Biodiversity Informatics, 4, 2007, pp. 1-13

A QUANTITATIVE COMPARISON OF XML SCHEMAS FOR TAXONOMIC PUBLICATIONS

GUIDO SAUTTER[1,3], KLEMENS BÖHM[1], AND DONAT AGOSTI[2]

[1]*Department of Computer Science, Universität Karlsruhe (TH), 76128 Karlsruhe, Germany;*
[2] *Division of Invertebrate Zoology, American Museum of Natural History, New York NY 10024-5192, and Naturmuseum der Burgergemeinde Bern, 3005 Bern Switzerland;*
[3] sautter@ipd.uka.de

Abstract.— Large numbers of legacy taxonomic publications are currently being digitized to make them online available and ready for full text search. The documents are being marked up with XML for two purposes: To preserve the document structure, and to facilitate access via standard query languages like XQuery. With regard to the second aspect, the choice of an appropriate XML schema is crucial. It affects both query performance and the correctness of query results. Over the last few years, several different XML schemas have been proposed as markup standards for taxonomic publications. In this paper, we report on a thorough evaluation and comparison of these schemas. We have examined if they facilitate formulation and correct processing of queries that are common when it comes to taxonomic literature. We also compare the performance of these queries on documents that are marked up with the different schemas. Finally, we propose extensions to the schemas that enhance correctness of query results.

Key words. — Heritage literature, quantiative analysis, systematics, taxonomy, TaxonX, xml schema

At present, legacy taxonomic publications are being digitized in large numbers (e.g. Biodiversity Heritage Library[1]). The intention is to store these documents in digital archives and to make them available online. The documents are marked up with XML for two purposes. The first one is to preserve the original document structure and publication-related in-formation like publisher, title and issue. The sec-ond one is to facilitate deployment of standard query languages like XPath (XPath) to access the document collection. In the recent past, several institutions and projects have proposed a variety of XML schemas for this purpose, such as ABCD (ABCD), SDD (SDD), TaxonX (TaxonX), and taXMLit (Weitzman & Lyall). In this paper, we compare these schemas and include some additional ones. Our comparison focuses on the second aspect mentioned above, relevant to the work of biologists. The queries typical for this domain are fine-grained (at the level of individual characters or distribution records), and their results are individual treatments, i.e., descriptions of a particular taxon. We have identified three basic types of criteria on which queries can be based: Taxonomic names, the collection locations, i.e., the locations where specimens of a particular taxon have been collected, and the morphological feature concepts as the selection criterion. We investigate both the ease of formulation and the execution performance of these queries. Our results show that the four schemas mentioned above support queries over taxonomic names very well. The same is true for the collection locations. However, SDD is the only schema to allow formulating queries over morphological features at least to a certain degree, and they execute slowly in the environment investigated here. The other schemas do not provide any markup to identify individual concepts within morphologic descriptions. It is not possible to represent the relationship of a concept name and the associated descriptive terms. To overcome this problem, we propose some detail-level extensions to the different schemas. We show that the extended schemas better support queries that use morphological concepts as selection criteria. In our experiments, we have observed a moderate performance decrease due to the more complex markup.

The issues investigated here are orthogonal to the question how the markup should actually be created, be it automated, semi-automated or completely manual. Clearly, this question is important as well, and much research has

[1] http://www.bhl.si.edu/

addressed it, including our own (Sautter et al. 2006, Sautter et al. 2007). In this current study, we assume that the documents are already marked up.

We assume that the reader has some familiarity with XML (markup language for semi-structured data) (XML) and respective query languages (XPath, XQuery in particular, (XPath)).

QUERIES

In order to make maximum use of digital biosystematics archives, the information from the documents needs to be accessible on the treatment level. Figure 1 displays such a treatment, an excerpt from Wheeler (1922), which is also the basis for all examples throughout this paper. In the marked-up examples to follow, we will only use the passages printed in bold in Figure 1. They will be sufficient for our purposes.

Pheidole lacerta, new species
[...]
Mandibles and clypeus smooth and shining, the former with small scattered, elongate punctures. **Head and thorax subopaque**, the head transversely rugose above, more reticulate-rugose laterally and in the occipital region, the scrobes finely and densely punctate. The gula is also reticulate but more loosely and finely and its sides are smooth and shining. Thorax and petiole very finely and densely punctate, the pronotum also transversely rugulose above. Postpetiole, gaster and legs smooth and shining, with fine, sparse, piliferous punctures.
[...]
A single specimen swept from foliage near Port of Spain by Prof. Roland Thaxter.
[...]

Figure 1: The example treatment (from Wheeler 1922)

There are three important information needs in biosystematics, or, in other words, three different ways of searching for information. They differ in the search criteria, which may be taxonomic names, collecting locations, or observable morphological features. In this section, we describe these three basic types of queries in more detail. The query types are also the basis for our evaluation experiments. These queries are fine-grained, i.e., they return individual treatments rather than entire publications. After the definitions, we give examples in natural language. The search criteria of the example queries are in italics. We will refer to these examples throughout this paper.

Name Queries

Name queries find all information available on a given taxon. The selection criterion is the name of the taxon. The result is the set of all treatments on the particular taxon, i.e., descriptions as well as any collection event available. The latter serve to compute the dispersal of the taxon. The taxon range considered here is from genus down to variety.

The natural language formulation of a typical Name Query would be: “Find all treatments of a taxon with the given name. In particular, find the description and the dispersal”.

Example NQ: “Find the treatments on *Pheidole lacerta*.”

Location Queries

Location queries retrieve information on the fauna of a given location or area. The selection criterion is the name of a location or, ideally, longitude and latitude degrees. The result is the set of descriptions of all taxa that have ever been collected at this location, i.e., all taxa with a collection event which refers to it.

