	-	Q14	Q6
– **Universal**	-	-	-	Q7
Regular Path Expressions				
– **Unknown Element**	Q2	Q15-Q16	-	Q8
– **Unknown Subpath**	Q4-Q5	Q6-Q7	-	Q9
Sorting				
– **By String**	Q8	Q19	-	Q10
– **By Non-string**	-	-	Q8	Q11
Document Construction				
– **Structure Preserving**	-	Q13	Q16	Q12
– **Structure Transforming**	-	Q10	Q6, Q9, Q12, Q16	Q13
Irregularity				
– **Missing Elements**	-	-	-	Q14
– **Empty (Null) Values**	-	Q17	-	Q15
Retrieve Individual Documents	Q1	-	-	Q16
Text Search				
– **Uni-gram Search**	-	Q14	-	Q17
– **N-gram Search**	Q2	-	Q5	Q18
Reference and Joins	Q1-Q2, Q6	Q8-Q9, Q11-Q12	Q10-Q11, Q15, Q17-Q18	Q19
Datatype Cast	-	Q5	-	Q20

Table 6.2: Query Functionality

Chapter 7

Conclusion

XBench is a comprehensive XML database benchmark that covers a large number of XML database applications. These applications are characterized by whether they are data-centric or text-centric and whether they consist of a single document or multiple documents. XBench workload covers the functionality of XQuery as captured in the Use Cases.

XBench is designed as an application benchmark rather than a micro-benchmark. As such it is most appropriate to use it for testing the performance and functionality of a DBMS in supporting one or more XML applications as indicated above.

This first version of XBench has a number of limitation that will be addressed in subsequent releases. There are the following:

- Support for distributed environments. The current version assumes that the DBMS and the clients are located on the same machine. Extensions to distributed settings will be incorporated.

- Support for updates and bulk loading. The current version incorporates only query workloads. Update and bulk loading workloads will be added.

- More realistic data-centric database design. The current version uses a database schema that is based on a modified version of TPC-W database. As indicated earlier, this is due to the lack of suitable data sets that lead themselves to statistical analysis. As these data sets become available, the database designs for data-centric applications will change.

Chapter 8

Acknowledgements

The Reuters Corpus is provided by Reuters Limited (UK). The Springer Digital Library data is provided by Springer-Verlag GmbH (Germany).

We thank Ning Zhang for his help in verifying the query workload.

Bibliography

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)." http://www.w3.org/TR/2000/REC-xml-20001006, October 2000. World Wide Web Consortium (W3C).

[2] "ISO 8879:1986(E). Information processing – Text and Office Systems – Standard Generalized Markup Language (SGML)," October 1986. ISO (International Organization for Standardization).

[3] World Wide Web Consortium (W3C), "HyperText Markup Language." http://www.w3.org/MarkUp/, 2001.

[4] A. Sahuguet, "KWEELT, the Making-of: Mistakes Made and Lessons Learned," technical report, Department of Computer and Information Science, University of Pennsylvania, November 2000.

[5] IBM, "IBM DB2 Universal Database XML Extender - Administration and Programming." http://www-4.ibm.com/software/data/db2/extenders/xmlext, 1999.

[6] T. Lahiri, S. Abiteboul, and J. Widom, "Ozone: Integrating Structured and Semistructured Data," in *Proceedings of the Seventh International Conference on Database Programming Languages*, (Kinloch Rannoch, Scotland), pp. 297–323, September 1999.

[7] H. Schöning and J. Wäsch, "Tamino - An Internet Database System," in *Advances in Database Technology - EDBT 2000, 6th International Conference on Extending Database Technology, Konstanz, Germany, March 27-31, 2000, Proceedings* (C. Zaniolo, P. C. Lockemann, M. H. Scholl, and T. Grust, eds.), vol. 1777 of *Lecture Notes in Computer Science*, pp. 383–387, Springer, 2000.

[8] T. Fiebig, S. Helmer, C.-C. Kanne, J. Mildenberger, G. Moerkotte, R. Schiele, and T. Westmann, "Anatomy of a Native XML Base Management System," Technical Report TR-02-001, Universität Mannheim, Fakultät für Mathematik und Informatik, D7, 27, Universität Mannheim, 68131 Mannheim, Germany, January 2002.

[9] "Publications on Xyleme," 2002. http://osage.inria.fr/verso/PUBLI/all-bykey.php?mytexte=xyleme.

[10] T. Böhme and E. Rahm, "XMach-1: A Benchmark for XML Data Management," in *Proceedings of German database conference BTW2001*, pp. 264–273, March 2001.

[11] A. R. Schmidt, F. Waas, M. L. Kersten, D. Florescu, I. Manolescu, M. J. Carey, and R. Busse, "The XML Benchmark Project," Tech. Rep. INS-R0103, CWI, Amsterdam, The Netherlands, April 2001.

[12] S. Bressan, M. L. Lee, Y. G. Li, Z. Lacroix, and U. Nambiar, "The XOO7 XML Management System Benchmark," Technical Report TR21/00, NUS CS Department, November 2001.

[13] J. Gray, *The Benchmark Handbook for Database and Transaction Processing Systems*. San Mateo, California: Morgan Kaufmann Publishers, Inc., 1991.

[14] D. Chamberlin, D. Florescu, J. Robie, J. Siméon, and M. Stefanescu, "XQuery: A Query Language for XML." http://www.w3.org/TR/xquery, February 2001. World Wide Web Consortium (W3C).

[15] I. Tatarinov, Z. Ives, A. Halevy, and D. Weld, "Updating XML," in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, (Santa Barbara, California), pp. 413–424, May 21-24 2001.

[16] D. Chamberlin, P. Fankhauser, M. Marchiori, and J. Robie, "XML Query Use Cases." http://www.w3.org/TR/xmlquery-use-cases.

[17] "The Internet Movie Database (IMDB)," 2000. http://www.imdb.com.

[18] "The GNU version of The Collaborative International Dictionary of English," 2001. The GNU, http://www.ibiblio.org/webster/.

[19] "Oxford English Dictionary (OED)," 1994. Oxford University Press, http://www.oed.com.

[20] J. Bosak, "Complete Plays of Shakespeare in XML," 1999. http://www.ibiblio.org/xml/examples/shakespeare.

[21] M. Ley, "Computer Science Billiography," 2000. http://www.informatik.uni-trier.de/ ley/db.

[22] I. W. Burr, *Applied Statistical Methods*. Academic Press, 1974.

[23] "Electronic Catalog XML (eCX)," 2002. Requisite Technology, http://www.ecx-xml.org.

[24] "Commerce XML (cXML)," 2002. cXML, http://www.cxml.org.

[25] "XMLPay," 2002. Verisign, Inc., http://www.verisign.com/developer/xml/ xml-pay.html.

[26] "XML Common Business Library (xCBL)," 2002. xCBL, http://www.xcbl.org.

[27] Transaction Processing Performance Council (TPC). http://www.tpc.org, 2001.

[28] V. Turau, "Making Legacy Data Accessible for XML Applications." http://www.informatik.fh-wiesbaden.de/ turau/veroeff.html, 1999.

[29] D. Lee, M. Mani, F. Chiu, and W. Chu, "Nesting-Based Relational-to-XML Schema Translation," in *International Workshop on the Web and Databases (WebDB)*, (Santa Barbara, CA), pp. 61–66, May 2001.

[30] D. Lee, M. Mani, F. Chiu, and W. Chu, "Effective Schema Conversions between XML and Relational Models," in *Proceedings of European Conference on Artificial Intelligence (ECAI), Knowledge Transformation Workshop*, (Lyon, France), July 2002.

[31] D. Barbosa, A. Mendelzon, J. Keenleyside, and K. Lyons, "ToXgene: A Template-Based Data Generator for XML," in *Proceedings of the Fifth International Workshop on the Web and Databases (WebDB 2002)*, (Madison, Wisconsin), pp. 49–54, ACM Press, June 6-7 2002.

[32] D. C. Fallside, "XML Schema Part 0: Primer." http://www.w3.org/TR/xmlschema-0, May 2001. World Wide Web Consortium (W3C).

[33] K. Runapongsa, J. M. Patel, H. V. Jagadish, and S. Al-Khalifa, "The Michigan Benchmark," technical report, University of Michigan, 2002. http://www.eecs.umich.edu/db/mbench/.

Appendix A

More XML Document Characterization

The parameters in addition to those in Section 4.1, that are used to characterize XML documents fall into two groups. The first group is for XML documents themselves while the second group is for meta data — i.e., DTDs and schemas. We ignore some XML features like entities since they can be parsed and resolved by XML processors before the XML documents are loaded into databases. We calculate average, minimum and maximum values on any parameter if applicable.

A.1 XML Documents

Document size. The total size of a document in bytes. The sizes of XML documents vary in different application domains. Usually, the sizes of average XML documents are approximately between 40Kb and 600Kb. However, e-commerce catalog files, for instance, can be relatively large – over 500 MB.

Number of elements. The total number of elements in a XML document basically depends on the size of the document and the type of the document (text or non text, etc.).

Number of attributes. The total number of attributes in a document.

Tree depth and fan out factor. These parameters are used to describe the tree structure of a XML document. The first one defines the levels of nesting with an element, and the second one specifies how many children a node can have — i.e., how many direct sub-elements an element contains. Given the average, minimum and maximum values of these parameters, we simulate the structures of XML documents, including skewed structures.

Tag-to-content ratio. The ratio of the tags to data content in the document. Text documents and non-text documents may have very different ratios. A text document (e.g., book) contains significantly more text than element tags, while a data-centric document (e.g., catalog) dedicates more parts to tags.

Number of links. The total number of links (such as IDREF or IDREFS) in a document includes the number of internal links and the number of external links.

Recursions. It is not unusual that the tag name of an element is the same as its direct parent or one of its ancestors, generating recursion. For example, a "chapter" element may have a "chapter" sub-element. The parameter captures the number of elements with recursions and their depths.

Number of element types. The total number of element types in a XML document should be less than the total number of elements. The fewer the number of element types, the more repetition there will be of elements, which leads to more recursions.

Values of attributes. The types of values (e.g., String, Float, and Integer) of attributes appearing in a XML document, as well as the sizes of those values. In String type, the vocabulary of the String values is also kept.

A.2 Meta Data

Number of DTDs and schemas. According to our requirements, we need to consider the situations where XML documents have DTDs and/or schemas, or none of them. Therefore, fixed numbers of DTDs and schemas will be generated to which some generated XML document conform.

Distribution of documents per DTD (or schema). One or more generated XML documents maybe associate with each DTD or schema. This parameter defines how many documents a DTD or schema can associate with and how they distribute.

A.3 Experiments

Based on above parameters, statistical data were gathered on XML documents of some applications. Unfortunately most of XML documents that were found have little or no any meta data. Therefore the meta data are not analyzed. Among these applications, Xmark is synthetically generated data, some are sample data (cXML, WCS, and OLAP Cube), and others are real data or subsets of real data.

Aggregated results (see Figures A.1 and A.2) provide a general idea of the characteristics of various XML documents in different application domains. According to these, it is obvious that text-centric (TC) documents have higher percentages of text content[1] in the documents. All the TC documents (DBLP, GCIDE, Reuters, and Shakespeare) have more than 50% text content. Among them, Text-Centric

[1] "Content", refers to not only the content of an element but also the value of an attribute. In this context, we treat elements and attributes the same.

Applications		Size	Elems	Attrs	Avg A/E	Min A/E	Max A/E	Avg Dept	Min Dept	Max Dept	Avg FanOut	Min FanOut	Max FanOut	Text	Text%	AttV	AttV%
cXML(46)	Avg	2,340.0	33.2	20.9	0.8	0.0	3.9	4.2	2.5	5.1	1.9	1.0	4.6	189.0	7.9%	192.0	10.3%
	Min	294.0	3.0	6.0	0.3	0.0	2.0	2.0	1.0	2.0	1.0	1.0	1.0	0.0	0.0%	59.0	4.3%
	Max	6,954.0	87.0	60.0	2.0	0.0	5.0	8.1	3.0	10.0	3.4	1.0	10.0	543.0	16.9%	515.0	29.8%
DBLP(4362)	Avg	581.8	15.4	3.9	0.2	0.0	1.0	1.0	1.0	1.0	14.4	14.4	14.4	271.1	47.7%	40.3	6.3%
	Min	233.0	5.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	4.0	1.0	4.0	98.0	16.0%	12.0	0.6%
	Max	5,937.0	138.0	125.0	0.9	0.0	1.0	1.2	1.0	2.0	137.0	137.0	137.0	2,273.0	67.4%	1,462.0	33.1%
WCS		1,598,165.0	11,294.0	65,107.0	5.8	1.0	27.0	1.0	1.0	1.0	11,293.0	11,293.0	11,293.0	11,294.0	0.7%	458,222.0	28.7%
GCIDE		57,917,440.0	2,267,510.0	23.0	0.0	0.0	2.0	2.4	1.0	7.0	4.0	1.0	239,185.0	33,594,747.0	58.0%	572.0	0.0%
IMDB1		4,006,587.0	155,887.0	21,960.0	0.1	0.0	1.0	2.0	2.0	2.0	8.4	3.0	18,590.0	1,143,552.0	28.5%	36,437.0	0.9%
IMDB2		11,036,866.0	283,065.0	24,881.0	0.1	0.0	1.0	3.0	2.0	4.0	5.4	1.0	11,024.0	4,446,604.0	40.3%	72,054.0	0.7%
OLAPCube		63,618.0	662.0	3,805.0	5.7	0.0	8.0	2.1	2.0	4.0	55.1	1.0	613.0	2,488.0	3.9%	14,557.0	22.9%
Reuters (1952)	Avg	2,967.1	38.6	40.7	1.1	0.0	4.0	2.2	1.0	4.0	3.6	1.0	14.1	1,557.0	47.5%	480.8	18.1%
	Min	1,146.0	20.0	23.0	0.2	0.0	4.0	1.9	1.0	4.0	2.2	1.0	7.0	223.0	17.8%	263.0	2.5%
	Max	11,214.0	229.0	161.0	1.5	0.0	4.0	2.9	1.0	4.0	20.9	1.0	200.0	10,005.0	89.2%	1,966.0	30.2%
Shakspeare (37)	Avg	213,448.5	4,856.5	0.0	0.0	0.0	0.0	3.9	1.0	5.0	5.6	1.0	155.3	126,893.6	59.5%	0.0	0.0%
	Min	141,345.0	3,153.0	0.0	0.0	0.0	0.0	3.9	1.0	4.0	4.6	1.0	71.0	84,458.0	55.8%	0.0	0.0%
	Max	288,735.0	6,636.0	0.0	0.0	0.0	0.0	4.0	1.0	5.0	7.0	2.0	434.0	170,648.0	64.2%	0.0	0.0%
Xmark		116,524,435.0	1,666,315.0	381,878.0	0.2	0.0	2.0	4.6	2.0	11.0	3.7	1.0	25,500.0	81,286,567.0	69.8%	4,284,980.0	3.7%

