2019 PAGE XML Format for Page Content

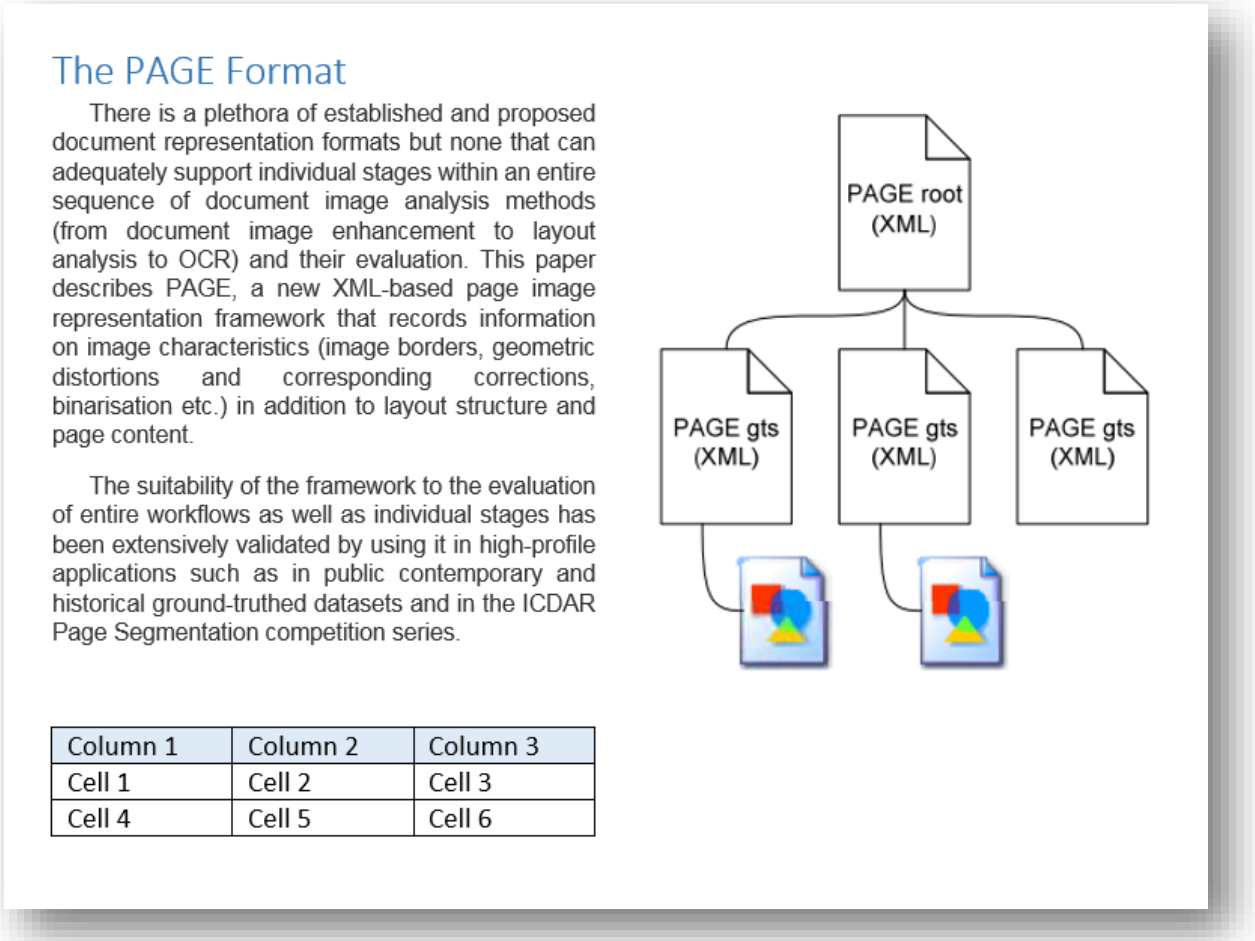
In this document we show the essential structure of the PAGE XML file format.

More information can be found here: http://www.primaresearch.org/tools/PAGELibraries

The example shows how metadata, regions, and reading order are stored. More complex concepts (such as text line, word and glyph objects) are not discussed.

Example Image

Images can be in TIFF, PNG, or JPEG format.