In natural language, a typical Location Query could be: “Find all taxa with a specimen having been collected at the given location”.

Example LQ: “Find all treatments on taxa endemic in *Port of Spain*”.

Concept Queries

Concept queries identify a taxon based on its morphological feature concepts. The results sought are treatments on taxa with these concepts, in particular the treatments with a description.

A Concept Query is as follows in natural language: “Find all treatments describing a taxon showing a given morphological feature concept”. – Note that our example treatment (see Figure 1) will match both Example CQ1 and Example CQ2, but it does not match Example CQ3. The latter is because in the example text, shining refers to mandibles and clypheus, but not to head.

Example CQ1: “Find the treatments on all taxa whose *head* is *subopaque*”.

Example CQ2: “Find the treatments on all taxa whose *thorax* is *subopaque*”.

Example CQ3: “Find the treatments on all taxa whose *head* is *shining*”.

XML SCHEMAS

In this section, we briefly introduce and discuss the different existing XML schemas that have been proposed for taxonomic publications. To our knowledge, no other relevant schemas exist. The second topic of this section is how different types of queries can be formulated (see previous section) based on the various schemas, and its performance estimations regarding query execution.

Due to space limitations, we restrict the examples to the treatment-internal markup, i.e., we do not provide the structural markup that delimits the treatments in the document. Consequently, we also omit the markup of document meta-data, e.g., title, author(s), etc. in the mods elements of TaxonX (TaxonX). This applies to the markup examples as well as to the XPath expressions. In addition, we only include those XML elements in the markup examples that are relevant for the processing of the queries defined in the previous section. The markup examples are all based on the treatment shown in Figure 1.

ABCD

A design objective behind the ABCD schema (ABCD) is the preservation of the original document structure. ABCD includes very detailed markup of publication-related data such as authors or publishers. Treatments are marked up as Units in this schema. Figure 2 displays the example treatment marked up according to ABCD.

```
  <Unit>
  <UnitID/>
  <Identifications>
    <Identification>
      <TaxonIdentified>
        <ScientificNameString>
          Pheidole lacerta, new species
        </ScientificNameString>
      </TaxonIdentified>
    </Identification>
  </Identifications>
  <UnitDescription Language="en">
    Mandibles and clypeus smooth and shining.
    Head and thorax sub-opaque.
  </UnitDescription>
  <Gathring>
    A single specimen swept from foliage near
    <GatheringSite>
  <LocalityText>Port of Spain</LocalityText>
</GatheringSite> by Prof. Roland Thaxter.
  </Gathring>
</Unit>
```

Figure 2: Example treatment marked up in ABCD.

For a document marked up in ABCD, the three types of queries can be formulated as the following XPath expressions:

Name Query (Example NQ):

```
Unit[contains(./Identifications/Identification/TaxonIdentified
/ ScientificNameString, “Pheidole lactera”)]
```

Location Query (Example LQ):

```
Unit[./Gathring/GatheringSite/LocalityText = “Port of
Spain”]
```

Concept Query (Example CQ1):

```
Unit[contains(./UnitDescription, “Head”) and
contains(./UnitDescription, “sub-opaque”)]
```

The ABCD schema supports the Name Queries and Location Queries very well. According to Altinel and Franklin (2000), the markup depth of a document affects query performance. Therefore, we expect the Name Queries to execute quite slowly because of the deep nesting of the taxon name – there are five hierarchy levels Unit, Identifications, Identification, TaxonIdentified, and ScientificNameString. The Concept Queries cannot be formulated precise enough – we would obtain incorrect results. The reason is that ABCD markup does not make the relationship of the name of a morphological feature and its description explicit. For instance, the markup example would also match the query below, which formulates Example CQ3 as exact as possible, even though ‘shining’ is not part of the description of the morphological feature ‘head’.

```
Unit[contains(./UnitDescription, “Head”) and
contains(./UnitDescription, “shining”)]
```

SDD / UBIF

The SDD schema (SDD) provides detailed markup for textual descriptions. It imports the UBIF schema (UBIF). The combination of both provides detailed markup for document-related as well as for datacentric aspects. Figure 3 shows the example treatment marked up according to SDD.

The key idea of this schema is that data on different aspects (taxon name, locations, morphologic feature concepts, textual description) is represented separately and linked by ref attributes: The taxonomic name is contained in a ClassName element. A Geography element marks up the collection locations. The description is wrapped in a DescriptiveData element, which has two children among others: A Terminology element (transitively) contains Concept elements, which

represent the individual morphological features the description refers to. They have an id attribute identifying them. The treatment text itself is enclosed in a NaturalLanguageDescription element. Concept elements enclose the individual sentences of the description. These Concept elements are linked to the ones in Terminology with id-ref references. In particular, a Concept in the NaturalLanguageDescription having a certain value as its ref attribute refers to the Concept in the Terminology whose id attribute has the same value.

```
<Dataset>
<ExternalDataInterface>
  <ClassNames>
    <ClassName id="cl-1">
      <Label>
        <Representation language="en">
          <Text>Pheidole lacerta</Text>
        </Representation>
      </Label>
    </ClassName>
  </ClassNames>
  <Geography>
    <Locality id="l-1">
      <Label>Port of Spain</Label>
    </Locality>
  </Geography>
</ExternalDataInterface>
<DescriptiveData>
  <Terminology>
    <ConceptTrees>
      <ConceptTree id="ct-1">
        <Concept id="c-1">
          <Nodes>
            <Concept id="c-11">
              <Label>
                <Representation language="en">
                  <Text>HEAD</Text>
                </Representation>
              </Label>
            </Concept>
            <Concept id="c-12">
              <Label>
                <Representation language="en">
                  <Text>MANDIBLES</Text>
                </Representation>
              </Label>
            </Concept>
          </Nodes>
        </Concept>
      </ConceptTree>
      <ConceptTree id="ct-2">
        <Label>
          <Representation language="en">
            <Text>Methodology</Text>
          </Representation>
        </Label>
        <Concept id="c-2">
          <Nodes>
            <Concept id="c-21">
              <Label>
                <Representation language="en">
                  <Text>
                  MATERIALS EXAMINED
                  </Text>
                </Representation>
              </Label>
            </Concept>
          </Nodes>
        </Concept>
      </ConceptTree>
    </ConceptTrees>
  </Terminology>
  <NaturalLanguageDescriptions>
    <NaturalLanguageDescription id="nld-1">
      <Header>
        <ClassName ref="cl-1"/>
      </Header>
      <NaturalLanguageData>
        <Concept ref="c-12">
          <Text>
          Mandibles and clypeus smooth & shining
          </Text>
        </Concept>
        <Concept ref="c-11">
          <Text>
            Head and thorax subopaque.
          </Text>
        </Concept>
        <Concept ref="c-21">
          <Text>
            A single specimen swept from foliage
            near Port of Spain by Prof. R. Thaxter.
          </Text>
        </Concept>
      </NaturalLanguageData>
    </NaturalLanguageDescription>
  </NaturalLanguageDescriptions>
</DescriptiveData>
</Dataset>
```

