Generating Highly Structured XML from Word Processing Documents

An Overview of mPach

Carrick Rogers
carrickr@umich.edu
734/330-9246

MPublishing

- Heavy focus on electronic publishing:
  - 'Camera Ready' copy for consumption.
  - Archival copy for storage in Deep Blue, DLXS Collections, or Hathi Trust

- PTG (Publishing Technology Group) is a group of 4 programmers within MPublishing.

TEI Workflow

1. Publishing Partner submits a .rtf document.
2. MPub Staff make alterations to the document.
3. TEI is then produced using a number of Perl Scripts, Shell Scripts, and XSLT.
4. Other aspects of the submission package (ex: MARC record) created by hand or using tools from other groups within the University of Michigan Library.
5. Final product is placed in a DLXS Collection.

mPach Goals

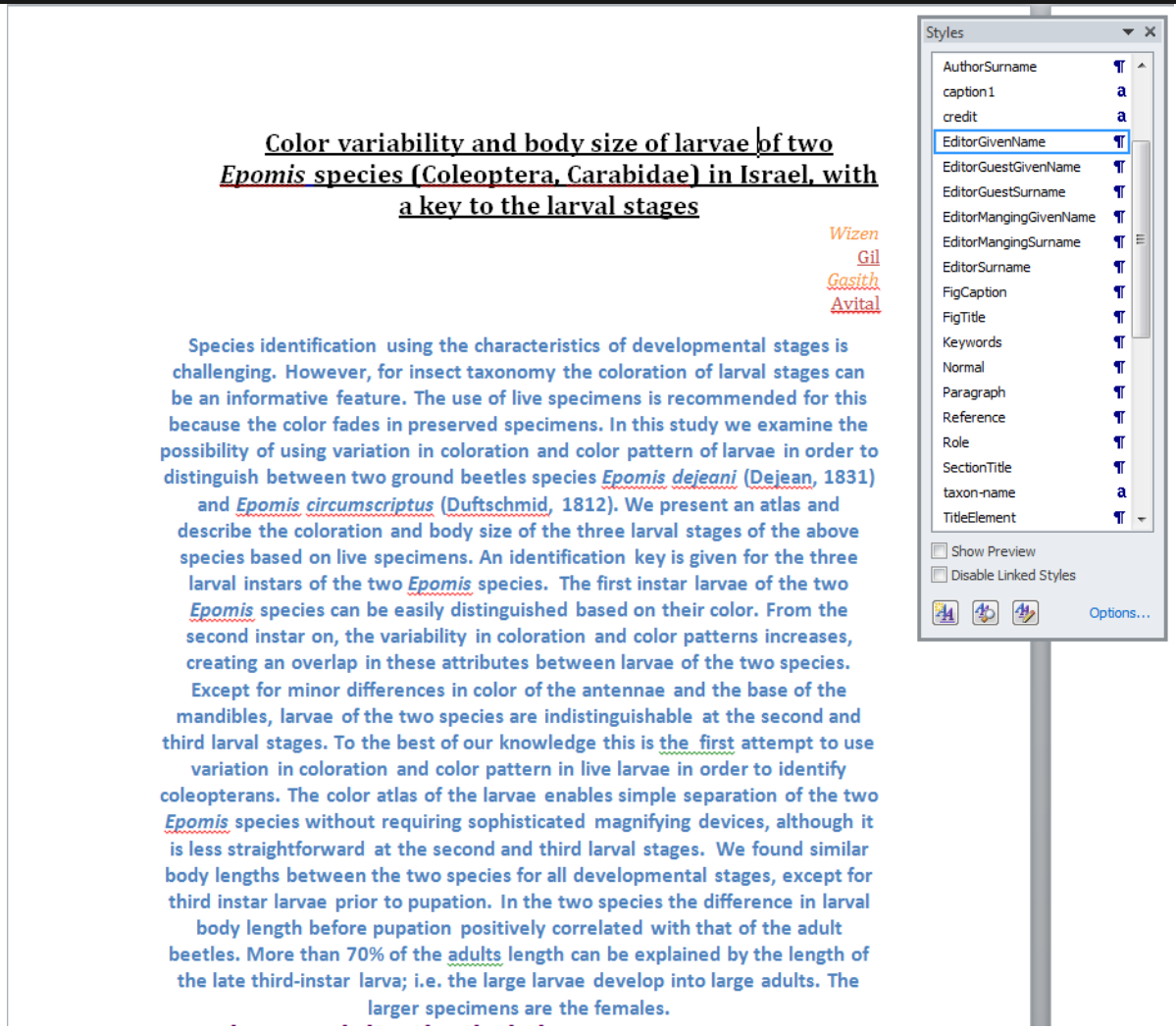
- Replace TEI with NISO JATS.
- One portal to produce a camera ready and an archival copy of the article that is automatically submitted to the Hathi Trust.
- Provide the journal editor with more control over the process .
  - ...and conveniently reduce the workload of our internal staff.
- Simplify the code base.