Figure A.1: Statistics of Parameters of Some Applications

Applications		T+V	(T+V)%	Links	Recu#	Avg Rec	min Rec	max Rec	Elem Type	Avg E/Type	Min E/T	Max E/Type	Avg VLe	Min VLe	Max VLe	Avg TLen	Min TLe	Max TLen
cXML(46)	Avg	381.0	18.2%	0.0	0.0	0.0	0.0	0.0	21.5	1.4	1.0	3.5	10.5	2.2	28.2	5.5	0.3	35.1
	Min	63.0	9.3%	0.0	0.0	0.0	0.0	0.0	3.0	1.0	1.0	1.0	5.3	1.0	18.0	0.0	0.0	9.0
	Max	932.0	41.3%	0.0	1.0	1.0	1.0	1.0	48.0	2.1	1.0	9.0	17.3	5.0	49.0	11.5	1.0	105.0
DBLP(4362)	Avg	311.4	54.0%	0.0	0.0	0.0	0.0	0.0	9.8	1.5	1.0	6.3	20.4	20.1	22.3	18.2	1.0	61.3
	Min	115.0	37.4%	0.0	0.0	0.0	0.0	0.0	5.0	1.0	1.0	1.0	2.5	1.0	12.0	7.1	1.0	31.0
	Max	3,031.0	71.3%	0.0	0.0	0.0	0.0	0.0	12.0	12.5	1.0	124.0	37.0	37.0	42.0	41.6	1.0	168.0
WCS		469,516.0	29.4%	0.0	0.0	0.0	0.0	0.0	55.0	205.3	1.0	3,000.0	7.0	1.0	51.0	1.0	0.0	1.0
GCIDE		33,595,319.0	58.0%	0.0	0.0	0.0	0.0	0.0	283.0	8,012.4	1.0	252,136.0	24.9	1.0	69.0	14.8	0.0	2,485.0
IMDB1		1,179,989.0	29.5%	0.0	0.0	0.0	0.0	0.0	8.0	19,485.9	1.0	40,783.0	1.7	1.0	5.0	7.3	0.0	105.0
IMDB2		4,518,658.0	40.9%	0.0	0.0	0.0	0.0	0.0	23.0	12,307.2	1.0	87,073.0	2.9	1.0	11.0	15.7	0.0	4,305.0
OLAPCube		17,045.0	26.8%	0.0	0.0	0.0	0.0	0.0	13.0	50.9	1.0	613.0	3.8	1.0	44.0	3.8	0.0	17.0
Reuters (1952)	Avg	2,037.8	65.6%	0.0	0.0	0.0	0.0	0.0	12.0	3.2	1.0	13.1	11.8	2.0	32.0	38.8	0.0	216.3
	Min	542.0	47.1%	0.0	0.0	0.0	0.0	0.0	11.0	1.7	1.0	5.0	11.3	2.0	32.0	9.2	0.0	40.0
	Max	10,281.0	91.7%	0.0	0.0	0.0	0.0	0.0	13.0	19.1	1.0	200.0	12.5	2.0	32.0	140.9	0.0	958.0
Shakspeare (37)	Avg	126,893.6	59.5%	0.0	0.0	0.0	0.0	0.0	16.4	297.0	1.0	2,914.4	0.0	0.0	0.0	26.2	0.5	239.2
	Min	84,458.0	55.8%	0.0	0.0	0.0	0.0	0.0	14.0	197.1	1.0	1,786.0	0.0	0.0	0.0	23.0	0.0	157.0
	Max	170,648.0	64.2%	0.0	0.0	0.0	0.0	0.0	19.0	442.4	1.0	4,014.0	0.0	0.0	0.0	30.9	1.0	1,010.0
Xmark		85,571,547.0	73.4%	0.0	93,312.0	1.0	1.0	1.0	74.0	22,517.8	1.0	105,114.0	11.2	3.0	17.0	48.8	0.0	2,403.0

Figure A.2: Statistics of Parameters of Some Applications - Cont'd

Single Documents (TC/SDs), are deeper (in terms of nesting) than those of other Text-Centric Multiple Documents (TC/MDs).

On the contrary, the text content of Data-Centric (DC) documents is much lower. In this case, the percentage of text content is less than 30%, and particularly, in Data-Centric Multiple Documents (DC/MDs), such as cXML, it is even lower than 20%.

The documents that were analyzed do not appear to have many cases of recursions and links. Some documents such as DBLP and IMDB do not have obvious features of the classes into which we originally placed them[2]. This is because some XML documents have features of different classes. Surprisingly, the text content of Xmark documents is extremely high, given that Xmark tries to simulate an e-commerce situation.

[2]For example, IMDB was originally classified as Data-Centric Single Document (DC/SD), but it's deep nesting feature of DC/SD is not obvious.

Appendix B

Database Generation Templates

B.1 TC/SD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY entry_num "777">
    <!ENTITY author_num "10000">
    <!ENTITY hw_1_odd "75">
    <!ENTITY hw_2_odd "20">
    <!ENTITY hw_3_odd "3">
    <!ENTITY hw_4_odd "1.5">
    <!ENTITY hw_5_odd "0.5">
    <!ENTITY pr_has_odd "81">
    <!ENTITY pr_no_odd "19">
    <!ENTITY pr_val_dis_min "2">
    <!ENTITY pr_val_dis_max "24">
    <!ENTITY pr_val_dis_mean "2.39">
    <!ENTITY pr_val_dis_var "0.34">
    <!ENTITY ps_n_odd "27">
    <!ENTITY ps_v_odd "25">
    <!ENTITY ps_adj_odd "20">
    <!ENTITY ps_adv_odd "15">
    <!ENTITY ps_prep_odd "10">
    <!ENTITY ps_conj_odd "0.95">
    <!ENTITY ps_int_odd "0.05">
    <!ENTITY ps_others_odd "2">
    <!ENTITY vfl_has_odd "25">
    <!ENTITY vfl_no_odd "75">
    <!ENTITY vfl_num_dis_min "1">
    <!ENTITY vfl_num_dis_max "9">
    <!ENTITY vfl_num_dis_mean "1.48">
    <!ENTITY vfl_num_dis_var "0.98">
```

```
<!ENTITY vf_1_odd "70">
<!ENTITY vf_2_odd "27">
<!ENTITY vf_3_odd "2">
<!ENTITY vf_4_odd "1">
<!ENTITY vd_has_odd "30">
<!ENTITY vd_no_odd "70">
<!ENTITY cf_num_dis_min "1">
<!ENTITY cf_num_dis_max "16">
<!ENTITY cf_num_dis_mean "1.13">
<!ENTITY cf_num_dis_var "0.47">
<!ENTITY et_has_odd "68">
<!ENTITY et_no_odd "32">
<!ENTITY sense_num_dis_min "1">
<!ENTITY sense_num_dis_max "10">
<!ENTITY sense_num_dis_mean "1.29">
<!ENTITY sense_num_dis_var "0.88">
<!ENTITY def_val_dis_min "1">
<!ENTITY def_val_dis_max "459">
<!ENTITY def_val_dis_mean "4.35">
<!ENTITY def_val_dis_var "0.73">
<!ENTITY def_cr_0_odd "67.5">
<!ENTITY def_cr_1_odd "22.5">
<!ENTITY def_cr_2_odd "8">
<!ENTITY def_cr_3_odd "1.5">
<!ENTITY def_cr_4_odd "0.5">
<!ENTITY q_num_dis_min "1">
<!ENTITY q_num_dis_max "20">
<!ENTITY q_num_dis_mean "3.68">
<!ENTITY q_num_dis_var "2.56">
<!ENTITY a_has_odd "55">
<!ENTITY a_no_odd "45">
<!ENTITY bib_has_odd "8">
<!ENTITY bib_no_odd "92">
<!ENTITY w_val_dis_min "2">
<!ENTITY w_val_dis_max "60">
<!ENTITY w_val_dis_mean "2.55">
<!ENTITY w_val_dis_var "0.51">
<!ENTITY lc_val_dis_min "1">
<!ENTITY lc_val_dis_max "58">
<!ENTITY lc_val_dis_mean "2.48">
<!ENTITY lc_val_dis_var "0.4">
<!ENTITY bib_p1_odd "87">
<!ENTITY bib_p2_odd "13">
<!ENTITY bib_2_odd "52">
<!ENTITY bib_3_odd "48">
```

```
    <!ENTITY bib_p2_val_min "8">
    <!ENTITY bib_p2_val_max "27">
    <!ENTITY qt_cr_0_odd "94">
    <!ENTITY qt_cr_1_odd "5">
    <!ENTITY qt_cr_2_odd "1">
    <!ENTITY qt_ib_0_odd "85">
    <!ENTITY qt_ib_1_odd "11">
    <!ENTITY qt_ib_2_odd "2.5">
    <!ENTITY qt_ib_3_odd "1">
    <!ENTITY qt_ib_4_odd "0.5">
    <!ENTITY ib_val_dis_min "1">
    <!ENTITY ib_val_dis_max "52">
    <!ENTITY ib_val_dis_mean "2.14">
    <!ENTITY ib_val_dis_var "0.59">
    <!ENTITY qt_val_dis_min "4">
    <!ENTITY qt_val_dis_max "507">
    <!ENTITY qt_val_dis_mean "4.54">
    <!ENTITY qt_val_dis_var "0.51">
]>
<tox-template>
<!-- distribution list -->
<tox-distribution name="pr_val_dis" minInclusive="&pr_val_dis_min;"
    maxInclusive="&pr_val_dis_max;" mean="&pr_val_dis_mean;"
    variance="&pr_val_dis_var;" type="lognormal"/>
<tox-distribution name="vfl_num_dis" minInclusive="&vfl_num_dis_min;"
    maxInclusive="&vfl_num_dis_max;" mean="&vfl_num_dis_mean;"
    variance="&vfl_num_dis_var;" type="normal"/>
<tox-distribution name="cf_num_dis" minInclusive="&cf_num_dis_min;"
    maxInclusive="&cf_num_dis_max;" mean="&cf_num_dis_mean;"
    variance="&cf_num_dis_var;" type="lognormal"/>
<tox-distribution name="sense_num_dis" minInclusive="&sense_num_dis_min;"
    maxInclusive="&sense_num_dis_max;" mean="&sense_num_dis_mean;"
    variance="&sense_num_dis_var;" type="normal"/>
<tox-distribution name="def_val_dis" minInclusive="&def_val_dis_min;"
    maxInclusive="&def_val_dis_max;" mean="&def_val_dis_mean;"
    variance="&def_val_dis_var;" type="lognormal"/>
<tox-distribution name="q_num_dis" minInclusive="&q_num_dis_min;"
    maxInclusive="&q_num_dis_max;" mean="&q_num_dis_mean;"
    variance="&q_num_dis_var;" type="normal"/>
<tox-distribution name="w_val_dis" minInclusive="&w_val_dis_min;"
    maxInclusive="&w_val_dis_max;" mean="&w_val_dis_mean;"
    variance="&w_val_dis_var;" type="lognormal"/>
<tox-distribution name="lc_val_dis" minInclusive="&lc_val_dis_min;"
    maxInclusive="&lc_val_dis_max;" mean="&lc_val_dis_mean;"
    variance="&lc_val_dis_var;" type="lognormal"/>
```

```
<tox-distribution name="ib_val_dis" minInclusive="&ib_val_dis_min;"
    maxInclusive="&ib_val_dis_max;" mean="&ib_val_dis_mean;"
    variance="&ib_val_dis_var;" type="lognormal"/>
<tox-distribution name="qt_val_dis" minInclusive="&qt_val_dis_min;"
    maxInclusive="&qt_val_dis_max;" mean="&qt_val_dis_mean;"
    variance="&qt_val_dis_var;" type="lognormal"/>
<!-- list -->
<tox-list name="cents0">
<element name="cent" minOccurs="400" maxOccurs="400">
<complexType>
    <element name="cent_start">
        <simpleType>
            <restriction base="int">
                <tox-number maxInclusive="20" minInclusive="0"/>
            </restriction>
        </simpleType>
    </element>
    <element name="cent_last">
        <simpleType>
            <restriction base="int">
                <tox-number minInclusive="0" maxInclusive="20"/>
            </restriction>
        </simpleType>
    </element>
</complexType>
</element>
</tox-list>
<tox-list name="cents">
<element name="cent" minOccurs="400" maxOccurs="400">
    <complexType>
        <tox-sample path="[cents0/cent]">
            <element name="cent_start" type="int">
                <tox-expr value="[cent_start]"/>
            </element>
            <element name="cent_end" type="int">
                <tox-expr value="[cent_start]+[cent_last]"/>
            </element>
        </tox-sample>
    </complexType>
</element>
</tox-list>
<tox-list name="entries">
<element name="entry" minOccurs="&entry_num;" maxOccurs="&entry_num;">
    <complexType>
        <element name="id">
```

```
            <simpleType>
                <restriction base="positiveInteger">
                    <maxInclusive value="&entry_num;"/>
                    <tox-number sequential="yes"/>
                </restriction>
            </simpleType>
        </element>
    </complexType>
</element>
</tox-list>
<tox-list name="words" readFrom="input/words.xml">
    <element name="word" type="string"/>
</tox-list>
<tox-list name="firstnames" readFrom="input/firstnames.xml">
    <element name="name" type="string"/>
</tox-list>
<tox-list name="lastnames" readFrom="input/lastnames.xml">
    <element name="name" type="string"/>
</tox-list>
<tox-list name="author_list">
<element name="author" minOccurs="&author_num;" maxOccurs="&author_num;">
    <complexType>
        <tox-sample path="[firstnames/name]">
            <element name="fname">
                <tox-expr value="[!]"/>
            </element>
        </tox-sample>
        <tox-sample path="[lastnames/name]">
            <element name="lname">
                <tox-expr value="[!]"/>
            </element>
        </tox-sample>
    </complexType>
</element>
</tox-list>
<!-- type list -->
<complexType name="posType">
    <tox-alternatives>
        <tox-option odds="&ps_n_odd;">
            <element name="pos">
                <simpleType>
                    <restriction base="string">
                        <tox-value>n.</tox-value>
                    </restriction>
                </simpleType>
```

```
    </element>
</tox-option>
<tox-option odds="&ps_v_odd;">
    <element name="pos">
        <simpleType>
            <restriction base="string">
                <tox-value>v.</tox-value>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="&ps_adj_odd;">
    <element name="pos">
        <simpleType>
            <restriction base="string">
                <tox-value>adj.</tox-value>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="&ps_adv_odd;">
    <element name="pos">
        <simpleType>
            <restriction base="string">
                <tox-value>adv.</tox-value>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="&ps_prep_odd;">
    <element name="pos">
        <simpleType>
            <restriction base="string">
                <tox-value>prep.</tox-value>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="&ps_conj_odd;">
    <element name="pos">
        <simpleType>
            <restriction base="string">
                <tox-value>conj.</tox-value>
            </restriction>
        </simpleType>
```

```
            </element>
        </tox-option>
        <tox-option odds="&ps_int_odd;">
            <element name="pos">
                <simpleType>
                    <restriction base="string">
                        <tox-value>int.</tox-value>
                    </restriction>
                </simpleType>
            </element>
        </tox-option>
        <tox-option odds="&ps_others_odd;"/>
    </tox-alternatives>
</complexType>
<complexType name="hwprType">
    <element name="hw">
        <simpleType>
            <restriction base="string">
                <tox-sample path="[words/word]">
                    <tox-expr value="[!]"/>
                </tox-sample>
            </restriction>
        </simpleType>
    </element>
    <tox-alternatives>
        <tox-option odds="&pr_has_odd;">
            <element name="pr">
                <simpleType>
                    <restriction base="string">
                        <tox-string type="gibberish"
                        tox-distribution="pr_val_dis"/>
                    </restriction>
                </simpleType>
            </element>
        </tox-option>
        <tox-option odds="&pr_no_odd;"/>
    </tox-alternatives>
</complexType>
<complexType name="headword_group_type">
<tox-alternatives>
<tox-option odds="&hw_1_odd;">
    <element name="hwpr" type="hwprType" tox-omitTag="yes"
        minOccurs="1" maxOccurs="1"/>
</tox-option>
<tox-option odds="&hw_2_odd;">
```

```
    <element name="hwpr" type="hwprType" tox-omitTag="yes"
        minOccurs="2" maxOccurs="2"/>
</tox-option>
<tox-option odds="&hw_3_odd;">
    <element name="hwpr" type="hwprType" tox-omitTag="yes"
        minOccurs="2" maxOccurs="3"/>
</tox-option>
<tox-option odds="&hw_4_odd;">
    <element name="hwpr" type="hwprType" tox-omitTag="yes"
        minOccurs="4" maxOccurs="4"/>
</tox-option>
<tox-option odds="&hw_5_odd;">
    <element name="hwpr" type="hwprType" tox-omitTag="yes"
        minOccurs="5" maxOccurs="5"/>
</tox-option>
</tox-alternatives>
<element name="pos" type="posType" tox-omitTag="yes"/>
</complexType>
<complexType name="vfType">
    <element name="vf">
        <simpleType>
            <restriction base="string">
                <tox-sample path="[words/word]">
                    <tox-expr value="[!]"/>
                </tox-sample>
            </restriction>
        </simpleType>
    </element>
</complexType>
<complexType name="variant_form_list_type">
<element name="vfl" tox-omitTag="yes" tox-distribution="vfl_num_dis"
    maxOccurs="unbounded" minOccurs="0">
<complexType>
<tox-sample path="[cents/cent]">
    <element name="vd_dummy" tox-omitTag="yes">
    <complexType>
    <tox-alternatives>
    <tox-option odds="&vd_has_odd;">
    <tox-if expr="GT([cent_end],20)">
        <tox-then>
            <element name="vd" type="int">
                <tox-expr value="[cent_start]#'-20'"/>
            </element>
        </tox-then>
        <tox-else>
```