Figure 3: Example treatment marked up in SDD / UBIF

For a document marked up in SDD / UBIF, the three types of queries can be formulated as the following XPath expressions:

Name Query (Example NQ):

```
Dataset[./ExternalDataInterface/ClassNames/
ClassName/Label/Representation/Text = “Pheidole
lactera”]
```

Location Query (Example LQ):

```
Dataset[./ExternalDataInterface/Geography/Locality/
Label/Representation/Text = “Port of Spain”]
```

Concept Query (Example CQ1):

```
Dataset[DescriptiveData/Terminology/ConceptTrees/
ConceptTree//Concept[.//Label/Representation/Text =
"HEAD" and ./@id = string(//NaturalLanguageData/
Concept[contains(./Text, "subopaque")]/@ref)]]
```

The SDD / UBIF schema supports the Name Queries and Location Queries. But according to Altinel & Franklin (2000), we can expect both to execute slowly because of the deep nesting of the taxon name and the locations. The Concept Queries can be formulated in a way that we expect to yield correct results. This is an advantage over the other schemas considered. But we expect these queries to execute very slowly due to the id references. Dereferencing them results in a join, which is very complex and time-intensive to evaluate (Wu et al. 2003). For large documents, in-memory query processing is not feasible, and the query engine has to perform file-based query evaluation. If it builds an index over the ids while performing the scan, all the elements have to remain in

memory. This is likely to result in out-of-memory problems (Ives et al. 2000). To avoid this, the query engine has to scan the document twice: Once for collecting the referenced ids, and a second time to find the corresponding elements. This additional scan roughly doubles query-execution time. Scanning the file takes the largest share of time in file-based query evaluation (Böhm 2000).

An additional problem is that a concept element in the natural language description can refer to only one concept in terminology. Our fragment of the sample document matches Example CQ2 ("Find the treatments on all taxa whose thorax is subopaque."), but the Concept Query from above formulated against the SDD / UBIF schema, with 'HEAD' replaced with 'thorax', will not return a hit. An alternative formulation of Example CQ2 can overcome this problem, but would reduce the explicit modeling of the Terminology element to absurdity:

Dataset[DescriptiveData/NaturalLanguageDescriptions/ NaturalLanguageDescription/NaturalLanguageData/ Concept/Text[contains(., "thorax") and contains(., "subopaque"]]

Finally, the original document is hard to reproduce once it has been transferred to an SDD / UBIF representation.

TaxonX

The TaxonX schema (TaxonX) preserves part of the original document structure. In particular, it provides paragraphs and different types of divisions. The individual treatments are enclosed in a <treatment> tag. Figure 4 displays the example treatment marked up according to TaxonX.

```
<treatment>
  <nomenclature>
    <name>Pheidole lacerta</name>
    <status>new species</status>
  </nomenclature>
  <div type="description">
    <p>
      Mandibles and clypeus smooth and shining.
      Head and thorax sub-opaque.
    </p>
  </div>
  <div type="materials_examined">
    <p>
      <seg type="materials_examined">
        A single specimen swept from foliage near
        <collection_event>
          <locality>Port of Spain</locality>
        </collection_event>
        by Prof. Roland Thaxter.
      </seg>
    </p>
  </div>
</treatment>
```

Figure 4: Example treatment marked up in TaxonX

For a document marked up in TaxonX, one can formulate the three types of queries as the following XPath expressions:

Name Query (Example NQ):

treatment[./nomenclature/name = "Pheidole lactera"]

Location Query (Example LQ):

treatment[./div[./@type = "materials_examined"] /p/seg/collection_event/locality = "Port of Spain"]

Concept Query (Example CQ1):

treatment[./div[./@type = "description"] /p[contains(./text(), "Head") and contains(./text(), "sub-opaque")]]

We found the Name Queries and Location Queries easy to formulate on TaxonX. We expect the Name Queries to execute fast. The Location Queries, on the other hand, are likely to execute slowly because of the deep nesting in div, p, seg, collection_event, and locality tags. The Concept Queries can be formulated, but we expect incorrect results, for the same reason as with ABCD. The example query (producing incorrect results) is as follows for TaxonX:

treatment[./div[./@type = "description"] /p[contains(./text(), "Head") and contains(./text(), "shining")]]

taXMLit

The taXMLit schema (Weitzman et al.) provides very detailed markup on different levels, document-centric as well as data-centric. Its nesting is very deep: All textual content is wrapped in at least five hierarchy elements. It is somewhat cumbersome to formulate the long queries necessary for drilling down into the hierarchy. In addition, the documents are not exactly human-readable. The markup is relatively heavy, compared to the other schemas.