mPach Requirements

- Allow publishers to conduct their workflow using word processing documents.
  - Editors wanted to retain features such as track changes, comment support, etc.
  - Creating a replacement solution was not in scope.

- Allow publishers to customize the output of mPach and support journal specific styles (assuming JATS supports them).

Word Document Template



Color variability and body size of larvae of two *Epomis* species (Coleoptera, Carabidae) in Israel, with a key to the larval stages

Wizen
Gil
Gasith
Avital

Species identification using the characteristics of developmental stages is challenging. However, for insect taxonomy the coloration of larval stages can be an informative feature. The use of live specimens is recommended for this because the color fades in preserved specimens. In this study we examine the possibility of using variation in coloration and color pattern of larvae in order to distinguish between two ground beetles species *Epomis dejeani* (Dejean, 1831) and *Epomis circumscriptus* (Duftschmid, 1812). We present an atlas and describe the coloration and body size of the three larval stages of the above species based on live specimens. An identification key is given for the three larval instars of the two *Epomis* species. The first instar larvae of the two *Epomis* species can be easily distinguished based on their color. From the second instar on, the variability in coloration and color patterns increases, creating an overlap in these attributes between larvae of the two species. Except for minor differences in color of the antennae and the base of the mandibles, larvae of the two species are indistinguishable at the second and third larval stages. To the best of our knowledge this is the first attempt to use variation in coloration and color pattern in live larvae in order to identify coleopterans. The color atlas of the larvae enables simple separation of the two *Epomis* species without requiring sophisticated magnifying devices, although it is less straightforward at the second and third larval stages. We found similar body lengths between the two species for all developmental stages, except for third instar larvae prior to pupation. In the two species the difference in larval body length before pupation positively correlated with that of the adult beetles. More than 70% of the adults length can be explained by the length of the late third-instar larva; i.e. the large larvae develop into large adults. The larger specimens are the females.

Structured XML Challenge

An article title in NISO JATS:

```
<article>
   <front>
      <title>
         <article-meta>
            <title-group>
               <article-title>
               Color variability and body size of
               larvae of two <i>Epomis</i> species
               (Coleoptera, Carabidae) in Israel,
               with a key to the larval stages
               </article-title>
```

Structured XML Challenge

In MS Word

```
<w:body>
<w:p>
        <w:pPr><w:pStyle w:val="ArticleTitle"/></w:pPr>
    <w:r>
        <w:t>Color variability and body size of larvae of
        two</w:t>
    </w:r>
    <w:r>
        <w:rPr><w:i/></w:rPr>
        <w:t>Epomis</w:t>
    </w:r>
    <w:r>...
```

mPach Norm Usage

- Decoupled from Rails, can be called via command line
- Input: .docx or .odt file
- Output:

  document_name.zip/

    document_name.xml (NISO JATS)

    assets/

      image_1.png

      image_2.png

      ....

mPach Norm Class Map

norm.py

fileorganizer.py

dochandler.py

- docxhandler.py
- odthandler.py
- *latexhandler.py*

xmlcreation.py

- jatscreation.py
- *teicreation.py*

Moving Data Between DocHandler and XMLCreation

List of Tuples:

```
[(jatsElement, [content],wordStyle)]
```

where `content` is:

```
[(text, [inline styles], special)]
```

Moving Data Between DocHandler and XMLCreation

Title:

Color variability and body size of larvae of two *Epomis* species (Coleoptera, Carabidae) in Israel, with a key to the larval stages

As a Tuple:

```
('article-title',
   [('Color vari...of two', None, None),
    ('Epomis', ['i'], None)
    ('(Coleoptera...stages', None, None)],
 'ArticleTitle')
```

Structured XML Challenge

```
<w:body>
<w:p>
        <w:pPr><w:pStyle w:val="ArticleTitle"/></w:pPr>
    <w:r>
        <w:t>Color variability and body size of larvae of
        two</w:t>
    </w:r>
    <w:r>
        <w:rPr><w:i/></w:rPr>
        <w:t>Epomis</w:t>
    </w:r>
    <w:r>...
```

To Add Source Document Support To Norm

A new *something*Handler.py must:

a. Have a function that returns three lists of these tuples:
   i. frontItems (metadata, similar to <head> in html)
   ii. bodyItems (the content of the article)
   iii. backItems (citations, footnotes, etc)

b. Have a function to extract an media embedded in the document.

Still Don't Have The Depth Though...

An article title in NISO JATS:

```
<article>
    <front>
        <title>
            <article-meta>
                <title-group>
                    <article-title>
                    Color variability and body size of
                    larvae of two <i>Epomis</i> species
                    (Coleoptera, Carabidae) in Israel,
                    with a key to the larval stages
                    </article-title>
```

The .cfg File, Depth and User Friendly! (1 of 2)

[FRONT]
ArticleTitle = article-title
AuthorGivenName = given-names
AuthorSurname = surname
Keywords = kwd-group
Keyword = kwd
EditorGivenName = given-names
EditorSurname = surname
EditorGuestGivenName = given-names
EditorGuestSurname =surname
Abstract = abstract
TranslatedSubTitle = trans-subtitle

[BODY]
Paragraph = p
....

[BACK]
Reference = ref

[FRONT-PARENTS]
article-title=title-group
article-meta = front
title-group = article-meta
given-names = name
surname = name
name = contrib
role = contrib
contrib = contrib-group
contrib-group = article-meta
kwd-group = article-meta
author-notes = front
abstract = article-meta

The .cfg File, Depth and User Friendly! (2 of 2)

[CHILDRENLIMITS]
title-group= 1
article-title= 1
given-names=1
surname=1
caption=1
name = 1
title = 1
label = 1
suffix = 1
prefix = 1

[ATTRIBUTESNEEDED]
contrib = Yes
kwd-group = Yes
fig = Yes

[ATTRIBUTES]
AuthorGivenName=contrib-type,author
AuthorSurname = contrib-type,author
AuthorKeywords = kwd-group-type,author
EditorGivenName = contrib-type,volume-editor
EditorSurname = contrib-type,volume-editor
EditorGuestGivenName = contrib-type,guest-editor
EditorGuestSurname = contrib-type,guest-editor
EditorKeywords = kwd-group-type,editor
Keywords = kwd-group-type,author
Figure = orientation,portrait;position,float;id
FigTitle = id
FigCaption = id
table = id

So how is this is friendly?

Example 1: I want to add to support for the subtitle element.

```
<article>
    <front>
        <title>
            <article-meta>
                <title-group>
                    <article-title>Color...</article-
                    title>
                    <subtitle>Beetles are really
                    interesting</subtitle>
                </title-group>
            ...
```

So how is this is friendly?

1. Open Word and add a 'Subtitle' paragraph style.

2. Open your .cfg and add:

```
[FRONT]
Subtitle = subtitle
[FRONT-PARENTS]
subtitle = title-group
```

3. Tag your Word document and upload it to MPach.

.cfg Changes for Subtitle

[FRONT]
ArticleTitle = article-title
Subtitle=subtitle
AuthorGivenName = given-names
AuthorSurname = surname
Keywords = kwd-group
Keyword = kwd
EditorGivenName = given-names
EditorSurname = surname
EditorGuestGivenName = given-names
EditorGuestSurname =surname
Abstract = abstract

[BODY]
Paragraph = p
....