```
            <element name="vd" type="int">
                <tox-expr value="[cent_start]#'-'#[cent_end]"/>
            </element>
        </tox-else>
    </tox-if>
    </tox-option>
    <tox-option odds="&vd_no_odd;"/>
    </tox-alternatives>
    </complexType>
    </element>
</tox-sample>
<tox-alternatives>
<tox-option odds="&vf_1_odd;">
    <element name="vf" tox-omitTag="yes" type="vfType"
        minOccurs="1" maxOccurs="1"/>
</tox-option>
<tox-option odds="&vf_2_odd;">
    <element name="vf" tox-omitTag="yes" type="vfType"
        minOccurs="2" maxOccurs="2"/>
</tox-option>
<tox-option odds="&vf_3_odd;">
    <element name="vf" tox-omitTag="yes" type="vfType"
        minOccurs="3" maxOccurs="3"/>
</tox-option>
<tox-option odds="&vf_4_odd;">
    <element name="vf" tox-omitTag="yes" type="vfType"
        minOccurs="4" maxOccurs="4"/>
</tox-option>
</tox-alternatives>
</complexType>
</element>
</complexType>
<complexType name="id_ref">
    <tox-sample path="[entries/entry]" duplicates="no">
        <tox-expr value="'E'#[id]"/>
    </tox-sample>
</complexType>
<simpleType name="qt_text_type">
    <restriction base="string">
        <tox-string type="text" minLength="0" maxLength="unbounded"
            tox-distribution="qt_val_dis"/>
    </restriction>
</simpleType>
<complexType name="cr_text_type">
    <element name="cr" type="id_ref"/>
```

```
    <element name="qt_text" tox-omitTag="yes" type="qt_text_type"/>
</complexType>
<simpleType name="ib_text_type">
    <restriction base="string">
        <tox-string type="text" minLength="0" maxLength="unbounded"
            tox-distribution="ib_val_dis"/>
    </restriction>
</simpleType>
<complexType name="ib_qt_text_type">
    <tox-alternatives>
        <tox-option odds="50">
            <element name="i" type="ib_text_type"/>
        </tox-option>
        <tox-option odds="50">
            <element name="b" type="ib_text_type"/>
        </tox-option>
    </tox-alternatives>
    <element name="qt_text" tox-omitTag="yes" type="qt_text_type"/>
</complexType>
<complexType name="etymology_type">
    <element name="cr" tox-distribution="cf_num_dis" minOccurs="0"
        maxOccurs="unbounded" type="id_ref">
    </element>
</complexType>
<complexType name="def_text_Type">
<element name="def_text" tox-omitTag="yes">
<simpleType>
    <restriction base="string">
        <tox-string type="text" minLength="0" maxLength="unbounded"
            tox-distribution="def_val_dis"/>
    </restriction>
</simpleType>
</element>
</complexType>
<complexType name="senses_type">
<element name="s" tox-distribution="sense_num_dis" minOccurs="0"
    maxOccurs="unbounded">
<complexType>
    <element name="def">
    <complexType mixed="true">
    <tox-alternatives>
    <tox-option odds="&def_cr_0_odd;">
        <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    </tox-option>
    <tox-option odds="&def_cr_1_odd;">
```

```
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
</tox-option>
<tox-option odds="&def_cr_2_odd;">
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
</tox-option>
<tox-option odds="&def_cr_3_odd;">
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
</tox-option>
<tox-option odds="&def_cr_4_odd;">
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
    <element name="cr" type="id_ref"/>
    <element name="text" type="def_text_Type" tox-omitTag="yes"/>
</tox-option>
</tox-alternatives>
</complexType>
</element>
<element name="qp">
<complexType>
<element name="q" tox-distribution="q_num_dis" minOccurs="0"
    maxOccurs="unbounded">
<complexType>
<element name="qd">
    <simpleType>
        <restriction base="int">
            <tox-number minInclusive="0" maxInclusive="2002"/>
        </restriction>
    </simpleType>
```

```
</element>
<tox-alternatives>
    <tox-option odds="&a_has_odd;">
        <element name="a">
            <simpleType>
                <restriction base="string">
                    <tox-sample path="[author_list/author]">
                        <tox-expr value="[fname]#' '#[lname]"/>
                    </tox-sample>
                </restriction>
            </simpleType>
        </element>
    </tox-option>
    <tox-option odds="&a_no_odd;"/>
</tox-alternatives>
<element name="w">
    <simpleType>
        <restriction base="string">
            <tox-string type="text" tox-distribution="w_val_dis"
                minLength="0" maxLength="unbounded"/>
        </restriction>
    </simpleType>
</element>
<tox-alternatives>
<tox-option odds="&bib_has_odd;">
<tox-alternatives>
<tox-option odds="&bib_p1_odd;">
<tox-alternatives>
<tox-option odds="&bib_2_odd;">
    <element name="bib">
    <simpleType>
        <restriction base="string">
            <tox-string type="text" minLength="2" maxLength="2"/>
        </restriction>
    </simpleType>
    </element>
</tox-option>
<tox-option odds="&bib_3_odd;">
    <element name="bib">
    <simpleType>
        <restriction base="string">
            <tox-string type="text" minLength="3" maxLength="3"/>
        </restriction>
    </simpleType>
    </element>
```

```
</tox-option>
</tox-alternatives>
</tox-option>
<tox-option odds="&bib_p2_odd;">
    <element name="bib">
    <simpleType>
        <restriction base="string">
            <tox-string type="text" minLength="&bib_p2_val_min;"
                maxLength="&bib_p2_val_max;"/>
        </restriction>
    </simpleType>
    </element>
</tox-option>
</tox-alternatives>
</tox-option>
<tox-option odds="&bib_no_odd;"/>
</tox-alternatives>
<element name="loc">
<simpleType>
    <restriction base="string">
        <tox-string type="gibberish" minLength="0"
            maxLength="unbounded" tox-distribution="lc_val_dis"/>
    </restriction>
</simpleType>
</element>
<element name="qt">
<complexType>
    <element name="qt_text" tox-omitTag="yes" type="qt_text_type"/>
    <tox-alternatives>
    <tox-option odds="&qt_cr_0_odd;"/>
    <tox-option odds="&qt_cr_1_odd;">
        <element name="cr_text" tox-omitTag="yes"
            type="cr_text_type"/>
    </tox-option>
    <tox-option odds="&qt_cr_2_odd;">
        <element name="cr_text" tox-omitTag="yes" minOccurs="2"
            maxOccurs="2" type="cr_text_type"/>
    </tox-option>
    </tox-alternatives>
    <tox-alternatives>
    <tox-option odds="&qt_ib_0_odd;"/>
    <tox-option odds="&qt_ib_1_odd;">
        <element name="ib_qt_text" tox-omitTag="yes"
            type="ib_qt_text_type"/>
    </tox-option>
```

```
        <tox-option odds="&qt_ib_2_odd;">
            <element name="ib_qt_text" tox-omitTag="yes"
                minOccurs="2" maxOccurs="2" type="ib_qt_text_type"/>
        </tox-option>
        <tox-option odds="&qt_ib_3_odd;">
            <element name="ib_qt_text" tox-omitTag="yes"
                minOccurs="3" maxOccurs="3" type="ib_qt_text_type"/>
        </tox-option>
        <tox-option odds="&qt_ib_4_odd;">
            <element name="ib_qt_text" tox-omitTag="yes" minOccurs="4"
                maxOccurs="4" type="ib_qt_text_type"/>
        </tox-option>
        </tox-alternatives>
    </complexType>
    </element>
    </complexType>
    </element>
    </complexType>
    </element>
</complexType>
</element>
</complexType>
<!-- document -->
<tox-document name="output/dictionary">
    <element name="dictionary">
    <complexType>
    <element name="e" minOccurs="&entry_num;" maxOccurs="&entry_num;">
    <complexType>
        <tox-scan path="[entries/entry]" name="entry">
            <attribute name="id" type="ID">
                <tox-expr value="'E'#[id]"/>
            </attribute>
        </tox-scan>
        <element name="hwg" type="headword_group_type"/>
        <tox-alternatives>
            <tox-option odds="&vfl_has_odd;">
                <element name="vfl" type="variant_form_list_type"/>
            </tox-option>
            <tox-option odds="&vfl_no_odd;"/>
        </tox-alternatives>
        <tox-alternatives>
            <tox-option odds="&et_has_odd;">
                <element name="et" type="etymology_type"/>
            </tox-option>
            <tox-option odds="&et_no_odd;"/>
```

```
        </tox-alternatives>
        <element name="ss" type="senses_type"/>
    </complexType>
    </element>
    </complexType>
    </element>
</tox-document>
</tox-template>
```

B.2 TC/MD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY article_num "100">
    <!ENTITY name_email_num "10000">
]> <tox-template>
    <!-- distributions -->
    <tox-distribution name="title_val_dis" type="lognormal"
    minInclusive="3" maxInclusive="252" mean="3.95" variance="0.33"/>
    <tox-distribution name="author_num_dis" type="lognormal"
    minInclusive="1" maxInclusive="48" mean="1.05" variance="0.68"/>
    <tox-distribution name="keyword_num_dis" type="lognormal"
    minInclusive="1" maxInclusive="19" mean="1.87" variance="0.22"/>
    <tox-distribution name="keyword_val_dis" type="lognormal"
    minInclusive="1" maxInclusive="96" mean="2.95" variance="0.38"/>
    <tox-distribution name="chapter_num_dis" type="normal"
    minInclusive="1" maxInclusive="15" mean="4.19" variance="1.29"/>
    <tox-distribution name="head_val_dis" type="normal"
    minInclusive="1" maxInclusive="200" mean="24.96" variance="24.18"/>
    <tox-distribution name="chapter_p_num_dis" type="lognormal"
    minInclusive="1" maxInclusive="29" mean="1.28" variance="0.67"/>
    <tox-distribution name="section_p_num_dis" type="normal"
    minInclusive="1" maxInclusive="46" mean="1.93" variance="1.49"/>
    <tox-distribution name="section_num_dis" type="lognormal"
    minInclusive="1" maxInclusive="23" mean="1.58" variance="0.37"/>
    <tox-distribution name="p_val_dis1" type="normal"
    minInclusive="1" maxInclusive="64" mean="37.64" variance="15.60"/>
    <tox-distribution name="p_val_dis2" type="lognormal"
    minInclusive="5" maxInclusive="20000" mean="5.41" variance="1.19"/>
    <tox-distribution name="subsec_p_num_dis" type="normal"
    minInclusive="1" maxInclusive="11" mean="1.76" variance="1.45"/>
    <tox-distribution name="subsec_num_dis" type="lognormal"
    minInclusive="0" maxInclusive="31" mean="0.89" variance="0.49"/>
    <tox-distribution name="subsubsec_p_num_dis" type="exponential"
    minInclusive="0" maxInclusive="12" mean="0.49"/>
    <tox-distribution name="subsubsec_num_dis" type="lognormal"
    minInclusive="0" maxInclusive="8" mean="0.87" variance="0.43"/>
    <tox-distribution name="pa_val_dis" type="lognormal"
    minInclusive="5" maxInclusive="3000" mean="5.66" variance="0.60"/>
    <tox-distribution name="citation_num_dis" type="lognormal"
    minInclusive="1" maxInclusive="&article_num;" mean="3.74"
    variance="0.44"/>
    <tox-distribution name="citation_val_dis" type="lognormal"
    minInclusive="5" maxInclusive="600" mean="5.77" variance="0.17"/>
```

```
<!-- prolog_type -->
<tox-list name="firstnames" readFrom="input/firstnames.xml">
    <element name="name" type="string"/>
</tox-list>
<tox-list name="lastnames" readFrom="input/lastnames.xml">
    <element name="name" type="string"/>
</tox-list>
<tox-list name="emails" readFrom="input/emails.xml">
    <element name="email" type="string"/>
</tox-list>
<tox-list name="cities" readFrom="input/cities.xml">
    <element name="city" type="string"/>
</tox-list>
<tox-list name="countries" readFrom="input/countries.xml">
    <element name="country" type="string"/>
</tox-list>
<tox-list name="genre_list" unique="genre/name"
readFrom="input/genres.xml">
    <element name="genre">
        <complexType>
            <element name="name" type="string"/>
        </complexType>
    </element>
</tox-list>
<tox-list name="name_email_list">
    <element name="name_email" minOccurs="&name_email_num;"
    maxOccurs="&name_email_num;">
        <complexType>
            <element name="l_name">
                <simpleType>
                    <restriction base="string">
                        <tox-sample path="[lastnames/name]">
                            <tox-expr value="[!]"/>
                        </tox-sample>
                    </restriction>
                </simpleType>
            </element>
            <element name="email">
                <simpleType>
                    <restriction base="string">
                        <tox-sample path="[emails/email]">
                            <tox-expr value="[!]"/>
                        </tox-sample>
                    </restriction>
                </simpleType>
```

```
            </element>
        </complexType>
    </element>
</tox-list>
<complexType name="author_type">
<tox-sample path="[name_email_list/name_email]" name="ne">
    <element name="name">
    <simpleType>
        <restriction base="string">
            <tox-sample path="[firstnames/name]">
                <tox-expr value="[!]#' '#[$ne/l_name]"/>
            </tox-sample>
        </restriction>
    </simpleType>
    </element>
    <element name="contact">
    <complexType>
        <tox-alternatives>
        <tox-option odds="92">
            <element name="email">
                <complexType>
                    <tox-expr value="[l_name]#'@'#[email]"/>
                </complexType>
            </element>
        </tox-option>
        <tox-option odds="8"/>
        </tox-alternatives>
        <tox-alternatives>
        <tox-option odds="69">
            <element name="phone">
                <complexType mixed="true">
                    <tox-number minInclusive="1"
                    maxInclusive="99" format="'+'0"/>
                    <tox-number minInclusive="10"
                    maxInclusive="999" format="' ('0') '"/>
                    <tox-number minInclusive="123456"
                    maxInclusive="98765432"/>
                </complexType>
            </element>
        </tox-option>
        <tox-option odds="31"/>
        </tox-alternatives>
    </complexType>
    </element>
</tox-sample>
```

```
</complexType>
<complexType name="prolog_type">
    <element name="title">
        <simpleType>
            <restriction base="string">
                <tox-string type="text"
                tox-distribution="title_val_dis"/>
            </restriction>
        </simpleType>
    </element>
    <tox-alternatives>
    <tox-option odds="6"/>
    <tox-option odds="94">
        <element name="authors">
        <complexType>
            <element name="author" type="author_type"
            tox-distribution="author_num_dis" minOccurs="1"
            maxOccurs="24"/>
        </complexType>
        </element>
    </tox-option>
    </tox-alternatives>
    <tox-alternatives>
    <tox-option odds="94">
        <element name="dateline">
            <complexType>
                <element name="city">
                    <simpleType>
                    <restriction base="string">
                        <tox-sample path="[cities/city]">
                            <tox-expr value="[!]"/>
                        </tox-sample>
                    </restriction>
                    </simpleType>
                </element>
                <element name="country">
                    <simpleType>
                    <restriction base="string">
                        <tox-sample path="[countries/country]">
                            <tox-expr value="[!]"/>
                        </tox-sample>
                    </restriction>
                    </simpleType>
                </element>
                <element name="date">
```

```
                    <simpleType>
                    <restriction base="date">
                        <tox-date start-date="1980-01-01"
                        end-date="2000-12-31"/>
                    </restriction>
                    </simpleType>
                </element>
            </complexType>
        </element>
    </tox-option>
    <tox-option odds="6"/>
    </tox-alternatives>
    <tox-alternatives>
        <tox-option odds="55">
            <element name="genre">
                <simpleType>
                    <restriction base="string">
                        <tox-sample path="[genre_list/genre]">
                            <tox-expr value="[name]"/>
                        </tox-sample>
                    </restriction>
                </simpleType>
            </element>
        </tox-option>
        <tox-option odds="45"/>
    </tox-alternatives>
    <tox-alternatives>
    <tox-option odds="78">
        <element name="keywords">
        <complexType>
        <element name="keyword" tox-distribution="keyword_num_dis"
        minOccurs="1" maxOccurs="unbounded">
            <simpleType>
            <restriction base="string">
                <tox-string type="text"
                tox-distribution="keyword_val_dis"/>
            </restriction>
            </simpleType>
        </element>
        </complexType>
        </element>
    </tox-option>
    <tox-option odds="22"/>
    </tox-alternatives>
</complexType>
```

```
<!-- body_type -->
<complexType name="p_type">
    <tox-alternatives>
        <tox-option odds="25">
            <element name="dummy" tox-omitTag="yes">
                <simpleType>
                    <restriction base="string">
                        <tox-string type="text"
                        tox-distribution="p_val_dis1"/>
                    </restriction>
                </simpleType>
            </element>
        </tox-option>
        <tox-option odds="75">
            <element name="dummy" tox-omitTag="yes">
                <simpleType>
                    <restriction base="string">
                        <tox-string type="text"
                        tox-distribution="p_val_dis2"/>
                    </restriction>
                </simpleType>
            </element>
        </tox-option>
    </tox-alternatives>
</complexType>
<complexType name="subsubsec_type">
    <attribute name="head">
        <simpleType>
            <restriction base="string">
                <tox-string type="text"
                tox-distribution="head_val_dis"/>
            </restriction>
        </simpleType>
    </attribute>
    <element name="p" type="p_type"
    tox-distribution="subsubsec_p_num_dis" minOccurs="0"
    maxOccurs="unbounded"/>
</complexType>
<complexType name="subsec_type">
    <attribute name="head">
        <simpleType>
            <restriction base="string">
                <tox-string type="text"
                tox-distribution="head_val_dis"/>
            </restriction>
```

```
        </simpleType>
    </attribute>
    <tox-alternatives>
        <tox-option odds="1"/>
        <tox-option odds="99">
            <element name="p" type="p_type"
            tox-distribution="subsec_p_num_dis" minOccurs="0"
            maxOccurs="unbounded"/>
        </tox-option>
    </tox-alternatives>
    <tox-alternatives>
        <tox-option odds="98"/>
        <tox-option odds="2">
            <element name="subsec" type="subsubsec_type"
            tox-distribution="subsubsec_num_dis" minOccurs="0"
            maxOccurs="unbounded"/>
        </tox-option>
    </tox-alternatives>
</complexType>
<complexType name="section_type">
    <attribute name="head">
        <simpleType>
            <restriction base="string">
                <tox-string type="text"
                tox-distribution="head_val_dis"/>
            </restriction>
        </simpleType>
    </attribute>
    <tox-alternatives>
        <tox-option odds="3"/>
        <tox-option odds="97">
            <element name="p" type="p_type"
            tox-distribution="section_p_num_dis" minOccurs="0"
            maxOccurs="unbounded"/>
        </tox-option>
    </tox-alternatives>
    <tox-alternatives>
        <tox-option odds="95"/>
        <tox-option odds="5">
            <element name="subsec" type="subsec_type"
            tox-distribution="subsec_num_dis" minOccurs="0"
            maxOccurs="unbounded"/>
        </tox-option>
    </tox-alternatives>
</complexType>
```

```
<complexType name="chapter_type">
    <attribute name="head">
        <simpleType>
            <restriction base="string">
                <tox-string type="text"
                tox-distribution="head_val_dis"/>
            </restriction>
        </simpleType>
    </attribute>
    <tox-alternatives>
        <tox-option odds="70">
            <element name="p" type="p_type"
            tox-distribution="chapter_p_num_dis"
            minOccurs="0" maxOccurs="unbounded"/>
        </tox-option>
        <tox-option odds="30"/>
    </tox-alternatives>
    <tox-alternatives>
        <tox-option odds="62"/>
        <tox-option odds="38">
            <element name="subsec" type="section_type"
            tox-distribution="section_num_dis" minOccurs="0"
            maxOccurs="unbounded"/>
        </tox-option>
    </tox-alternatives>
</complexType>
<complexType name="body_type">
<tox-alternatives>
    <tox-option odds="7"/>
    <tox-option odds="93">
        <element name="abstract">
            <complexType>
                <tox-alternatives>
                    <tox-option odds="95">
                        <element name="p" type="p_type"/>
                    </tox-option>
                    <tox-option odds="1.5">
                        <element name="p" type="p_type"
                        minOccurs="2" maxOccurs="2"/>
                    </tox-option>
                    <tox-option odds="2.5">
                        <element name="p" type="p_type"
                        minOccurs="3" maxOccurs="3"/>
                    </tox-option>
                    <tox-option odds="0.5">
```

```
                        <element name="p" type="p_type"
                        minOccurs="4" maxOccurs="4"/>
                    </tox-option>
                    <tox-option odds="0.25">
                        <element name="p" type="p_type"
                        minOccurs="5" maxOccurs="5"/>
                    </tox-option>
                    <tox-option odds="0.125">
                        <element name="p" type="p_type"
                        minOccurs="6" maxOccurs="6"/>
                    </tox-option>
                    <tox-option odds="0.125">
                        <element name="p" type="p_type"
                        minOccurs="7" maxOccurs="7"/>
                    </tox-option>
                </tox-alternatives>
            </complexType>
        </element>
    </tox-option>
</tox-alternatives>
<element name="section" type="chapter_type"
tox-distribution="chapter_num_dis" minOccurs="0"
maxOccurs="unbounded"/>
</complexType>
<simpleType name="pa_type">
    <restriction base="string">
        <tox-string type="text" tox-distribution="pa_val_dis"/>
    </restriction>
</simpleType>
<tox-list name="art_id_list">
    <element name="id" minOccurs="&article_num;"
    maxOccurs="&article_num;">
        <simpleType>
            <restriction base="positiveInteger">
                <maxInclusive value="&article_num;"/>
                <tox-number sequential="yes"/>
            </restriction>
        </simpleType>
    </element>
</tox-list>
<complexType name="epilog_type">
    <tox-alternatives>
    <tox-option odds="27"/>
    <tox-option odds="73">
        <element name="acknowledgements">
```

```
            <complexType>
            <tox-alternatives>
            <tox-option odds="99.5">
                <element name="pa" type="pa_type" minOccurs="1"
                maxOccurs="1"/>
            </tox-option>
            <tox-option odds="0.4">
                <element name="pa" type="pa_type" minOccurs="2"
                maxOccurs="2"/>
            </tox-option>
            <tox-option odds="0.1">
                <element name="pa" type="pa_type" minOccurs="3"
                maxOccurs="3"/>
            </tox-option>
            </tox-alternatives>
            </complexType>
        </element>
    </tox-option>
    </tox-alternatives>
    <tox-alternatives>
    <tox-option odds="3"/>
    <tox-option odds="97">
        <element name="references">
        <complexType>
        <element name="a_id" tox-distribution="citation_num_dis"
        minOccurs="0" maxOccurs="unbounded">
        <complexType>
            <tox-sample path="[art_id_list/id]" duplicates="no">
                <tox-expr value="[!]"/>
            </tox-sample>
        </complexType>
        </element>
        </complexType>
        </element>
    </tox-option>
    </tox-alternatives>
</complexType>
<!-- document -->
<tox-document name="output/article" copies="&article_num;"
starting-number="1">
    <element name="article">
        <complexType>
            <attribute name="id">
                <simpleType>
                    <restriction base="positiveInteger">
```

```
                        <maxInclusive value="&article_num;"/>
                        <tox-number sequential="yes"/>
                    </restriction>
                </simpleType>
            </attribute>
            <attribute name="lang">
                <simpleType>
                    <restriction base="string">
                        <tox-value>en</tox-value>
                    </restriction>
                </simpleType>
            </attribute>
            <element name="prolog" type="prolog_type"/>
            <element name="body" type="body_type"/>
            <element name="epilog" type="epilog_type"/>
        </complexType>
    </element>
</tox-document>
</tox-template>
```