For a document marked up in taXMLit, the three types of queries can be formulated as the following XPath expressions:

Name Query (Example NQ):

TaxonTreatment[./TaxonHeading/TaxonHeadingName/ TaxonName/TaxonNameText = "Pheidole lactera"]

Location Query (Example LQ):

TaxonTreatment [./DistributionAndOrSpecimenCitations/ IndividualLocalities/Locality/DetailedLocation/ DetailedLocationText = "Port of Spain"]

Concept Query (Example CQ1):

TaxonTreatment[./Descriptions/ SameLanguageDescription/ SameLanguageDescriptonParagraphs/ SameLanguageDescriptionParagraph/ SameLanguageDescriptionText[contains(./text(), "Head") and contains(./text(), "sub-opaque")]

The taXMLit schema formally supports the Name Queries and Location Queries. But we again expect both to execute slowly because of the deep nesting. The Concept Queries can be formulated, but we expect incorrect results, for the same reasons as with ABCD and TaxonX. With taXMLit, Example CQ3 in XPath looks like this:

```
TaxonTreatment[./Descriptions/
SameLanguageDescription/
SameLanguageDescriptonParagraphs/
SameLanguageDescriptionParagraph/
SameLanguageDescriptionText[contains(./text(), “Head”)
and contains(./text(), “shining”)]
```

```
<TaxonTreatment TreatmentLanguage="English"
RecognizedInTreatment="true" TaxonID="tt-1">
  <TreatmentAuthors>...</TreatmentAuthors>
  <TaxonHeading>
    <TaxonHeadingBody ElementID="thb-1"
Explicit="true">
      <TaxonHeadingText>
        Pheidole lacerta, new species
      </TaxonHeadingText>
    </TaxonHeadingBody>
    <RankDesignation Explicit="false">
      <RankDesignationText>
        species
      </RankDesignationText>
    </RankDesignation>
    <TaxonHeadingName>
      <TaxonName InformalName="false">
        <TaxonNameText>
          Pheidole lacerta
        </TaxonNameText>
      </TaxonName>
    </TaxonHeadingName>
  </TaxonHeading>
  <Descriptions>
    <SameLanguageDescription>
      <SameLanguageDescriptonParagraphs>
        <SameLanguageDescriptionParagraph
          ElementID="sldp-1">
          <SameLanguageDescriptionText>
            Mandibles and clypeus smooth and shining.
            Head and thorax sub-opaque.
          </SameLanguageDescriptionText>
        </SameLanguageDescriptionParagraph>
      </SameLanguageDescriptonParagraphs>
    </SameLanguageDescription>
  </Descriptions>
  <DistributionAndOrSpecimenCitations>
    <DistributionAndOrSpecimenParagraph
ElementID="daosp-1">
      <DistributionAndOrSpecimenParagraphText>
        A single specimen swept from foliage near
        Port of Spain by Prof. Roland Thaxter.
      </DistributionAndOrSpecimenParagraphText>
    </DistributionAndOrSpecimenParagraph>
    <IndividualLocalities>
      <Locality>
        <DetailedLocation>
          <DetailedLocationText>
            Port of Spain
          </DetailedLocationText>
        </DetailedLocation>
      </Locality>
    </IndividualLocalities>
  </DistributionAndOrSpecimenCitations>
</TaxonTreatment>
```

Figure 5: Example treatment marked up in taXMLit

TCS

The Taxonomic Concept Schema (TCS) is intended for the transfer of taxonomic data rather than for document markup. This means that the tags provided by TCS are not designed for sophisticated search and querying of treatments, but for transferring individual concepts between applications or machines. Consequently, we do not include it in our evaluation of XML schemas for the markup of taxonomic publications.

LinneanCore

The LinneanCore schema (LinneanCore) provides very detailed markup for taxonomic names. It is intended for representing these names rather than for the markup of entire documents. Thus, there are no elements for geographical information or textual descriptions of a taxon. Hence, we do not include the LinnaeanCore schema in our evaluation either.

Natural Collections Descriptions

The Natural Collections Description schema (NCD) is currently being developed. Its intention is the description of specimen collections rather than the markup of taxonomic publications. Consequently, we do not include the Natural Collections Descriptions schema in our evaluation.

DarwinCore 2

The DarwinCore 2 schema (DarwinCore2) provides detailed markup for the key parts of taxonomic names and collection events. All these elements reside in a simple list element, thus the nesting of this schema is very flat. It may well be applicable for the representation of individual collection events. But it is not applicable for the markup of publications, for two reasons: First, it does not provide elements for the markup of descriptions. Second, the schema provides no root element to enclose the entire document, which may well contain more than one treatment. Consequently, we do not include the DarwinCore 2 schema in our evaluation of XML schemas for the markup of taxonomic publications.

Conclusion of Schema Analysis

Four of the eight schemas (ABCD, taXMLit, TaxonX and SDD / UBIF) presented in this section are intended for the markup of taxonomic publications. To keep this paper focused, we will restrict further considerations

and our evaluation to these four schemas. All the four schemas support the Name Queries and Location Queries well, but we expect different query execution costs for the various schemas because of different nesting depths. None of them supports the Concept Queries in a way that all morphological features can be queried, and that the results are correct.

ABCD, TaxonX and taXMLit are rather document-centric (Nambiar et al. 2000). They focus on the structure and text of the document and do not rearrange the content. The three schemas do not provide markup for individual morphological concepts. SDD / UBIF in turn clearly is data-centric, i.e., it organizes the content of a document with the focus on its semantics, and the original structure is (more or less) lost. On the other hand, SDD / UBIF is the only schema that, at least to a certain degree, supports Concept Queries in a way so that we can expect correct results.