[BACK]
Reference = ref

[FRONT-PARENTS]
article-title=title-group
subtitle=title-group
article-meta = front
title-group = article-meta
given-names = name
surname = name
name = contrib
role = contrib
contrib = contrib-group
contrib-group = article-meta
kwd-group = article-meta

So this is friendly how?

Example 2: I want to add a managing editor.

1. Open your .docx and add 'EditorManagingSurname' and 'EditorManagingGivenName' as a paragraph styles.
2. Open your .cfg file and add:

```
[FRONT]
EditorManagingGivenName = given-names
EditorManagingSurname = surname
[ATTRIBUTES]
EditorManagingGivenName = contrib-type,managing-editor
EditorManagingSurname = contrib-type,managing-editor
```

3. Tag in Word and upload to mPach.

.cfg Changes for Managing Editor

[FRONT]
ArticleTitle = article-title
AuthorGivenName = given-names
AuthorSurname = surname
Keywords = kwd-group
Keyword = kwd
EditorGivenName = given-names
EditorSurname = surname
EditorGuestGivenName = given-names
EditorManagingSurname =surname
EditorManagingGivenName = given-names
EditorGuestSurname =surname
Abstract = abstract
TranslatedSubTitle = trans-subtitle

[BACK]
Reference = ref

[FRONT-PARENTS]
article-title=title-group
article-meta = front
title-group = article-meta
given-names = name
surname = name
name = contrib
role = contrib
contrib = contrib-group
contrib-group = article-meta
kwd-group = article-meta
author-notes = front
abstract = article-meta

.cfg Changes for Managing Editor

[CHILDRENLIMITS]
title-group= 1
article-title= 1
given-names=1
surname=1
caption=1
name = 1
title = 1
label = 1
suffix = 1
prefix = 1

[ATTRIBUTESNEEDED]
contrib = Yes
kwd-group = Yes
fig = Yes

[ATTRIBUTES]
AuthorGivenName=contrib-type,author
AuthorSurname = contrib-type,author
AuthorKeywords = kwd-group-type,author
EditorGivenName = contrib-type,volume-editor
EditorSurname = contrib-type,volume-editor
EditorGuestGivenName = contrib-type,guest-editor
EditorGuestSurname = contrib-type,guest-editor
EditorManagingSurname =contrib-type,managing-editor
EditorManagingGivenName = contrib-type,managing-editor
EditorKeywords = kwd-group-type,editor
Keywords = kwd-group-type,author
Figure = orientation,portrait;position,float;id
FigTitle = id
FigCaption = id
table = id

So how is this is friendly?

Example 2: I want to add to support for the subtitle element.

```
<article>
    <front>
        <title>
            <article-meta>
                <title-group>
                    <article-title>Color...</article-
                    title>
                    <subtitle>Beetles are really
                    interesting</subtitle>
                </title-group>
            ...
```

Using the .cfg file to create JATS

1. `jatscreation` has been passed three lists of tuples, frontItems, bodyItems, and backItems
2. Initialize a basic JATS document:

```
<article>
    <front/>
    <body/>
    <back/>
</article>
```

3. Start running down each list and placing it.

Placing An Item

Example: `article-title` is the first item to place.

Article Title's tuple produces this:

```
<article-title>Color variability and body size of larvae
of two <i>Epomis</i> species (Coleoptera, Carabidae) in
Israel, with a key to the larval stages</article-title>
```

The DOM looks like this:

```
<article>
    <front/>
    <body/>
    <back/>
</article>
```

Element Placement Code

```
def itemPlacement(self, sec_item, item, elem, lastElem):

 #Get the name of the desired parent for article-title from the cfg file
        pTagName = self.files.Map.get(self.PARENT_SEC, item[0]).lower()
 #The result will be pTagName = 'title-group'

        #CASE 0, This Element Goes Right Under A Major Section, easy placement
        if(pTagName == sec_item.tagName.lower()):
        sec_item.appendChild(elem)
        return elem
 #This is not the case, our pTag is 'title-group', sec_item.tagName is 'front'

        #CASE 1, Can Be Placed under any element, append it to the last one
        if(pTagName == self.ANY):
        lastElem.append(elem)
        return elem
 #pTagName is not 'Any', not a valid case.
```