B.3 DC/SD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_items "250">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
<!-- distribution list -->
<tox-distribution name="author_num_dis" type="uniform"
    minInclusive="1" maxInclusive="4"/>
<tox-distribution name="related_num_dis" type="uniform"
    minInclusive="0" maxInclusive="5"/>
<tox-distribution name="address_num_dis" type="uniform"
    minInclusive="1" maxInclusive="2"/>
<!-- tox list -->
<tox-list name="items">
    <element name="item" minOccurs="&num_items;"
    maxOccurs="&num_items;">
        <complexType>
            <element name="id">
                <simpleType>
                    <restriction base="positiveInteger">
                        <maxInclusive value="&num_items;"/>
                        <tox-number sequential="yes"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="price">
                <simpleType>
                    <restriction base="float">
                        <maxInclusive value="9999.99"/>
                        <minInclusive value="1.00"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="price_rnd">
                <simpleType>
                    <restriction base="float">
                        <maxExclusive value="1.00"/>
                        <minExclusive value="0.50"/>
                        <tox-number format="0.0000"/>
```

```
                </restriction>
            </simpleType>
        </element>
        <element name="date_of_release">
            <simpleType>
                <restriction base="date">
                    <tox-date start-date="1930-01-01"
                    end-date="&current_date;"/>
                </restriction>
            </simpleType>
        </element>
        <element name="available_days">
            <simpleType>
                <restriction base="nonNegativeInteger">
                    <maxInclusive value="30"/>
                    <minInclusive value="1"/>
                </restriction>
            </simpleType>
        </element>
    </complexType>
</element>
</tox-list>
<tox-list name="titles" readFrom="input/titles.xml">
    <element name="title" type="string"/>
</tox-list>
<tox-list name="lnames" readFrom="input/lnames.xml">
    <element name="lname" type="string"/>
</tox-list>
<tox-list name="subjects" readFrom="input/subjects.xml">
    <element name="subject" type="string"/>
</tox-list>
<tox-list name="backings" readFrom="input/backings.xml">
    <element name="backing" type="string"/>
</tox-list>
<tox-list name="firstnames" readFrom="input/firstnames.xml">
    <element name="name" type="string"/>
</tox-list>
<tox-list name="emails" readFrom="input/emails.xml">
    <element name="email" type="string"/>
</tox-list>
<tox-list name="provinces" readFrom="input/provinces.xml">
    <element name="province" type="string"/>
</tox-list>
<tox-list name="cities" readFrom="input/cities.xml">
    <element name="city" type="string"/>
```

```
</tox-list>
<tox-list name="countries" readFrom="input/countriesDC.xml">
    <element name="country">
        <complexType>
            <element name="name" type="string"/>
            <element name="rate" type="string"/>
            <element name="currency" type="string"/>
        </complexType>
    </element>
</tox-list>
<!-- type definition -->
<complexType name="addType">
    <element name="street_information">
        <complexType>
            <element name="street_address" maxOccurs="unbounded"
            minOccurs="0" tox-distribution="address_num_dis">
                <simpleType>
                    <restriction base="string">
                        <tox-string type="gibberish" maxLength="40"
                        minLength="15"/>
                    </restriction>
                </simpleType>
            </element>
        </complexType>
    </element>
    <tox-sample path="[cities/city]">
        <element name="name_of_city">
            <tox-expr value="[!]"/>
        </element>
    </tox-sample>
    <tox-sample path="[provinces/province]">
        <element name="name_of_state">
            <tox-expr value="[!]"/>
        </element>
    </tox-sample>
    <tox-alternatives>
    <tox-option odds="20">
    <element name="zip_code">
        <simpleType>
            <restriction base="string">
                <pattern value="[A-Z][0-9][A-Z] [0-9][A-Z][0-9]"/>
            </restriction>
        </simpleType>
    </element>
    </tox-option>
```

```
    <tox-option odds="80">
        <element name="zip_code">
            <simpleType>
                <restriction base="string">
                    <pattern value="[0-9]{5}"/>
                </restriction>
            </simpleType>
        </element>
    </tox-option>
    </tox-alternatives>
</complexType>
<complexType name="auth_addType">
    <element name="add" type="addType" tox-omitTag="yes"/>
    <tox-sample path="[countries/country]">
        <element name="name_of_country">
            <tox-expr value="[name]"/>
        </element>
    </tox-sample>
</complexType>
<complexType name="pub_addType">
    <element name="add" type="addType" tox-omitTag="yes"/>
    <tox-sample path="[countries/country]">
        <element name="country">
            <complexType>
                <element name="name">
                    <tox-expr value="[name]"/>
                </element>
                <element name="exchange_rate">
                    <tox-expr value="[rate]"/>
                </element>
                <element name="currency">
                    <tox-expr value="[currency]"/>
                </element>
            </complexType>
        </element>
    </tox-sample>
</complexType>
<complexType name="authorType">
    <tox-sample path="[lnames/lname]" name="last_name"
        duplicates="no">
    <element name="name">
        <complexType>
            <tox-sample path="[firstnames/name]">
                <element name="first_name">
                    <tox-expr value="[!]"/>
```

```
            </element>
        </tox-sample>
        <element name="middle_name">
            <simpleType>
                <restriction base="string">
                    <tox-string type="text" maxLength="20"
                    minLength="1"/>
                </restriction>
            </simpleType>
        </element>
        <element name="last_name">
            <tox-expr value="[!]"/>
        </element>
    </complexType>
</element>
<element name="date_of_birth">
    <simpleType>
        <restriction base="date">
            <tox-date start-date="1800-01-01"
            end-date="1900-01-01"/>
        </restriction>
    </simpleType>
</element>
<element name="biography">
    <simpleType>
        <restriction base="string">
            <tox-string type="text" maxLength="500"
            minLength="125"/>
        </restriction>
    </simpleType>
</element>
<element name="contact_information">
<complexType>
<element name="mailing_address" type="auth_addType">
</element>
<element name="phone_number">
<complexType mixed="true">
    <tox-number minInclusive="1" maxInclusive="99"
        format="'+'0"/>
    <tox-number minInclusive="10" maxInclusive="999"
        format="'('0')'"/>
    <tox-number minInclusive="100000"
        maxInclusive="99999999"/>
</complexType>
</element>
```

```
    <tox-sample path="[emails/email]">
        <element name="email_address">
            <tox-expr value="[$last_name/!]#'@'#[!]"/>
        </element>
    </tox-sample>
    </complexType>
    </element>
    </tox-sample>
</complexType>
<complexType name="publisherType">
    <element name="name">
    <simpleType>
    <restriction base="string">
        <tox-string type="text" maxLength="60" minLength="14"/>
    </restriction>
    </simpleType>
    </element>
    <element name="contact_information">
    <complexType>
        <element name="mailing_address" type="pub_addType">
        </element>
        <element name="FAX_number" maxOccurs="1" minOccurs="0">
            <complexType mixed="true">
                <tox-number minInclusive="1" maxInclusive="99"
                format="'+'0"/>
                <tox-number minInclusive="10" maxInclusive="999"
                format="'('0')'"/>
                <tox-number minInclusive="100000"
                maxInclusive="99999999"/>
            </complexType>
        </element>
        <element name="phone_number">
            <complexType mixed="true">
                <tox-number minInclusive="1" maxInclusive="99"
                format="'+'0"/>
                <tox-number minInclusive="10" maxInclusive="999"
                format="'('0')'"/>
                <tox-number minInclusive="100000"
                maxInclusive="99999999"/>
            </complexType>
        </element>
        <tox-sample path="[emails/email]">
            <element name="web_site">
                <tox-expr value="'http://'#'www.'#[!]"/>
            </element>
```

```
        </tox-sample>
    </complexType>
    </element>
</complexType>
<complexType name="null">
</complexType>
<!-- document definition -->
<tox-document name="output/catalog">
<element name="catalog">
<complexType>
<element name="item" minOccurs="&num_items;" maxOccurs="&num_items;">
<complexType>
<tox-scan path="[items/item]" name="item">
<attribute name="id" type="ID">
    <tox-expr value="'I'#[id]"/>
</attribute>
<tox-sample path="[titles/title]" duplicates="no">
<element name="title">
        <tox-expr value="[!]"/>
</element>
</tox-sample>
<element name="authors">
<complexType>
    <element name="author" type="authorType"
    tox-distribution="author_num_dis" minOccurs="1"
    maxOccurs="unbounded">
    </element>
</complexType>
</element>
<element name="date_of_release">
    <tox-expr value="[date_of_release]"/>
</element>
<element name="publisher" type="publisherType">
</element>
<tox-sample path="[subjects/subject]">
    <element name="subject">
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<element name="description">
<simpleType>
    <restriction base="string">
        <tox-string type="text" minLength="100" maxLength="500"/>
    </restriction>
</simpleType>
```