We expect the Name Queries and Location Queries to execute fast with ABCD and TaxonX due to the relative simplicity of these two schemas. The taXMLit schema in turn provides a very complex element nesting. We expect this to result in a decrease of query performance. That complexity seems to be unnecessary from our specific perspective because it provides no advantages over the first two schemas from the querying point of view. All three schemas share the problem that Concept Queries cannot be formulated sufficiently exactly.

The SDD / UBIF schema supports all three types of queries, but we expect all of them to execute rather slowly. This is because resolving the id references results in higher query complexity. This in turn decreases query performance.

POSSIBLE EXTENSIONS

As the previous section has shown, only the SDD / UBIF schema allows formulating the Concept Queries in a way so that we can expect correct results. This is because the other schemas are too document-centric. In particular, there is no way of expressing the relation of a morphological feature name and the associated description. Even the level SDD / UBIF provides is insufficient: Morphological features can be queried exactly only if the Terminology part provides a corresponding concept. Otherwise, the problems are the same as those of the other schemas.

To support the Concept Queries better, a marked-up document should allow querying individual morphological feature concepts. In this section, we propose and discuss three options to mark up the descriptions on a lower level in a more data-centric fashion to achieve this goal. These different types of detail-level markup can be added seamlessly to the schemas as children of the paragraph or treatment elements. The latter are part of all of the schemas. In other words, our proposed extensions are independent of a particular schema.

Morphologic Indexing

Extending the concept element of SDD / UBIF, we create a morphologic index. This index contains all morphological feature names contained in the description, and all descriptive terms for each of them. A morphological feature concept (Concept) contains the feature name (Label) and a list of the descriptive terms assigned to the name (Description). Figure 6 displays the morphologic index for our example treatment. Due to space limitations, we restrict the figure to the Terminology part of the Dataset. In particular, we omit the ExternalDataInterface and NaturalLanguageDescription parts and the Representation tags. – This highly data-centric idea results in the following dilemma: If the original document structure is to be preserved, the morphologic index contains much redundant data. If redundancy is not desirable, on the other hand, the original document structure cannot be preserved.

```
  <Terminology>
  <ConceptTrees>
    <ConceptTree id="ct-1">
      <Concept>
        <Label>HEAD</Label>
        <Description>subopaque</Description>
      </Concept>
      <Concept>
        <Label>THORAX</Label>
        <Description>subopaque</Description>
      </Concept>
      <Concept>
        <Label>MANDIBLES</Label>
        <Description>smooth shining</Description>
      </Concept>
      <Concept>
        <Label>CLYPHEUS</Label>
        <Description>smooth shining</Description>
      </Concept>
    </ConceptTree>
  </ConceptTrees>
</Terminology>
```

Figure 6: Morphologic Index

As a consequence of this extension, our example Concept Query for SDD is now significantly less complex than the original one. This is because the extension allows formulating the query without a join. For the other three schemas, only this extension enables formulating the Concept Queries in a way similar to SDD, and thus obtaining correct results. Example CQ1 now formulates like this:

```
Dataset[DescriptiveData/Terminology/ConceptTrees/
ConceptTree//Concept[./Label = "HEAD" and
contains(./Description, "subopaque")]]
```

We expect this query to execute significantly faster than the original one. On the other hand, we also expect the Name Queries and Location Queries to perform a little worse. The reason is the increased document size and the additional XML elements induced by the redundant data.

Aspect Markup

If the original document structure is important, and redundancy is not desirable, we have to use a more document-centric approach to mark up the descriptions. The idea is to use in-line tags for partitioning the description paragraphs into individual description aspects. Such an aspect consists of a (set of) descriptive term(s) and the name(s) of the morphological features they refer to. In our example document, each sentence of the description is an aspect. The feature elements identify the names of the morphological features that are subject to the description. This fine-grained markup allows a sufficiently exact formulation of the Concept Queries. Using the TaxonX example as the basis, (a fragment of) our example document becomes the one in Figure 7.

```
<div type="description">
<p>
  <aspect>
    <feature>Mandibles</feature> and
    <feature>clypeus</feature> smooth and shining
  </aspect>,
  <aspect>
    <feature>Head</feature> and
    <feature>thorax</feature> sub-opaque
  </aspect>.
</p></div>
```

Figure 7: Morphologic description with individual aspects marked up.

Individually marking up the feature names is not necessary in this example. This is because the description applies to all feature names in a sentence. In more complex documents with longer sentences, however, such markup is necessary in order to distinguish the described features from the ones that are part of descriptive terms. Consider the aspect in Figure 8. The descriptive terms refer only to “mandibles” and “clypheus”. “hairs” is not a feature described. Without the features marked up explicitly, such differentiations would not be possible in a query.

```
<aspect>
<feature>Mandibles</feature> and
<feature>clypeus</feature>
smooth and shining, with fine hairs on them
</aspect>
```

Figure 8: A more complex morphologic description

Because the additional markup is added only on levels beneath the most fine-grained element of the original schema, we can easily apply the approach to ABCD and taXMLit as well. Due to space constraints, we only display the div element containing the description. The markup is similar to the character and state elements provided by TaxonX. The difference is that the latter mark up the description terms (state) instead of the morphological feature names. As a consequence of this extension, our Example CQ1 can now be formulated sufficiently exactly:

```
treatment[./div[./@type = “description”]
/p/aspect[./feature[./text() = “Head”] and contains(./text(),
“sub-opaque”)]]
```

Example CQ3 now would formulate as below, and this query would not match the markup example in Figure 7. This is because the aspect elements now delimit the individual concepts in the description.

```
treatment[./div[./@type = “description”]
/p/aspect[./feature[./text() = “Head”] and contains(./text(),
“shining”)]]
```

Consequently, we can express the relation between the name of a morphological feature and its associated description. Again, this is not for free: We expect the performance of the Name Queries and Location Queries to decrease due to larger document size and the increased number of XML elements.