Element Placement Code

```
#CASE 2, Can I Place It In the Current DOM?
        lE = lastElem
        while lastElem.tagName.lower() != sec_item.tagName.lower():
        if lastElem.tagName.lower() == pTagName and self.parentHasRoomForChild(lastElem, item[0]):
                lastElem.appendChild(elem)
                return elem
        else:
                lastElem = lastElem.parentNode
  #Since I just initialized this DOM, my lastElem is 'front', not a valid case


        #CASE 3, Could Not Place In Current DOM, Make The Parent And Try To Place the Parent in
the DOM
        nI = (pTagName, None, item[2])
        nE = self.createElement(nI)
        nE.appendChild(elem)
        return self.itemPlacement(sec_item, nI, nE, lE)
```

Element Placement Code

#I've now created:

<title-group>

<article-title>

Color variability and body size of larvae of two Epomis species (Coleoptera, Carabidae) in Israel, with a key to the larval stages

</article-title>

</title-group>

#I'm going to call this same function as:

self.itemPlacement(DOM Object For Front, ('title-group', None,None) DOM Object for Title Group, DOM Object for Front)

#Since the parent of title-group is article-meta we'll end up in CASE 3 again and repeat this process.

#article-meta has a parent of front though, so CASE 0 will fire when that is called

#The last element considered placed will be article-title

A Harder .cfg Change

Example 3: Add translated title support.

```
<article>
    <front>
        <title>
            <article-meta>
                <title-group>
                    <article-title>Color...</article-
                    title>
                    <trans-title-group xml:lang="zh">
                        <trans-title>颜色...</trans-title>
                    </trans-title-group>
                </title-group>
```

A Harder .cfg Change

1. Add the style 'TranslatedTitle' or 'TranslatedTitleZH' to your Word document.
2. Open your .cfg and add:

```
[FRONT]
TranslatedTitle = trans-title
[FRONT-PARENTS]
trans-title = trans-title-group
trans-title-group = title-group
[ATTRIBUTESNEEDED]
trans-title-group = yes
[ATTRIBUTES]
TranslatedTitle = xlm:lang,zh
```

3. Tag the document and upload.

.cfg Changes for Translated Title

[FRONT]
ArticleTitle = article-title
AuthorGivenName = given-names
AuthorSurname = surname
Keywords = kwd-group
Keyword = kwd
EditorGivenName = given-names
EditorSurname = surname
EditorGuestGivenName = given-names
TranslatedTitle = trans-title
EditorGuestSurname =surname
Abstract = abstract
TranslatedSubTitle = trans-subtitle

[BACK]
Reference = ref

[FRONT-PARENTS]
trans-title = trans-title-group
trans-title-group = title-group
article-title=title-group
article-meta = front
title-group = article-meta
given-names = name
surname = name
name = contrib
role = contrib
contrib = contrib-group
contrib-group = article-meta
kwd-group = article-meta

.cfg Changes for Managing Editor

[CHILDRENLIMITS]
title-group= 1
article-title= 1
given-names=1
surname=1
caption=1
name = 1
title = 1
label = 1
trans-title=1
trans-title-group = 1

[ATTRIBUTESNEEDED]
contrib = Yes
kwd-group = Yes
trans-title-group = Yes

[ATTRIBUTES]
AuthorGivenName=contrib-type,author
AuthorSurname = contrib-type,author
AuthorKeywords = kwd-group-type,author
EditorGivenName = contrib-type,volume-editor
EditorSurname = contrib-type,volume-editor
EditorGuestGivenName = contrib-type,guest-editor
EditorGuestSurname = contrib-type,guest-editor
TranslatedTitle = xml:lang, zh
EditorKeywords = kwd-group-type,editor
Keywords = kwd-group-type,author
Figure = orientation,portrait;position,float;id
FigTitle = id
FigCaption = id
table = id