```
</element>
<element name="related_items">
<complexType>
    <element name="related_item" tox-distribution="related_num_dis"
    minOccurs="0" maxOccurs="unbounded">
        <complexType>
            <tox-sample path="[items/item]" duplicates="no">
                <element name="item_id">
                    <tox-expr value="'I'#[id]"/>
                </element>
            </tox-sample>
        </complexType>
    </element>
</complexType>
</element>
<element name="media">
<complexType>
    <element name="thumbnail">
    <complexType>
        <tox-if expr="EQ(&include_thumbnail_data;,1)">
            <tox-then>
                <element name="data">
                    <simpleType>
                        <restriction base="string">
                            <tox-string type="gibberish"
                            minLength="5120" maxLength="5120"/>
                        </restriction>
                    </simpleType>
                </element>
            </tox-then>
            <tox-else>
                <element name="data" type="null">
                </element>
            </tox-else>
        </tox-if>
    </complexType>
    </element>
    <element name="image">
    <complexType>
    <tox-if expr="EQ(&include_image_data;,1)">
    <tox-then>
        <tox-alternatives>
            <tox-option odds="45">
                <element name="data">
                    <simpleType>
```

```
            <restriction base="string">
                <tox-string type="gibberish"
                minLength="5120" maxLength="5120"/>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="35">
    <element name="data">
        <simpleType>
            <restriction base="string">
                <tox-string type="gibberish"
                minLength="10240" maxLength="10240"/>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="15">
    <element name="data">
        <simpleType>
            <restriction base="string">
                <tox-string type="gibberish"
                minLength="51200" maxLength="51200"/>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="4">
    <element name="data">
        <simpleType>
            <restriction base="string">
                <tox-string type="gibberish"
                minLength="102400" maxLength="102400"/>
            </restriction>
        </simpleType>
    </element>
</tox-option>
<tox-option odds="1">
    <element name="data">
        <simpleType>
            <restriction base="string">
                <tox-string type="gibberish"
                minLength="256000" maxLength="256000"/>
            </restriction>
        </simpleType>
```

```
                </element>
            </tox-option>
        </tox-alternatives>
    </tox-then>
    <tox-else>
        <element name="data" type="null">
        </element>
    </tox-else>
    </tox-if>
    </complexType>
    </element>
</complexType>
</element>
<element name="pricing">
<complexType>
    <element name="suggested_retail_price">
        <complexType>
            <attribute name="currency">
                <simpleType>
                    <restriction base="string">
                        <tox-value>Dollars</tox-value>
                    </restriction>
                </simpleType>
            </attribute>
            <tox-expr value="[price]" format="0.00"/>
        </complexType>
    </element>
    <element name="cost">
        <complexType>
            <attribute name="currency">
                <simpleType>
                    <restriction base="string">
                        <tox-value>Dollars</tox-value>
                    </restriction>
                </simpleType>
            </attribute>
            <tox-expr value="[price]*[price_rnd]" format="0.00"/>
        </complexType>
    </element>
    <element name="when_is_available">
        <tox-expr value="[date_of_release]+[available_days]"/>
    </element>
    <element name="quantity_in_stock">
        <simpleType>
            <restriction base="integer">
```

```
                <maxInclusive value="30"/>
                <minInclusive value="10"/>
            </restriction>
        </simpleType>
    </element>
</complexType>
</element>
<element name="attributes">
<complexType>
    <element name="ISBN">
        <simpleType>
            <restriction base="string">
                <pattern value="[0-9]{13}"/>
            </restriction>
        </simpleType>
    </element>
    <element name="number_of_pages">
        <simpleType>
            <restriction base="integer">
                <maxInclusive value="9999"/>
                <minInclusive value="20"/>
            </restriction>
        </simpleType>
    </element>
    <tox-sample path="[backings/backing]">
        <element name="type_of_book">
            <tox-expr value="[!]"/>
        </element>
    </tox-sample>
    <element name="size_of_book">
    <complexType>
        <element name="length">
            <complexType>
                <attribute name="unit">
                    <simpleType>
                        <restriction base="string">
                            <tox-value>Inch</tox-value>
                        </restriction>
                    </simpleType>
                </attribute>
                <tox-number maxInclusive="99.99"
                    minInclusive="0.01"/>
            </complexType>
        </element>
        <element name="width">
```

```
            <complexType>
                <attribute name="unit">
                    <simpleType>
                        <restriction base="string">
                            <tox-value>Inch</tox-value>
                        </restriction>
                    </simpleType>
                </attribute>
                <tox-number maxInclusive="99.99"
                    minInclusive="0.01"/>
            </complexType>
        </element>
        <element name="height">
            <complexType>
                <attribute name="unit">
                    <simpleType>
                        <restriction base="string">
                            <tox-value>Inch</tox-value>
                        </restriction>
                    </simpleType>
                </attribute>
                <tox-number maxInclusive="99.99"
                    minInclusive="0.01"/>
            </complexType>
        </element>
    </complexType>
    </element>
</complexType>
</element>
</tox-scan>
</complexType>
</element>
</complexType>
</element>
</tox-document>
</tox-template>
```

B.4 DC/MD

B.4.1 Order

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_orders "2592">
    <!ENTITY num_customers "2880">
    <!ENTITY num_items "1000">
    <!ENTITY num_authors "250">
    <!ENTITY num_addresses "5760">
    <!ENTITY num_countries "92">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
<!-- tox list -->
<tox-list name="orders0">
    <element name="order" minOccurs="&num_orders;"
    maxOccurs="&num_orders;">
        <complexType>
            <element name="order_days">
                <simpleType>
                    <restriction base="nonNegativeInteger">
                        <maxInclusive value="60"/>
                        <minInclusive value="1"/>
                    </restriction>
                </simpleType>
            </element>
        </complexType>
    </element>
</tox-list>
<tox-list name="orders1">
<element name="order" minOccurs="&num_orders;"
maxOccurs="&num_orders;">
<complexType>
    <tox-scan path="[orders0/order]">
        <element name="order_date" type="date">
            <tox-expr value="'&current_date;'-[order_days]"/>
        </element>
    </tox-scan>
    <element name="ship_days">
        <simpleType>
```

```
        <restriction base="nonNegativeInteger">
            <maxInclusive value="7"/>
            <minInclusive value="0"/>
        </restriction>
    </simpleType>
</element>
<element name="subtotal">
    <simpleType>
        <restriction base="float">
            <maxInclusive value="9999.99"/>
            <minInclusive value="10.00"/>
        </restriction>
    </simpleType>
</element>
<element name="order_lines">
    <complexType>
        <element name="order_line" maxOccurs="5" minOccurs="1">
            <complexType>
                <element name="item_id">
                    <simpleType>
                        <restriction base="int">
                            <tox-number minInclusive="1"
                            maxInclusive="&num_items;"/>
                        </restriction>
                    </simpleType>
                </element>
                <element name="quantity_of_item">
                    <simpleType>
                        <restriction base="int">
                            <tox-number minInclusive="1"
                            maxInclusive="300"/>
                        </restriction>
                    </simpleType>
                </element>
                <element name="discount_rate">
                    <simpleType>
                        <restriction base="float">
                            <tox-number minInclusive="0.00"
                            maxInclusive="0.03" format="0.00"/>
                        </restriction>
                    </simpleType>
                </element>
                <element name="special_instructions">
                    <simpleType>
                        <restriction base="string">
```

```
                                <tox-string type="gibberish"
                                maxLength="100" minLength="20"/>
                            </restriction>
                        </simpleType>
                    </element>
                </complexType>
            </element>
        </complexType>
    </element>
</complexType>
</element>
</tox-list>
<tox-list name="orders2">
<element name="order" minOccurs="&num_orders;"
maxOccurs="&num_orders;">
<complexType>
<element name="exp_days">
    <simpleType>
        <restriction base="nonNegativeInteger">
            <maxInclusive value="730"/>
            <minInclusive value="10"/>
        </restriction>
    </simpleType>
</element>
<tox-scan path="[orders1/order]" name="order">
<element name="subtotal">
    <tox-expr value="[subtotal]"/>
</element>
<element name="tax" type="float">
    <tox-expr value="[subtotal]*0.0825" format="0.00"/>
</element>
<element name="total">
    <tox-expr
    value="[subtotal]*1.0825+3.00+
    SUM[order_lines/order_line/quantity_of_item]" format="0.00"/>
</element>
<element name="order_date">
    <tox-expr value="[order_date]"/>
</element>
<element name="ship_date" type="date">
    <tox-expr value="[order_date]+[ship_days]"/>
</element>
<element name="order_lines">
<complexType>
<element name="order_line" maxOccurs="unbounded" tox-reset="yes">
```

```
    <complexType>
        <tox-scan path="[$order/order_lines/order_line]">
            <element name="item_id">
                <tox-expr value="[item_id]"/>
            </element>
            <element name="quantity_of_item">
                <tox-expr value="[quantity_of_item]"/>
            </element>
            <element name="discount_rate">
                <tox-expr value="[discount_rate]"/>
            </element>
            <element name="special_instructions">
                <tox-expr value="[special_instructions]"/>
            </element>
        </tox-scan>
    </complexType>
</element>
</complexType>
</element>
</tox-scan>
</complexType>
</element>
</tox-list>
<!-- list from file -->
<tox-list name="order_statuses" readFrom="input/order_statuses.xml">
    <element name="order_status" type="string"/>
</tox-list>
<tox-list name="ship_types" readFrom="input/ship_types.xml">
    <element name="ship_type" type="string"/>
</tox-list>
<tox-list name="card_types" readFrom="input/card_types.xml">
    <element name="card_type" type="string"/>
</tox-list>
<!-- document -->
<tox-document name="output/order" copies="&num_orders;"
starting-number="1">
<element name="order">
<complexType>
<tox-scan path="[orders2/order]" name="order">
<attribute name="id">
<simpleType>
<restriction base="long">
    <tox-number minInclusive="1" maxInclusive="&num_orders;"
    sequential="yes"/>
</restriction>
```

```
</simpleType>
</attribute>
<element name="customer_id">
<simpleType>
<restriction base="long">
    <tox-number minInclusive="1" maxInclusive="&num_customers;"/>
</restriction>
</simpleType>
</element>
<element name="order_date" type="date">
    <tox-expr value="[order_date]"/>
</element>
<element name="subtotal" type="float">
    <tox-expr value="[subtotal]" format="0.00"/>
</element>
<element name="tax" type="float">
    <tox-expr value="[tax]" format="0.00"/>
</element>
<element name="total">
    <tox-expr value="[total]" format="0.00"/>
</element>
<tox-sample path="[ship_types/ship_type]">
    <element name="ship_type">
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<element name="ship_date" type="date">
    <tox-expr value="[ship_date]"/>
</element>
<element name="bill_address_id">
<simpleType>
<restriction base="long">
    <tox-number minInclusive="1" maxInclusive="&num_addresses;"/>
</restriction>
</simpleType>
</element>
<element name="ship_address_id">
<simpleType>
<restriction base="long">
    <tox-number minInclusive="1" maxInclusive="&num_addresses;"/>
</restriction>
</simpleType>
</element>
<tox-sample path="[order_statuses/order_status]">
    <element name="order_status">
```

```
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<element name="credit_card_transaction">
<complexType>
<tox-sample path="[card_types/card_type]">
    <element name="credit_card_type">
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<element name="credit_card_number">
    <simpleType>
        <restriction base="string">
            <pattern value="[0-9]{16}"/>
        </restriction>
    </simpleType>
</element>
<element name="name_on_credit_card">
<simpleType>
    <restriction base="string">
        <tox-string type="text" maxLength="30" minLength="14"/>
    </restriction>
</simpleType>
</element>
<element name="expiration_date" type="date">
    <tox-expr value="'&current_date;'+[exp_days]"/>
</element>
<element name="authorization_id">
<simpleType>
    <restriction base="string">
        <tox-string type="gibberish" maxLength="15" minLength="15"/>
    </restriction>
</simpleType>
</element>
<element name="transaction_amount">
    <tox-expr value="[total]"/>
</element>
<element name="authorization_date" type="date">
    <tox-expr value="[ship_date]"/>
</element>
<element name="transaction_country_id">
    <simpleType>
        <restriction base="byte">
            <tox-number maxInclusive="92" minInclusive="1"/>
        </restriction>
```

```
    </simpleType>
</element>
</complexType>
</element>
<element name="order_lines">
<complexType>
<element name="order_line" maxOccurs="unbounded" tox-reset="yes">
    <complexType>
        <tox-scan path="[$order/order_lines/order_line]">
            <attribute name="id">
            <simpleType>
            <restriction base="byte">
                <tox-number minInclusive="1" maxInclusive="100"/>
            </restriction>
            </simpleType>
            </attribute>
            <element name="item_id">
                <tox-expr value="[item_id]"/>
            </element>
            <element name="quantity_of_item">
                <tox-expr value="[quantity_of_item]"/>
            </element>
            <element name="discount_rate">
                <tox-expr value="[discount_rate]"/>
            </element>
            <element name="special_instructions">
                <tox-expr value="[special_instructions]"/>
            </element>
        </tox-scan>
    </complexType>
</element>
</complexType>
</element>
</tox-scan>
</complexType>
</element>
</tox-document>
</tox-template>
```

B.4.2 Customer

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_orders "2592">
    <!ENTITY num_customers "2880">
    <!ENTITY num_items "1000">
    <!ENTITY num_authors "250">
    <!ENTITY num_addresses "5760">
    <!ENTITY num_countries "92">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
<!-- tox-list -->
<tox-list name="digsyl" readFrom="input/digsyl.xml">
    <element name="customer">
        <complexType>
            <element name="name" type="string"/>
            <element name="password" type="string"/>
        </complexType>
    </element>
</tox-list>
<tox-list name="custs0">
    <element name="cust" minOccurs="&num_customers;"
    maxOccurs="&num_customers;">
        <complexType>
            <element name="since_days">
                <simpleType>
                    <restriction base="nonNegativeInteger">
                        <maxInclusive value="730"/>
                        <minInclusive value="1"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="last_days">
                <simpleType>
                    <restriction base="nonNegativeInteger">
                        <maxInclusive value="60"/>
                        <minInclusive value="0"/>
                    </restriction>
                </simpleType>
            </element>
```

```
        </complexType>
    </element>
</tox-list>
<tox-list name="custs1">
    <element name="cust" minOccurs="&num_customers;"
    maxOccurs="&num_customers;">
        <complexType>
            <tox-scan path="[custs0/cust]">
                <element name="date_of_registration" type="date">
                    <tox-expr value="'&current_date;'-[since_days]"/>
                </element>
            </tox-scan>
        </complexType>
    </element>
</tox-list>
<tox-list name="custs2">
    <element name="cust" minOccurs="&num_customers;"
    maxOccurs="&num_customers;">
    <complexType>
    <tox-scan path="[custs1/cust]" name="a">
    <element name="date_of_registration" type="date">
        <tox-expr value="[date_of_registration]"/>
    </element>
    <element name="date_of_last_visit" type="date">
    <complexType>
    <tox-sample path="[custs0/cust]">
        <tox-expr value="[$a/date_of_registration]+[last_days]"/>
    </tox-sample>
    </complexType>
    </element>
    </tox-scan>
    </complexType>
    </element>
</tox-list>
<tox-list name="custs3">
    <element name="cust" minOccurs="&num_customers;"
    maxOccurs="&num_customers;">
    <complexType>
    <tox-scan path="[custs2/cust]">
    <element name="date_of_registration" type="date">
        <tox-expr value="[date_of_registration]"/>
    </element>
    <tox-if expr="GT([date_of_last_visit],'&current_date;')">
        <tox-then>
            <element name="date_of_last_visit" type="date">
```

```
                <tox-expr value="'&current_date;'"/>
            </element>
        </tox-then>
        <tox-else>
            <element name="date_of_last_visit" type="date">
                <tox-expr value="[date_of_last_visit]"/>
            </element>
        </tox-else>
    </tox-if>
    <element name="email_m">
        <simpleType>
            <restriction base="string">
                <tox-string type="text" maxLength="9"
                minLength="2"/>
            </restriction>
        </simpleType>
    </element>
    </tox-scan>
    </complexType>
    </element>
</tox-list>
<!-- document -->
<tox-document name="output/customer">
<element name="customers">
<complexType>
<element name="customer" maxOccurs="&num_customers;"
    minOccurs="&num_customers;">
<complexType>
<attribute name="id">
<simpleType>
    <restriction base="long">
        <tox-number minInclusive="1"
        maxInclusive="&num_customers;" sequential="yes"/>
    </restriction>
</simpleType>
</attribute>
<tox-scan path="[digsyl/customer]" name="digsyl">
<element name="user_name">
    <tox-expr value="[name]"/>
</element>
<element name="password">
    <tox-expr value="[password]"/>
</element>
<tox-sample path="[custs3/cust]" duplicates="no">
<element name="first_name">
```