Normalized Aspect Markup

A slight problem of the Aspect Markup approach is that we use the original text for the identification of the feature names: We only enclose the respective words in feature markup. This can result in spelling-related errors due to singular/plural or capitalization differences. By adding a normalized form of the feature name (e.g., all lower case) to the feature tag as an attribute value, we can overcome this problem.

The example from above would now look like the fragment in Fig. 9.

```
<div type="description">
<p>
  <aspect>
    <feature name="mandibles">
      Mandibles
    </feature> and
    <feature name="clypheus">
      clypeus
    </feature> smooth and shining
  </aspect>,
  <aspect>
    <feature name="head">Head</feature> and
    <feature name="thorax">thorax</feature>
     sub-opaque
  </aspect>.
</p></div>
```

Figure 9: Morphologic description with individual aspects marked up, feature names normalized

The resulting changes to the query formulating Example CQ1 are minimal:

```
treatment[./div[./@type = "description"]
/p/aspect[./feature[./@name = "head"] and contains(./text(),
"sub-opaque")]]
```

The advantages are the same as those of the aspect markup, plus the spelling insensitivity. The latter comes at the cost of a little data redundancy and some more XML elements. Consequently, we expect this extension to affect the performance of Name Queries and Location Queries a little more than the aspect markup.

EVALUATION

In this section, we report on our evaluation of the different schemas and the extensions proposed in the previous section. Before we report on results, we briefly describe the experimental setup.

Experimental Setup

We have used **Altova XMLSpy 2005**, a widely used up-to-date XML editor, to execute the queries. The experiments have been run on a machine equipped with an **Intel Pentium IV Dual Core** and **1024 MB of RAM**. While we are aware of the fact that a native XML database or an SQL database with XML extensions might provide better performance, our setup corresponds to the way biosystematicists work with documents today. In addition, processing queries on files is an approach that the database-research community has paid much attention to in the recent past (Abiteboul et al. 1993, 1995).

Test Data

Because the digitization of biosystematics publications has just started, real documents marked in the different schemas are not available in numbers sufficiently large for large-scale performance experiments. Hence, we have generated artificial documents based on data taken from Wheeler (1922). The important parts for our experiments are the taxonomic names, collection events and textual descriptions of the taxa. We have generated these parts in the following way:

The **taxonomic names** were randomly assembled from genus, subgenus, species, subspecies and variety names we have extracted from (Wheeler 1922). Genus and species are always given, the remaining parts were added with the following probabilities:

Subgenus:	30%
Subspecies:	60%
Variety:	30%

Although the resulting taxonomic names do not exist in the real world, they are syntactically identical to real ones. This is sufficient for performance measurements.

The **collection events** mainly consist of a location, often accompanied by the name of the biologist who collected the specimen. We have synthesized these parts by inserting a location name into a sentence pattern, which lets it look more natural. The location was randomly picked from a given list. Again, identity on the syntactic level is sufficient for performance experiments.

The **textual descriptions** were generated by randomly lining up descriptive sentences. We have extracted the sentences from the textual descriptions in Wheeler (1922). Out of 70 different sentences, we have used 2 to 6 for each description paragraph. 2 to 7 paragraphs form a complete description. The intention was to produce descriptions of varying length, in order to arrive at a realistic distribution of the sizes of the documents.

By inserting these three parts into a pattern, we have obtained artificial treatments. For our experiments, we have generated documents containing 1,000 of such treatments. The plain text has a size of about 2 MB. The markup in the various schemas results in the document sizes listed below:

ABCD:	2.5 MB
TaxonX:	2.6 MB
taXMLit:	4.6 MB
SDD / UBIF:	9.6 MB

The difference between the ABCD and TaxonX is minimal. It results from the slightly higher level of layout details in TaxonX. But the

difference of these two schemas to taXMLit is significant. The reason is the large number of tags used in the latter schema. Finally, SDD / UBIF requires almost four times the storage space of ABCD and TaxonX. But in contrast to taXMLit, this comparison in isolation is not very significant. This is because SDD / UBIF is the only data-centric schema in this evaluation.

Results with Plain Schemas

In this subsection, we present the results of our performance experiments with the original schemas. In particular, we have run the Name Queries, Location Queries and Concept Queries on the documents. Table 1 lists the execution times. As expected, ABCD and TaxonX provide almost equal performance. The more complex nesting in taXMLit results in the duplication of execution time. The processing time for the queries on the SDD / UBIF document is significantly higher. The reason is that this particular document contains more than twice the number of XML elements of the others.

Schema	Name Queries	Location Queries	Concept Queries
ABCD	2.25	2	2.25 IR
TaxonX	2	3	3.25 IR
taXMLit	4	5	5.5 IR
SDD/UBIF	12.5	22.5	OOM

Table 1: Results with Unmodified Schemas (query execution time in seconds; OOM Out of Memory error).

SDD / UBIF is the only schema that allows formulating the Concept Queries such that results are always correct. However, resolving the ID references (i.e., computing the join) results in an Out Of Memory (OOM) error for large documents, as we had hypothesized. As expected, all the other schemas produce incorrect results for the Concept Queries ('IR' in the table). In particular, the queries return treatments where the queried attribute does not describe the morphologic feature concept queried.

Results with Extended Schemas

The experimental results presented in the last section have substantiated our expectation that none of the schemas supports the Concept Queries properly, except for SDD / UBIF. We have proposed three possible extensions to overcome this problem. In this section, we present the results of our experiments with the extended schemas.

The creation of a **morphologic index** for each treatment is the most data-centric extension. After generating the index, our test documents have the following sizes:

ABCD:	5.2 MB
TaxonX:	5.3 MB
taXMLit:	7.3 MB
SDD / UBIF:	12.4 MB

As expected, the size of the documents marked up in document-centric schemas has almost doubled. This is due to the redundancy caused by the index. Only SDD / UBIF is less affected (30% larger). This is because we could simply attach a description element to the existing concept elements instead of creating a complete index. Nevertheless, the SDD / UBIF document is still 2.5 times as big as the ones marked in ABCD and TaxonX, respectively. Table 2 lists the execution times of the different queries:

Schema	Name Queries	Location Queries	Concept Queries
ABCD	4.25	7	9
TaxonX	4.5	8	9
taXMLit	6.5	11.5	14
SDD / UBIF	14	30.75	34.25

Table 2: Results with Morphologic Index (query execution time in seconds)

With the morphologic index, all schemas allow formulating the Concept Queries sufficiently exactly so that they return correct results. Avoiding the join also overcomes the out-of-memory problems with SDD. The decreased performance of the Name Queries and Location Queries results from the increased number of XML elements.

The fine-grained markup of description aspects preserves the original document structure and produces no redundant data. The document size increases only by the additional tags. In particular, our test documents have the following sizes with this extension:

ABCD:	3.2 MB
TaxonX:	3.3 MB
taXMLit:	5.3 MB
SDD / UBIF:	10.3 MB

This extension increases the document size by about 25% for ABCD and TaxonX. The increase is less for the other documents. This is due to their larger original size because the additional tags are the same for all schemas. Table 3 lists the resulting query-execution times.

Schema	Name Queries	Location Queries	Concept Queries
ABCD	4.75	7	8.75
TaxonX	5	8	10.25
taXMLit	7	10.5	13
SDD/UBIF	15	30	OOM / 37

Table 3: Results with Aspect Markup (query-execution time in seconds; OOM Out of Memory error).

The impact on the performance of the Name Queries and Location Queries is slightly higher than that of the morphologic index. This is because the special markup of all features and description aspects introduces more additional XML elements. But the Concept Queries all produce correct results. For SDD / UBIF, we report two results in the Concept Queries column because the original query using the ID references did not execute successfully, but produced an Out Of Memory (OOM) again. In order to avoid the join, we then reformulated the query for Example CQ1 so that it does not involve the concept elements in Terminology, but only the aspects (see below). The new query executed successfully and produced correct results.

```
Dataset[DescriptiveData/NaturalLanguageDescriptions/
NaturalLanguageDescription/NaturalLanguageData/
Concept/Tex/aspect[./feature[./text() = “Head”] and
contains(./text(), “sub-opaque”)]]
```

Finally, the extension of the aspect markup with normalized feature names enlarges the documents as much as the markup of the aspects. The document sizes now are as follows:

ABCD:	3.5 MB
TaxonX:	3.6 MB
taXMLit:	5.6 MB
SDD / UBIF:	10.7 MB

This extension increases the document size a little more than the sole markup of the aspects and features. This is due to the additional name attribute in the feature tags. The execution times of the different queries are listed in Table 4. Name Queries and Location Queries execute slower than with the plain aspect markup and with the morphologic index. This is because the additional name attribute of the feature element introduces an additional XML node that has to be processed. On the other hand, the normalized aspect extension provides the best support for the Concept Queries. This is because it abstracts from singular/plural and other spelling-related differences between different instances of the same feature name. The reason that we list two results for the SDD / UBIF document is the same as in the previous section: The query is only processed successfully if we ignore the concept elements. Otherwise, it produces an Out Of Memory error.

Schema	Name Queries	Location Queries	Concept Queries
ABCD	9.75	12.25	13
TaxonX	6	11	14
taXMLit	8	14.5	18
SDD/UBIF	18	35	OOM/45

Table 4: Results with Normalized Aspect Markup (query execution time in seconds; OOM Out of Memory error)

DISCUSSION

Our evaluation points out several similarities and differences between the schemas. With regard to query formulation, TaxonX and ABCD are almost equivalent. The latter produces slightly smaller documents, while the former preserves the original document structure better. The query performance is approximately equal, and both have the semantic problem with the Concept Queries. But these in turn are easy to solve with one of the extensions proposed in this paper. Despite its complexity, taXMLit offers little semantic or structural advantages with regard to the aspects investigated here. It enlarges the documents by about 80%, compared to the first two schemas. Finally, SDD / UBIF is the only schema that supports the Concept Queries without schema modifications, at least in theory. The ID references result in Out Of Memory errors if the document size exceeds a certain limit. In addition, this advantage goes along with an increased document size, almost four times the one of documents marked in ABCD or TaxonX. Finally, the advantage becomes less significant if we use one of the extensions with the latter two schemas. Even with the aspect extension, the size of the ABCD and TaxonX documents is about a third of the size of a document marked up in original SDD / UBIF. Consequently, query execution is about twice as fast with the former two schemas.

The three extensions we have proposed serve their intended purpose well: They all facilitate correct results for the Concept Queries with each of the schemas. But they also have some side effects: The data-centric approach of adding a morphologic index to the documents induces the most redundancy, and the documents become significantly larger. Nevertheless, it does not offer any advantages

over the detailed markup of aspects in the textual description of the taxon. This applies to query formulation as well as to performance. The two flavors of the aspect markup only have slight differences with regard to the document size. Regarding query execution, however, they differ significantly: While the normalized version provides slightly better support for formulating the Concept Queries, it also has a non-negligible impact on query performance. Though the redundancy induced is very small, the number of XML elements increase significantly because attributes are represented as extra elements. The unnormalized version yields better querying performance for the Name Queries and the Location Queries, comparable to the morphological index, while avoiding the redundancy. Thus it does not enlarge documents very much. The only non-negligible drawback in comparison to the other two versions is that formulations of Concept Queries have to pay attention to different possible spellings of morphological feature names.