```
    <simpleType>
        <restriction base="string">
            <tox-string type="text" maxLength="15" minLength="8"/>
        </restriction>
    </simpleType>
</element>
<element name="last_name">
    <simpleType>
        <restriction base="string">
            <tox-string type="text" maxLength="15" minLength="8"/>
        </restriction>
    </simpleType>
</element>
<element name="address_id">
    <simpleType>
        <restriction base="long">
            <tox-number minInclusive="1"
            maxInclusive="&num_addresses;"/>
        </restriction>
    </simpleType>
</element>
<element name="phone_number">
    <simpleType>
        <restriction base="unsignedLong">
            <tox-number minInclusive="100000000"
            maxInclusive="9999999999999999"/>
        </restriction>
    </simpleType>
</element>
<element name="email_address">
    <tox-expr value="[$digsyl/name]#'@'#[email_m]#'.com'"/>
</element>
<element name="date_of_registration">
    <tox-expr value="[date_of_registration]"/>
</element>
<element name="date_of_last_visit">
    <tox-expr value="[date_of_last_visit]"/>
</element>
<element name="start_of_current_session">
    <tox-expr value="'&current_date;'#'T00:00:00.000-05:00'"/>
</element>
<element name="current_session_expiry">
    <tox-expr value="'&current_date;'#'T02:00:00.000-05:00'"/>
</element>
<element name="discount_rate">
```

```
    <simpleType>
        <restriction base="float">
            <tox-number minInclusive="0.00" maxInclusive="0.50"
            format="0.00"/>
        </restriction>
    </simpleType>
</element>
<element name="balance">
    <simpleType>
        <restriction base="float">
            <tox-value>0.00</tox-value>
        </restriction>
    </simpleType>
</element>
<element name="YTD_payment">
    <simpleType>
        <restriction base="float">
            <tox-number minInclusive="0.00" maxInclusive="999.99"
            format="0.00"/>
        </restriction>
    </simpleType>
</element>
<element name="birth_date">
    <simpleType>
        <restriction base="date">
            <tox-date start-date="1880-01-01"
            end-date="&current_date;"/>
        </restriction>
    </simpleType>
</element>
<element name="miscellaneous_information">
    <simpleType>
        <restriction base="string">
            <tox-string type="gibberish" maxLength="500"
            minLength="100"/>
        </restriction>
    </simpleType>
</element>
</tox-sample>
</tox-scan>
</complexType>
</element>
</complexType>
</element>
</tox-document>
```

```
</tox-template>
```

B.4.3 Item

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_orders "2592">
    <!ENTITY num_customers "2880">
    <!ENTITY num_items "1000">
    <!ENTITY num_authors "250">
    <!ENTITY num_addresses "5760">
    <!ENTITY num_countries "92">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
    <!-- tox list -->
    <tox-list name="id_list">
        <element name="item" minOccurs="&num_items;"
        maxOccurs="&num_items;">
            <complexType>
                <element name="id">
            <simpleType>
                <restriction base="positiveInteger">
                    <maxInclusive value="&num_items;"/>
                    <tox-number sequential="yes"/>
                </restriction>
            </simpleType>
            </element>
            </complexType>
        </element>
    </tox-list>
    <tox-list name="qu_ids">
        <element name="id" maxOccurs="unbounded">
            <complexType>
                <tox-scan path="[id_list/item]"
                where="LT([id],&num_items;/4)">
                    <tox-expr value="[id]"/>
                </tox-scan>
            </complexType>
        </element>
    </tox-list>
    <tox-list name="items">
        <element name="item" minOccurs="&num_items;"
        maxOccurs="&num_items;">
```

```
<complexType>
    <element name="id">
        <simpleType>
            <restriction base="positiveInteger">
                <maxInclusive value="&num_items;"/>
                <tox-number sequential="yes"/>
            </restriction>
        </simpleType>
    </element>
    <element name="price">
        <simpleType>
            <restriction base="float">
                <maxInclusive value="9999.99"/>
                <minInclusive value="1.00"/>
            </restriction>
        </simpleType>
    </element>
    <element name="price_rnd">
        <simpleType>
            <restriction base="float">
                <maxExclusive value="1.00"/>
                <minExclusive value="0.50"/>
                <tox-number format="0.0000"/>
            </restriction>
        </simpleType>
    </element>
    <element name="date_of_release">
        <simpleType>
            <restriction base="date">
                <tox-date start-date="1930-01-01"
                end-date="&current_date;"/>
            </restriction>
        </simpleType>
    </element>
    <element name="available_days">
        <simpleType>
            <restriction base="nonNegativeInteger">
                <maxInclusive value="30"/>
                <minInclusive value="1"/>
            </restriction>
        </simpleType>
    </element>
    <element name="book_l">
        <simpleType>
            <restriction base="float">
```

```
                        <maxInclusive value="99.99"/>
                        <minInclusive value="0.01"/>
                        <tox-format value="0.00"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="book_h">
                <simpleType>
                    <restriction base="float">
                        <maxInclusive value="99.99"/>
                        <minInclusive value="0.01"/>
                        <tox-format value="0.00"/>
                    </restriction>
                </simpleType>
            </element>
            <element name="book_w">
                <simpleType>
                    <restriction base="float">
                        <maxInclusive value="99.99"/>
                        <minInclusive value="0.01"/>
                        <tox-format value="0.00"/>
                    </restriction>
                </simpleType>
            </element>
        </complexType>
    </element>
</tox-list>
<tox-list name="titles" readFrom="input/titles.xml">
    <element name="title" type="string"/>
</tox-list>
<tox-list name="subjects" readFrom="input/subjects.xml">
    <element name="subject" type="string"/>
</tox-list>
<tox-list name="backings" readFrom="input/backings.xml">
    <element name="backing" type="string"/>
</tox-list>
<!-- empty -->
<complexType name="null">
</complexType>
<!-- document definition -->
<tox-document name="output/item">
<element name="items">
<complexType>
<element name="item" minOccurs="&num_items;"
    maxOccurs="&num_items;">
```

```
<complexType>
<tox-scan path="[items/item]" name="item">
<attribute name="id">
    <tox-expr value="[id]"/>
</attribute>
<tox-sample path="[titles/title]"
duplicates="no">
    <element name="title">
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<tox-if expr="LEQ([id],&num_items;/4)">
    <tox-then>
        <element name="author_id">
            <tox-expr value="[id]"/>
        </element>
    </tox-then>
    <tox-else>
        <tox-sample path="[qu_ids/id]"
        duplicates="no">
            <element name="author_id">
                <tox-expr value="[!]"/>
            </element>
        </tox-sample>
    </tox-else>
</tox-if>
<element name="date_of_release">
    <tox-expr value="[date_of_release]"/>
</element>
<element name="name_of_publisher">
    <simpleType>
        <restriction base="string">
            <tox-string type="text"
            maxLength="60" minLength="14"/>
        </restriction>
    </simpleType>
</element>
<tox-sample path="[subjects/subject]">
    <element name="subject">
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<element name="description">
    <simpleType>
        <restriction base="string">
```

```
            <tox-string type="gibberish"
            minLength="100" maxLength="500"/>
        </restriction>
    </simpleType>
</element>
<element name="related_item_id" minOccurs="5" maxOccurs="5">
    <complexType>
        <tox-sample path="[id_list/item]"
        duplicates="no">
            <tox-expr value="[id]"/>
        </tox-sample>
    </complexType>
</element>
        <tox-if expr="EQ(&include_thumbnail_data;,1)">
            <tox-then>
                <element name="thumbnail">
                    <simpleType>
                        <restriction base="string">
                            <tox-string type="gibberish"
                            minLength="5120" maxLength="5120"/>
                        </restriction>
                    </simpleType>
                </element>
            </tox-then>
            <tox-else>
                <element name="thumbnail" type="null">
                        </element>
            </tox-else>
        </tox-if>
        <tox-if expr="EQ(&include_image_data;,1)">
            <tox-then>
                <tox-alternatives>
                    <tox-option odds="45">
                        <element name="image">
                            <simpleType>
                                <restriction base="string">
                                    <tox-string type="gibberish"
                                    minLength="5120"
                                    maxLength="5120"/>
                                </restriction>
                            </simpleType>
                        </element>
                    </tox-option>
                    <tox-option odds="35">
                        <element name="image">
```

```
                    <simpleType>
                        <restriction base="string">
                            <tox-string type="gibberish"
                            minLength="10240"
                            maxLength="10240"/>
                        </restriction>
                    </simpleType>
                </element>
            </tox-option>
            <tox-option odds="15">
                <element name="image">
                    <simpleType>
                        <restriction base="string">
                            <tox-string type="gibberish"
                            minLength="51200"
                            maxLength="51200"/>
                        </restriction>
                    </simpleType>
                </element>
            </tox-option>
            <tox-option odds="4">
                <element name="image">
                    <simpleType>
                        <restriction base="string">
                            <tox-string type="gibberish"
                            minLength="102400"
                            maxLength="102400"/>
                        </restriction>
                    </simpleType>
                </element>
            </tox-option>
            <tox-option odds="1">
                <element name="image">
                    <simpleType>
                        <restriction base="string">
                            <tox-string type="gibberish"
                            minLength="256000"
                            maxLength="256000"/>
                        </restriction>
                    </simpleType>
                </element>
            </tox-option>
        </tox-alternatives>
    </tox-then>
    <tox-else>
```

```
                <element name="image" type="null">
                        </element>
            </tox-else>
        </tox-if>
<element name="suggested_retail_price">
    <tox-expr value="[price]" format="0.00"/>
</element>
<element name="cost">
    <tox-expr value="[price]*[price_rnd]" format="0.00"/>
</element>
<element name="when_is_available">
    <tox-expr value="[date_of_release]+[available_days]"/>
</element>
<element name="quantity_in_stock">
    <simpleType>
        <restriction base="integer">
            <maxInclusive value="30"/>
            <minInclusive value="10"/>
        </restriction>
    </simpleType>
</element>
<element name="ISBN">
    <simpleType>
        <restriction base="string">
            <pattern value="[0-9][A-Z]{13}"/>
        </restriction>
    </simpleType>
</element>
<element name="number_of_pages">
    <simpleType>
        <restriction base="integer">
            <maxInclusive value="9999"/>
            <minInclusive value="20"/>
        </restriction>
    </simpleType>
</element>
<tox-sample path="[backings/backing]">
    <element name="type_of_book">
        <tox-expr value="[!]"/>
    </element>
</tox-sample>
<element name="size_of_book">
    <tox-expr value="[book_l]#'x'#[book_w]#'x'#[book_h]"/>
</element>
</tox-scan>
```

```
    </complexType>
    </element>
    </complexType>
    </element>
    </tox-document>
</tox-template>
```

B.4.4 Author

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_orders "2592">
    <!ENTITY num_customers "2880">
    <!ENTITY num_items "1000">
    <!ENTITY num_authors "250">
    <!ENTITY num_addresses "5760">
    <!ENTITY num_countries "92">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
    <!-- tox-list -->
    <tox-list name="lnames" readFrom="input/lnames.xml">
        <element name="lname" type="string"/>
    </tox-list>
    <!-- document -->
    <tox-document name="output/author">
        <element name="authors">
            <complexType>
                <element name="author" maxOccurs="&num_authors;"
                minOccurs="&num_authors;">
                    <complexType>
                        <attribute name="id">
                            <simpleType>
                                <restriction base="long">
                                    <tox-number minInclusive="1"
                                    maxInclusive="&num_authors;"
                                    sequential="yes"/>
                                </restriction>
                            </simpleType>
                        </attribute>
                        <element name="first_name">
                            <simpleType>
                                <restriction base="string">
                                    <tox-string type="text"
                                    maxLength="20" minLength="3"/>
                                </restriction>
                            </simpleType>
                        </element>
                        <element name="middle_name">
```

```
                        <simpleType>
                            <restriction base="string">
                                <tox-string type="text"
                                maxLength="20" minLength="1"/>
                            </restriction>
                        </simpleType>
                    </element>
                    <tox-sample path="[lnames/lname]"
                    name="last_name" duplicates="no">
                    <element name="last_name">
                        <tox-expr value="[!]"></tox-expr>
                    </element>
                    </tox-sample>
                    <element name="date_of_birth">
                        <simpleType>
                            <restriction base="date">
                                <tox-date start-date="1800-01-01"
                                end-date="1990-01-01"></tox-date>
                            </restriction>
                        </simpleType>
                    </element>
                    <element name="biography">
                        <simpleType>
                            <restriction base="string">
                                <tox-string type="text"
                                maxLength="500" minLength="125"/>
                            </restriction>
                        </simpleType>
                    </element>
                </complexType>
            </element>
        </complexType>
    </element>
</tox-document>
</tox-template>
```

B.4.5 Address

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_orders "2592">
    <!ENTITY num_customers "2880">
    <!ENTITY num_items "1000">
    <!ENTITY num_authors "250">
    <!ENTITY num_addresses "5760">
    <!ENTITY num_countries "92">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
    <!-- document -->
    <tox-document name="output/address">
        <element name="addresses">
            <complexType>
                <element name="address" maxOccurs="&num_addresses;"
                minOccurs="&num_addresses;">
                    <complexType>
                        <attribute name="id">
                            <simpleType>
                                <restriction base="long">
                                    <tox-number minInclusive="1"
                                    maxInclusive="&num_addresses;"
                                    sequential="yes"/>
                                </restriction>
                            </simpleType>
                        </attribute>
                        <element name="street_address" maxOccurs="2"
                        minOccurs="2">
                            <simpleType>
                                <restriction base="string">
                                    <tox-string type="gibberish"
                                    maxLength="40" minLength="15"/>
                                </restriction>
                            </simpleType>
                        </element>
                        <!--
                        <element name="street_address_2">
                            <simpleType>
                                <restriction base="string">
```

```
                                <tox-string type="gibberish"
                                maxLength="40" minLength="15"/>
                            </restriction>
                        </simpleType>
                    </element>
                    -->
                    <element name="name_of_city">
                        <simpleType>
                            <restriction base="string">
                                <tox-string type="text"
                                maxLength="30" minLength="4"/>
                            </restriction>
                        </simpleType>
                    </element>
                    <element name="name_of_state">
                        <simpleType>
                            <restriction base="string">
                                <tox-string type="text"
                                maxLength="20" minLength="2"/>
                            </restriction>
                        </simpleType>
                    </element>
                    <element name="zip_code">
                        <simpleType>
                            <restriction base="string">
                                <tox-string type="text"
                                maxLength="10" minLength="5"/>
                            </restriction>
                        </simpleType>
                    </element>
                    <element name="country_id">
                        <simpleType>
                            <restriction base="byte">
                                <tox-number minInclusive="1"
                                maxInclusive="&num_countries;"/>
                            </restriction>
                        </simpleType>
                    </element>
                </complexType>
            </element>
        </complexType>
    </element>
</tox-document>
</tox-template>
```

B.4.6 Country

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE tox-template
SYSTEM "ToXgene1_1.dtd" [
    <!ENTITY num_orders "2592">
    <!ENTITY num_customers "2880">
    <!ENTITY num_items "1000">
    <!ENTITY num_authors "250">
    <!ENTITY num_addresses "5760">
    <!ENTITY num_countries "92">
    <!ENTITY current_date "2002-12-31">
    <!ENTITY include_thumbnail_data "0">
    <!ENTITY include_image_data "0">
]>
<tox-template>
    <!-- tox-list -->
    <tox-list name="countries" readFrom="input/countriesDC.xml">
        <element name="country">
            <complexType>
                <element name="name" type="string"/>
                <element name="rate" type="string"/>
                <element name="currency" type="string"/>
            </complexType>
        </element>
    </tox-list>
    <!-- document -->
    <tox-document name="output/country">
        <element name="countries">
            <complexType>
                <element name="country" maxOccurs="&num_countries;"
                    minOccurs="&num_countries;">
                    <complexType>
                        <attribute name="id">
                            <simpleType>
                                <restriction base="byte">
                                    <tox-number minInclusive="1"
                                    maxInclusive="&num_countries;"
                                    sequential="yes"/>
                                </restriction>
                            </simpleType>
                        </attribute>
                        <tox-scan path="[countries/country]">
                            <element name="name">
                                <tox-expr value="[name]"/>
```

```
                        </element>
                        <element name="exchange_rate">
                            <tox-expr value="[rate]"/>
                        </element>
                        <element name="currency">
                            <tox-expr value="[currency]"/>
                        </element>
                    </tox-scan>
                </complexType>
            </element>
        </complexType>
    </element>
</tox-document>
</tox-template>
```

Appendix C

Parameters for Templates

C.1 TC/SD

Name	Type	Min	Max	Mean	Var
Number of Headword	-	1	5	-	-
Value of Pronunciation	Lognormal	2	24	2.39	0.34
Number of Variant Form	-	1	4	-	-
Number of Variant Form List	Normal	1	9	1.48	0.98
Number of Citation in Etymology	Lognormal	1	16	1.13	0.47
Number of Sense	Normal	1	10	1.29	0.88
Value of Definition	Lognormal	1	459	4.35	0.73
Number of Citation in Definition	-	0	5	-	-
Number of Quotation	Normal	1	20	3.68	2.56
Value of Work	Lognormal	2	60	2.55	0.51
Value of Location	Lognormal	1	58	2.48	0.4
Value of Bibliography	Uniform	8	27	-	-
Value of Text in Font Tags	Lognormal	1	52	2.14	0.59
Value of Quotation Text	Lognormal	4	507	4.54	0.51
Number of Citation in Quotation Text	-	0	2	-	-

C.2 TC/MD

Name	Type	Min	Max	Mean	Var
Value of Title	Lognormal	3	252	3.95	0.33
Number of Author	Lognormal	1	48	1.05	0.68
Number of Keyword	Logormal	1	19	1.87	0.22
Value of Keyword	Lognormal	1	96	2.95	0.38
Number of Section	Normal	1	15	4.19	1.29
Value of Heading	Normal	1	200	24.96	24.18
Number of P in Section	Lognormal	1	29	1.28	0.67
Number of P in Subsection	Lognormal	1	23	1.58	0.37
Number of Subsection	Normal	1	13	5	1
Value of P1	Normal	1	64	37.64	15.60
Value of P2	Lognormal	5	20k	5.41	1.19
Number of P in Subsubsection	Normal	1	11	1.76	1.45
Number of Subsubsection	Lognormal	0	31	0.89	0.49
Number of P in Subsubsubsection	Exponential	0	12	0.49	-
Number of Subsubsubsection	Lognormal	0	8	0.87	0.43
Value of P in Ack	Lognormal	5	3000	5.66	0.60
Number of References	Lognormal	1	-	3.74	0.44

C.3 DC/SD

Name	Type	Min	Max	Mean	Var
Number of Author	Uniform	1	4	2.5	-
Number of Street	Uniform	1	2	1.5	-
Number of Related Item	Uniform	0	5	2.5	-

See the TPC-W specification for the description of other parameters.

C.4 DC/MD

Name	Type	Min	Max	Mean	Var
Number of Order Line	Normal	1	5	3	-

See the TPC-W specification for the description of other parameters.

Appendix D

Schemas for Document Classes

D.1 TC/SD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="a" type="xs:string"/>
    <xs:element name="bib">
        <xs:simpleType>
            <xs:restriction base="xs:string"/>
        </xs:simpleType>
    </xs:element>
    <xs:element name="cr" type="xs:IDREF"/>
    <xs:element name="def">
        <xs:complexType mixed="true">
            <xs:choice minOccurs="0" maxOccurs="4">
                <xs:element ref="cr"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="dictionary">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="e" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="e">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="hwg"/>
                <xs:element ref="vfl" minOccurs="0"/>
```

```
            <xs:element ref="et" minOccurs="0"/>
            <xs:element ref="ss"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:ID"/>
    </xs:complexType>
</xs:element>
<xs:element name="et">
    <xs:complexType>
        <xs:sequence maxOccurs="16">
            <xs:element ref="cr"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="hw">
    <xs:simpleType>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
</xs:element>
<xs:element name="hwg">
    <xs:complexType>
        <xs:choice maxOccurs="15">
            <xs:element ref="hw"/>
            <xs:element ref="pr" minOccurs="0"/>
            <xs:element ref="pos" minOccurs="0"/>
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="pos">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="adj."/>
            <xs:enumeration value="adv."/>
            <xs:enumeration value="conj."/>
            <xs:enumeration value="n."/>
            <xs:enumeration value="prep."/>
            <xs:enumeration value="v."/>
            <xs:enumeration value="int."/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="pr" type="xs:string"/>
<xs:element name="q">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="qd"/>
```

```
            <xs:element ref="a" minOccurs="0"/>
            <xs:element ref="w"/>
            <xs:element ref="bib" minOccurs="0"/>
            <xs:element ref="loc"/>
            <xs:element ref="qt"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="qd">
    <xs:simpleType>
        <xs:restriction base="xs:short"/>
    </xs:simpleType>
</xs:element>
<xs:element name="qt">
    <xs:complexType mixed="true">
        <xs:choice minOccurs="0" maxOccurs="6">
            <xs:element ref="cr"/>
            <xs:element name="i" type="xs:string"/>
            <xs:element name="b" type="xs:string"/>
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="loc" type="xs:string"/>
<xs:element name="qp">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="q" maxOccurs="20"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="s">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="def"/>
            <xs:element ref="qp"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ss">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="s" maxOccurs="10"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

```
    <xs:element name="vd">
        <xs:simpleType>
            <xs:restriction base="xs:string"/>
        </xs:simpleType>
    </xs:element>
    <xs:element name="vf">
        <xs:simpleType>
            <xs:restriction base="xs:string"/>
        </xs:simpleType>
    </xs:element>
    <xs:element name="vfl">
        <xs:complexType>
            <xs:choice maxOccurs="45">
                <xs:element ref="vd" minOccurs="0"/>
                <xs:element ref="vf" maxOccurs="4"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>
    <xs:element name="w" type="xs:string"/>
</xs:schema>
```

D.2 TC/MD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
<xs:element name="a_id" type="xs:string"/>
<xs:element name="abstract">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="p" maxOccurs="7"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="article">
<xs:complexType>
    <xs:sequence>
    <xs:element ref="prolog"/>
    <xs:element ref="body"/>
    <xs:element name="epilog">
    <xs:complexType>
        <xs:sequence>
        <xs:element name="acknowledgements" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                <xs:element ref="pa" maxOccurs="3"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="references" minOccurs="0">
            <xs:complexType>
                <xs:sequence>
                <xs:element ref="a_id" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        </xs:sequence>
    </xs:complexType>
    </xs:element>
    </xs:sequence>
    <xs:attribute name="id" type="xs:byte" use="required"/>
    <xs:attribute name="lang" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="author">
    <xs:complexType>
```

```
        <xs:sequence>
            <xs:element ref="name"/>
            <xs:element ref="contact"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="authors">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="author" maxOccurs="48"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="body">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="abstract" minOccurs="0"/>
            <xs:element ref="section" maxOccurs="15"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="section">
    <xs:complexType>
    <xs:sequence>
        <xs:element ref="p" minOccurs="0" maxOccurs="29"/>
        <xs:element ref="subsec" minOccurs="0" maxOccurs="23"/>
    </xs:sequence>
    <xs:attribute name="heading" type="xs:string" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="city" type="xs:string"/>
<xs:element name="contact">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="email" minOccurs="0"/>
            <xs:element ref="phone" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="country" type="xs:string"/>
<xs:element name="date" type="xs:date"/>
<xs:element name="dateline">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="city"/>
```

```
            <xs:element ref="country"/>
            <xs:element ref="date"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="email">
    <xs:simpleType>
        <xs:restriction base="xs:string"/>
    </xs:simpleType>
</xs:element>
<xs:element name="genre" type="xs:string"/>
<xs:element name="keyword" type="xs:string"/>
<xs:element name="keywords">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="keyword" maxOccurs="19"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="name" type="xs:string"/>
<xs:element name="p" type="xs:string"/>
<xs:element name="pa" type="xs:string"/>
<xs:element name="phone" type="xs:string"/>
<xs:element name="prolog">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="title"/>
            <xs:element ref="authors" minOccurs="0"/>
            <xs:element ref="dateline" minOccurs="0"/>
            <xs:element ref="genre" minOccurs="0"/>
            <xs:element ref="keywords" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="subsec">
<xs:complexType>
<xs:sequence>
    <xs:element ref="p" minOccurs="0" maxOccurs="46"/>
    <xs:element name="subsec" minOccurs="0" maxOccurs="31">
    <xs:complexType>
    <xs:sequence>
    <xs:element ref="p" minOccurs="0" maxOccurs="11"/>
    <xs:element name="subsec" minOccurs="0" maxOccurs="8">
        <xs:complexType>
            <xs:sequence>
```

```
                <xs:element ref="p" maxOccurs="12"/>
            </xs:sequence>
            <xs:attribute name="heading" type="xs:string"
            use="required"/>
        </xs:complexType>
    </xs:element>
    </xs:sequence>
    <xs:attribute name="heading" type="xs:string" use="required"/>
    </xs:complexType>
    </xs:element>
</xs:sequence>
<xs:attribute name="heading" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
<xs:element name="title" type="xs:string"/>
</xs:schema>
```

D.3 DC/SD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="FAX_number" type="xs:string"/>
    <xs:element name="ISBN" type="xs:string"/>
    <xs:element name="attributes">
        <xs:complexType>
            <xs:all>
                <xs:element ref="ISBN"/>
                <xs:element ref="number_of_pages"/>
                <xs:element ref="type_of_book"/>
                <xs:element ref="size_of_book"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="author">
        <xs:complexType>
            <xs:all>
                <xs:element name="name">
                    <xs:complexType>
                        <xs:all>
                            <xs:element ref="first_name"/>
                            <xs:element ref="middle_name"/>
                            <xs:element ref="last_name"/>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
                <xs:element ref="date_of_birth"/>
                <xs:element ref="biography"/>
                <xs:element name="contact_information">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="mailing_address">
                                <xs:complexType>
                                <xs:all>
                                <xs:element ref="street_information"/>
                                <xs:element ref="name_of_city"/>
                                <xs:element ref="name_of_state"/>
                                <xs:element ref="zip_code"/>
                                <xs:element name="name_of_country"
                                    type="xs:string"/>
                                </xs:all>
                                </xs:complexType>
```

```
                        </xs:element>
                        <xs:element ref="phone_number"/>
                        <xs:element ref="email_address"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="authors">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="author" maxOccurs="4"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="biography" type="xs:string"/>
<xs:element name="catalog">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="item" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="cost">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:decimal">
                <xs:attribute name="currency" type="xs:string"
                    use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="country">
    <xs:complexType>
        <xs:all>
            <xs:element name="name" type="xs:string"/>
            <xs:element ref="exchange_rate"/>
            <xs:element ref="currency"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="currency" type="xs:string"/>
<xs:element name="data" type="xs:string"/>
```

```
<xs:element name="date_of_birth" type="xs:date"/>
<xs:element name="date_of_release" type="xs:date"/>
<xs:element name="description" type="xs:string"/>
<xs:element name="email_address" type="xs:string"/>
<xs:element name="web_site" type="xs:string"/>
<xs:element name="exchange_rate" type="xs:decimal"/>
<xs:element name="first_name" type="xs:string"/>
<xs:element name="height">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:decimal">
                <xs:attribute name="unit" type="xs:string"
                    use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="image">
    <xs:complexType>
        <xs:all>
            <xs:element ref="data"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="item">
    <xs:complexType>
        <xs:all>
            <xs:element ref="title"/>
            <xs:element ref="authors"/>
            <xs:element ref="date_of_release"/>
            <xs:element ref="publisher"/>
            <xs:element ref="subject"/>
            <xs:element ref="description"/>
            <xs:element ref="related_items"/>
            <xs:element ref="media"/>
            <xs:element ref="pricing"/>
            <xs:element ref="attributes"/>
        </xs:all>
        <xs:attribute name="id" type="xs:ID" use="required"/>
    </xs:complexType>
</xs:element>
<xs:element name="item_id" type="xs:IDREF"/>
<xs:element name="last_name" type="xs:string"/>
<xs:element name="length">
    <xs:complexType>
```

```
        <xs:simpleContent>
            <xs:extension base="xs:decimal">
                <xs:attribute name="unit" type="xs:string"
                    use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="media">
    <xs:complexType>
        <xs:all>
            <xs:element ref="thumbnail"/>
            <xs:element ref="image"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="middle_name" type="xs:string"/>
<xs:element name="name_of_city" type="xs:string"/>
<xs:element name="name_of_country" type="xs:string"/>
<xs:element name="name_of_state" type="xs:string"/>
<xs:element name="number_of_pages" type="xs:short"/>
<xs:element name="phone_number" type="xs:string"/>
<xs:element name="pricing">
    <xs:complexType>
        <xs:all>
            <xs:element ref="suggested_retail_price"/>
            <xs:element ref="cost"/>
            <xs:element ref="when_is_available"/>
            <xs:element ref="quantity_in_stock"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="publisher">
    <xs:complexType>
        <xs:all>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="contact_information">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="mailing_address">
                            <xs:complexType>
                            <xs:all>
                            <xs:element ref="street_information"/>
                            <xs:element ref="name_of_city"/>
                            <xs:element ref="name_of_state"/>
```

```
                            <xs:element ref="zip_code"/>
                            <xs:element ref="country"/>
                            </xs:all>
                            </xs:complexType>
                        </xs:element>
                        <xs:element ref="FAX_number"
                            minOccurs="0"/>
                        <xs:element ref="phone_number"/>
                        <xs:element ref="web_site"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="quantity_in_stock" type="xs:byte"/>
<xs:element name="related_item">
    <xs:complexType>
        <xs:all>
            <xs:element ref="item_id"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="related_items">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="related_item" minOccurs="0"
                maxOccurs="5"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="size_of_book">
    <xs:complexType>
        <xs:all>
            <xs:element ref="length"/>
            <xs:element ref="width"/>
            <xs:element ref="height"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="street_address" type="xs:string"/>
<xs:element name="street_information">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="street_address" maxOccurs="2"/>
```

```
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="subject" type="xs:string"/>
    <xs:element name="suggested_retail_price">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:decimal">
                    <xs:attribute name="currency" type="xs:string"
                        use="required"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="thumbnail">
        <xs:complexType>
            <xs:all>
                <xs:element ref="data"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="type_of_book" type="xs:string"/>
    <xs:element name="when_is_available" type="xs:date"/>
    <xs:element name="width">
        <xs:complexType>
            <xs:simpleContent>
                <xs:extension base="xs:decimal">
                    <xs:attribute name="unit" type="xs:string"
                        use="required"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
    </xs:element>
    <xs:element name="zip_code" type="xs:string"/>
</xs:schema>
```

D.4 DC/MD

D.4.1 OrderXXX

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="authorization_date" type="xs:date"/>
    <xs:element name="authorization_id" type="xs:string"/>
    <xs:element name="bill_address_id" type="xs:short"/>
    <xs:element name="credit_card_number" type="xs:long"/>
    <xs:element name="credit_card_transaction">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="credit_card_type"/>
                <xs:element ref="credit_card_number"/>
                <xs:element ref="name_on_credit_card"/>
                <xs:element ref="expiration_date"/>
                <xs:element ref="authorization_id"/>
                <xs:element ref="transaction_amount"/>
                <xs:element ref="authorization_date"/>
                <xs:element ref="transaction_country_id"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="credit_card_type" type="xs:string"/>
    <xs:element name="customer_id" type="xs:int"/>
    <xs:element name="discount_rate" type="xs:decimal"/>
    <xs:element name="expiration_date" type="xs:date"/>
    <xs:element name="item_id" type="xs:int"/>
    <xs:element name="name_on_credit_card" type="xs:string"/>
    <xs:element name="order">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="customer_id"/>
                <xs:element ref="order_date"/>
                <xs:element ref="subtotal"/>
                <xs:element ref="tax"/>
                <xs:element ref="total"/>
                <xs:element ref="ship_type"/>
                <xs:element ref="ship_date"/>
                <xs:element ref="bill_address_id"/>
                <xs:element ref="ship_address_id"/>
                <xs:element ref="order_status"/>
                <xs:element ref="credit_card_transaction"/>
```

```
                <xs:element ref="order_lines"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:int" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="order_date" type="xs:date"/>
    <xs:element name="order_line">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="item_id"/>
                <xs:element ref="quantity_of_item"/>
                <xs:element ref="discount_rate"/>
                <xs:element ref="special_instructions"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:byte" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="order_lines">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="order_line" maxOccurs="5"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="order_status" type="xs:string"/>
    <xs:element name="quantity_of_item" type="xs:short"/>
    <xs:element name="ship_address_id" type="xs:int"/>
    <xs:element name="ship_date" type="xs:date"/>
    <xs:element name="ship_type" type="xs:string"/>
    <xs:element name="special_instructions" type="xs:string"/>
    <xs:element name="subtotal" type="xs:decimal"/>
    <xs:element name="tax" type="xs:decimal"/>
    <xs:element name="total" type="xs:decimal"/>
    <xs:element name="transaction_amount" type="xs:decimal"/>
    <xs:element name="transaction_country_id" type="xs:byte"/>
</xs:schema>
```