CONCLUSION

In this paper, we have presented and compared the XML schemas that are being used or developed as standards for the markup of taxonomic publications. We have considered both the size of the marked-up documents and the performance of queries against documents marked up using these standards. In particular, we have used three types of queries, which cover the three basic information needs in biosystematics:

1. Finding the description and dispersal of a taxon with a given name.
2. Finding all taxa ever reported to appear in a given area.
3. Finding all taxa that have a given morphological feature.

Although there are several more schemas, we have restricted our comparison to the four which allow formulating these three types of queries. Three of the schemas (ABCD, TaxonX and taXMLit) are document-centric, i.e., they intend to preserve the original structure of the publication. The fourth schema (SDD / UBIF) is more data-centric, i.e., it focuses on representing the data in a form that better supports query formulation and execution.

Because none of the document-centric schemas properly supports the third type of queries, we have proposed and evaluated three possible extensions to overcome this problem. Our evaluation has shown that they are all feasible for this purpose. They differ in the amount of redundant data introduced to the documents:

1. The creation of a morphologic index for each treatment is the most data-centric approach. Unfortunately, it induces a redundant representation of almost the entire textual description.
2. The markup of description aspects works in-line. It adds data-centric fine-grained markup to the leaves of the document-centric schema.
3. The normalization of the feature names in the aspects slightly accelerates query execution, at the cost of a little redundancy.

From the querying point of view, we deem ABCD and TaxonX most feasible for the markup of taxonomic publications. With the aspect markup extension, both support all queries and only slightly enlarge the documents. They provide acceptable query performance.

taXMLit introduces very complex markup. It enlarges the documents to almost twice the size of ABCD and TaxonX, but offers no advantages over the latter two schemas, at least not with regard to the aspects covered by this evaluation. The number of XML elements and the nesting complexity also significantly decrease query performance.

Finally, SDD / UBIF is the only schema that natively supports the third type of queries, but at a high price: The documents are almost four times the size of ABCD or TaxonX documents. In addition, this type of queries only executes for small documents. Larger ones produce errors because of insufficient computation resources. Finally, SDD / UBIF does not preserve the original structure of the document. Even with our description aspect extension, the size of an ABCD or TaxonX document is little more than a third of the one of an SDD / UBIF document. On the other hand, the aspect markup compensates all querying advantages that SDD / UBIF has over native ABCD and TaxonX. This applies to both correctness and performance.

ACKNOWLEDGMENTS

The authors thank the members of the project team (supported by awards from the US National Science Foundation IIS-0241229 and Deutsche Forschungsgemeinschaft BIB47) for their comments.

REFERENCES

ABCD. Access to Biological Collection Data[2]

Abiteboul, S., S. Cluet, and T. Milo, 1993. Querying and Updating the File. Proceedings of the 19th International Conference on Very Large Data Bases, 73 – 84, Dublin, Ireland

Abiteboul, s., S. Cluet, and T. Milo, 1995. A database interface for file update. ACM SIGMOD Record 24(2): 386-397

Altinel, M., and M. J. Franklin, 2000. Efficient Filtering of XML Documents for Selective Dissemination of Information. Proceedings of the 26th International Conference on Very Large Data Bases, 53 –64, Cairo, Egypt

Böhm, K., 2000. On Extending the XML Engine with Query-Processing Capabilities. in Proceedings of IEEE Advances in Digital Libraries, 127-138, Washington, DC, USA

DarwinCore2[3]

Ives, Z.,, A. Levy, D. Weld, 2000. Efficient Evaluation of Regular Path Expressions on Streaming XML Data, Technical Report, University of Washington, Seattle, WA, USA

LinneanCore[4]

NCD: Natural Collections Description[5]

Nambiar, U., Z. Lacroix, S. Bressan, L. L. Mong, and L. Yingguang, 2000. Current approaches to XML management. IEEE Internet Computing 6(4): 43-51.

Sautter, G., D. Agosti, K. Böhm, 2006. A Combining Approach to Find All Taxon Names (FAT) in Legacy Biosystematics Literature, Artikel, Biodiversity Informatics 3: 41-53

Sautter, G., D. Agosti, K. Böhm, 2007. Semi-Automated XML Markup of Biosystematics Legacy Literature with the GoldenGATE Editor, in Proceedings of PSB 2007, Weilea, HI, USA

SDD: Structure of Descriptive Data[6]

TaxonX[7]

TCS: Taxonomic Concept transfer Schema[8]

UBIF: Unified Biosciences Information Framework.[9]

Weitzman, A. L., C. H. C. Lyal. An XML schema for taxonomic literature – taXMLit[10]

Wheeler, W. M., 1992. The Ants of Trinidad. American Museum Novitates 45: 1-16

Wu, Y. J.M. Patel, and H. Jagadish, 2003. Structural Join Order Selection for XML Query Optimization, in Proceedings of ICDE, 443-454, Bangalore, India

XML: Extensible Markup Language[11]

Xpath[12]

[2] http://www.bgbm.org/TDWG/CODATA/Schema/

[3] http://darwincore.calacademy.org/

[4] http://wiki.cs.umb.edu/twiki/bin/view/UBIF/LinneanCore

[5] http://www.tdwg.org/NCD/TDWG_NCD_Subgroup.htm

[6] http://wiki.cs.umb.edu/twiki/bin/view/SDD

[7] http://sourceforge.net/projects/taxonx

[8] http://tdwg.napier.ac.uk

[9] http://wiki.cs.umb.edu/twiki/bin/view/UBIF

[10] http://www.sil.si.edu/digitalcollections/bca/status.cfm

[11] http://www.w3.org/XML/

[12] http://www.w3.org/TR/xpath