D.4.2 Customer

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="YTD_payment" type="xs:decimal"/>
    <xs:element name="address_id" type="xs:long"/>
    <xs:element name="balance" type="xs:decimal"/>
    <xs:element name="birth_date" type="xs:date"/>
    <xs:element name="current_session_expiry" type="xs:dateTime"/>
    <xs:element name="customer">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="user_name"/>
                <xs:element ref="password"/>
                <xs:element ref="first_name"/>
                <xs:element ref="last_name"/>
                <xs:element ref="address_id"/>
                <xs:element ref="phone_number"/>
                <xs:element ref="email_address"/>
                <xs:element ref="date_of_registration"/>
                <xs:element ref="date_of_last_visit"/>
                <xs:element ref="start_of_current_session"/>
                <xs:element ref="current_session_expiry"/>
                <xs:element ref="discount_rate"/>
                <xs:element ref="balance"/>
                <xs:element ref="YTD_payment"/>
                <xs:element ref="birth_date"/>
                <xs:element ref="miscellaneous_information"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:long" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="customers">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="customer" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="date_of_last_visit" type="xs:date"/>
    <xs:element name="date_of_registration" type="xs:date"/>
    <xs:element name="discount_rate" type="xs:decimal"/>
    <xs:element name="email_address" type="xs:string"/>
    <xs:element name="first_name" type="xs:string"/>
```

```
    <xs:element name="last_name" type="xs:string"/>
    <xs:element name="miscellaneous_information" type="xs:string"/>
    <xs:element name="password" type="xs:string"/>
    <xs:element name="phone_number" type="xs:long"/>
    <xs:element name="start_of_current_session" type="xs:dateTime"/>
    <xs:element name="user_name" type="xs:string"/>
</xs:schema>
```

D.4.3 Item

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="ISBN" type="xs:string"/>
    <xs:element name="author_id" type="xs:long"/>
    <xs:element name="cost" type="xs:decimal"/>
    <xs:element name="date_of_release" type="xs:date"/>
    <xs:element name="description" type="xs:string"/>
    <xs:element name="image" type="xs:string"/>
    <xs:element name="item">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="title"/>
                <xs:element ref="author_id"/>
                <xs:element ref="date_of_release"/>
                <xs:element ref="name_of_publisher"/>
                <xs:element ref="subject"/>
                <xs:element ref="description"/>
                <xs:element ref="related_item_id" maxOccurs="5"
                    minOccurs="5"/>
                <xs:element ref="thumbnail"/>
                <xs:element ref="image"/>
                <xs:element ref="suggested_retail_price"/>
                <xs:element ref="cost"/>
                <xs:element ref="when_is_available"/>
                <xs:element ref="quantity_in_stock"/>
                <xs:element ref="ISBN"/>
                <xs:element ref="number_of_pages"/>
                <xs:element ref="type_of_book"/>
                <xs:element ref="size_of_book"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:long" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="items">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="item" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="name_of_publisher" type="xs:string"/>
    <xs:element name="number_of_pages" type="xs:short"/>
```

```
    <xs:element name="quantity_in_stock" type="xs:byte"/>
    <xs:element name="related_item_id" type="xs:int"/>
    <xs:element name="size_of_book" type="xs:string"/>
    <xs:element name="subject" type="xs:string"/>
    <xs:element name="suggested_retail_price" type="xs:decimal"/>
    <xs:element name="thumbnail" type="xs:string"/>
    <xs:element name="title" type="xs:string"/>
    <xs:element name="type_of_book" type="xs:string"/>
    <xs:element name="when_is_available" type="xs:date"/>
</xs:schema>
```

D.4.4 Author

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="author">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="first_name"/>
                <xs:element ref="middle_name"/>
                <xs:element ref="last_name"/>
                <xs:element ref="date_of_birth"/>
                <xs:element ref="biography"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:long" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="authors">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="author" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="biography" type="xs:string"/>
    <xs:element name="date_of_birth" type="xs:date"/>
    <xs:element name="first_name" type="xs:string"/>
    <xs:element name="last_name" type="xs:string"/>
    <xs:element name="middle_name" type="xs:string"/>
</xs:schema>
```

D.4.5 Address

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="address">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="street_address" maxOccurs="2"
                    minOccurs="2"/>
                <xs:element ref="name_of_city"/>
                <xs:element ref="name_of_state"/>
                <xs:element ref="zip_code"/>
                <xs:element ref="country_id"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:long" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="addresses">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="address" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="country_id" type="xs:int"/>
    <xs:element name="name_of_city" type="xs:string"/>
    <xs:element name="name_of_state" type="xs:string"/>
    <xs:element name="street_address" type="xs:string"/>
    <xs:element name="zip_code" type="xs:string"/>
</xs:schema>
```

D.4.6 Country

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <xs:element name="countries">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="country" maxOccurs="92"
                    minOccurs="92"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="country">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="name"/>
                <xs:element ref="exchange_rate"/>
                <xs:element ref="currency"/>
            </xs:sequence>
            <xs:attribute name="id" type="xs:long" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="currency" type="xs:string"/>
    <xs:element name="exchange_rate">
        <xs:simpleType>
            <xs:restriction base="xs:decimal">
                <xs:enumeration value="7.75473"/>
                <xs:enumeration value="8.278"/>
                <xs:enumeration value="8.54477"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="name" type="xs:string"/>
</xs:schema>
```

Appendix E

DTDs for Document Classes

E.1 TC/SD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT a (#PCDATA)>
<!ELEMENT bib (#PCDATA)>
<!ELEMENT cr (#PCDATA)>
<!ELEMENT def (#PCDATA | cr)*>
<!ELEMENT dictionary (e+)>
<!ELEMENT e (hwg, vfl?, et?, ss)>
<!ATTLIST e
    id ID #IMPLIED
>
<!ELEMENT et (cr)+>
<!ELEMENT hw (#PCDATA)>
<!ELEMENT hwg (hw | pr? | pos?)+>
<!ELEMENT loc (#PCDATA)>
<!ELEMENT pos (#PCDATA)>
<!ELEMENT pr (#PCDATA)>
<!ELEMENT q (qd, a?, w, bib?, loc, qt)>
<!ELEMENT qd (#PCDATA)>
<!ELEMENT qp (q+)>
<!ELEMENT qt (#PCDATA | cr | i | b)*>
<!ELEMENT s (def, qp)>
<!ELEMENT ss (s+)>
<!ELEMENT vd (#PCDATA)>
<!ELEMENT vf (#PCDATA)>
<!ELEMENT vfl (vd* | vf+)+>
<!ELEMENT w (#PCDATA)>
<!ELEMENT i (#PCDATA)>
<!ELEMENT b (#PCDATA)>
```

E.2 TC/MD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT a_id (#PCDATA)>
<!ELEMENT abstract (p+)>
<!ELEMENT article (prolog, body, epilog)>
<!ATTLIST article
    id CDATA #REQUIRED
    lang CDATA #REQUIRED
>
<!ELEMENT author (name, contact)>
<!ELEMENT authors (author+)>
<!ELEMENT body (abstract?, section+)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT contact (email?, phone?)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT dateline (city, country, date)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT keyword (#PCDATA)>
<!ELEMENT keywords (keyword+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT p (#PCDATA)>
<!ELEMENT pa (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT prolog (title, authors?, dateline?, genre?, keywords?)>
<!ELEMENT section (p*, subsec*)>
<!ATTLIST section
    heading CDATA #REQUIRED
>
<!ELEMENT subsec (p*, subsec*)>
<!ATTLIST subsec
    heading CDATA #REQUIRED
>
<!ELEMENT title (#PCDATA)>
<!ELEMENT epilog (acknowledgements?, references?)>
<!ELEMENT acknowledgements (pa+)>
<!ELEMENT references (a_id+)>
```

E.3 DC/SD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT FAX_number (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT attributes (ISBN, number_of_pages, type_of_book, size_of_book)>
<!ELEMENT author (name, date_of_birth, biography, contact_information)>
<!ELEMENT authors (author+)>
<!ELEMENT biography (#PCDATA)>
<!ELEMENT catalog (item+)>
<!ATTLIST catalog
    xmlns:xsi CDATA #REQUIRED
    xsi:noNamespaceSchemaLocation CDATA #REQUIRED
>
<!ELEMENT contact_information (mailing_address, FAX_number?,
    phone_number, email_address?, web_site?)>
<!ELEMENT cost (#PCDATA)>
<!ATTLIST cost
    currency CDATA #REQUIRED
>
<!ELEMENT country (name, exchange_rate, currency)>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT data (#PCDATA)>
<!ELEMENT date_of_birth (#PCDATA)>
<!ELEMENT date_of_release (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT email_address (#PCDATA)>
<!ELEMENT web_site (#PCDATA)>
<!ELEMENT exchange_rate (#PCDATA)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT height (#PCDATA)>
<!ATTLIST height
    unit CDATA #REQUIRED
>
<!ELEMENT image (data)>
<!ELEMENT item (title, authors, date_of_release, publisher,
    subject, description, related_items, media, pricing, attributes)>
<!ATTLIST item
    id ID #REQUIRED
>
<!ELEMENT item_id (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT length (#PCDATA)>
<!ATTLIST length
    unit CDATA #REQUIRED
```

```
>
<!ELEMENT mailing_address (street_information, name_of_city,
    name_of_state, zip_code, name_of_country?, country?)>
<!ELEMENT media (thumbnail, image)>
<!ELEMENT middle_name (#PCDATA)>
<!ELEMENT name (#PCDATA | first_name | middle_name | last_name)*>
<!ELEMENT name_of_city (#PCDATA)>
<!ELEMENT name_of_country (#PCDATA)>
<!ELEMENT name_of_state (#PCDATA)>
<!ELEMENT number_of_pages (#PCDATA)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT pricing (suggested_retail_price, cost, when_is_available,
    quantity_in_stock)>
<!ELEMENT publisher (name, contact_information)>
<!ELEMENT quantity_in_stock (#PCDATA)>
<!ELEMENT related_item (item_id)>
<!ELEMENT related_items (related_item+)>
<!ELEMENT size_of_book (length, width, height)>
<!ELEMENT street_address (#PCDATA)>
<!ELEMENT street_information (street_address+)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT suggested_retail_price (#PCDATA)>
<!ATTLIST suggested_retail_price
    currency CDATA #REQUIRED
>
<!ELEMENT thumbnail (data)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT type_of_book (#PCDATA)>
<!ELEMENT when_is_available (#PCDATA)>
<!ELEMENT width (#PCDATA)>
<!ATTLIST width
    unit CDATA #REQUIRED
>
<!ELEMENT zip_code (#PCDATA)>
```

E.4 DC/MD

E.4.1 Order

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT authorization_date (#PCDATA)>
<!ELEMENT authorization_id (#PCDATA)>
<!ELEMENT bill_address_id (#PCDATA)>
<!ELEMENT credit_card_number (#PCDATA)>
<!ELEMENT credit_card_transaction (credit_card_type,
    credit_card_number, name_on_credit_card, expiration_date,
    authorization_id, transaction_amount, authorization_date,
    transaction_country_id)>
<!ELEMENT credit_card_type (#PCDATA)>
<!ELEMENT customer_id (#PCDATA)>
<!ELEMENT discount_rate (#PCDATA)>
<!ELEMENT expiration_date (#PCDATA)>
<!ELEMENT item_id (#PCDATA)>
<!ELEMENT name_on_credit_card (#PCDATA)>
<!ELEMENT order (customer_id, order_date, subtotal, tax, total,
    ship_type, ship_date, bill_address_id, ship_address_id,
    order_status, credit_card_transaction, order_lines)>
<!ATTLIST order
    id CDATA #REQUIRED
>
<!ELEMENT order_date (#PCDATA)>
<!ELEMENT order_line (item_id, quantity_of_item, discount_rate,
    special_instructions)>
<!ATTLIST order_line
    id CDATA #REQUIRED
>
<!ELEMENT order_lines (order_line+)>
<!ELEMENT order_status (#PCDATA)>
<!ELEMENT quantity_of_item (#PCDATA)>
<!ELEMENT ship_address_id (#PCDATA)>
<!ELEMENT ship_date (#PCDATA)>
<!ELEMENT ship_type (#PCDATA)>
<!ELEMENT special_instructions (#PCDATA)>
<!ELEMENT subtotal (#PCDATA)>
<!ELEMENT tax (#PCDATA)>
<!ELEMENT total (#PCDATA)>
<!ELEMENT transaction_amount (#PCDATA)>
<!ELEMENT transaction_country_id (#PCDATA)>
```

E.4.2 Customer

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT YTD_payment (#PCDATA)>
<!ELEMENT address_id (#PCDATA)>
<!ELEMENT balance (#PCDATA)>
<!ELEMENT birth_date (#PCDATA)>
<!ELEMENT current_session_expiry (#PCDATA)>
<!ELEMENT customer (user_name, password, first_name,
    last_name, address_id, phone_number, email_address,
    date_of_registration, date_of_last_visit,
    start_of_current_session, current_session_expiry,
    discount_rate, balance, YTD_payment, birth_date,
    miscellaneous_information)>
<!ATTLIST customer
    id CDATA #REQUIRED
>
<!ELEMENT customers (customer+)>
<!ELEMENT date_of_last_visit (#PCDATA)>
<!ELEMENT date_of_registration (#PCDATA)>
<!ELEMENT discount_rate (#PCDATA)>
<!ELEMENT email_address (#PCDATA)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT miscellaneous_information (#PCDATA)>
<!ELEMENT password (#PCDATA)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT start_of_current_session (#PCDATA)>
<!ELEMENT user_name (#PCDATA)>
```

E.4.3 Item

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT author_id (#PCDATA)>
<!ELEMENT cost (#PCDATA)>
<!ELEMENT date_of_release (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT image (#PCDATA)>
<!ELEMENT item (title, author_id, date_of_release,
    name_of_publisher, subject, description, related_item_id+,
    thumbnail, image, suggested_retail_price,
    cost, when_is_available, quantity_in_stock, ISBN,
    number_of_pages, type_of_book, size_of_book)>
<!ATTLIST item
    id CDATA #REQUIRED
>
<!ELEMENT items (item+)>
<!ELEMENT name_of_publisher (#PCDATA)>
<!ELEMENT number_of_pages (#PCDATA)>
<!ELEMENT quantity_in_stock (#PCDATA)>
<!ELEMENT related_item_id (#PCDATA)>
<!ELEMENT size_of_book (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT suggested_retail_price (#PCDATA)>
<!ELEMENT thumbnail (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT type_of_book (#PCDATA)>
<!ELEMENT when_is_available (#PCDATA)>
```

E.4.4 Author

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT author (first_name, middle_name, last_name,
    date_of_birth, biography)>
<!ATTLIST author
    id CDATA #REQUIRED
>
<!ELEMENT authors (author+)>
<!ELEMENT biography (#PCDATA)>
<!ELEMENT date_of_birth (#PCDATA)>
<!ELEMENT first_name (#PCDATA)>
<!ELEMENT last_name (#PCDATA)>
<!ELEMENT middle_name (#PCDATA)>
```

E.4.5 Address

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT address (street_address+, name_of_city,
    name_of_state, zip_code, country_id)>
<!ATTLIST address
    id CDATA #REQUIRED
>
<!ELEMENT addresses (address+)>
<!ELEMENT country_id (#PCDATA)>
<!ELEMENT name_of_city (#PCDATA)>
<!ELEMENT name_of_state (#PCDATA)>
<!ELEMENT street_address (#PCDATA)>
<!ELEMENT zip_code (#PCDATA)>
```

E.4.6 Country

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT countries (country+)>
<!ELEMENT country (name, exchange_rate, currency)>
<!ATTLIST country
    id CDATA #REQUIRED
>
<!ELEMENT currency (#PCDATA)>
<!ELEMENT exchange_rate (#PCDATA)>
<!ELEMENT name (#PCDATA)>
```