The PAGE Format

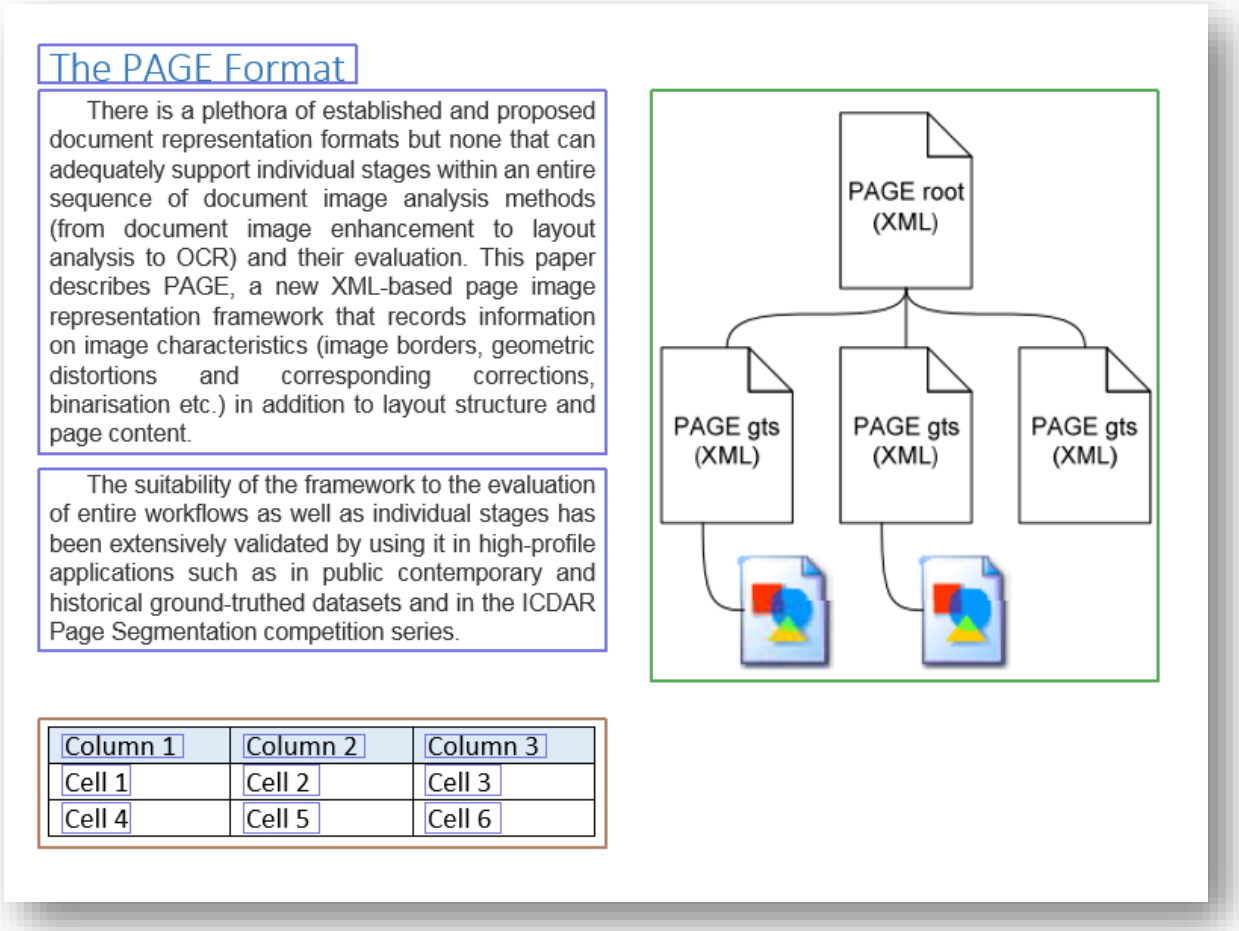
There is a plethora of established and proposed document representation formats but none that can adequately support individual stages within an entire sequence of document image analysis methods (from document image enhancement to layout analysis to OCR) and their evaluation. This paper describes PAGE, a new XML-based page image representation framework that records information on image characteristics (image borders, geometric distortions and corresponding corrections, binarisation etc.) in addition to layout structure and page content.

The suitability of the framework to the evaluation of entire workflows as well as individual stages has been extensively validated by using it in high-profile applications such as in public contemporary and historical ground-truthed datasets and in the ICDAR Page Segmentation competition series.

Column 1	Column 2	Column 3
Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

Annotated Page Content

The page content is can be annotated using the [Aletheia Document Analysis System](#).



PAGE XML (Page Content Ground Truth and Storage)

The XML schema can be found here:
http://schema.primaresearch.org/PAGE/gts/pagecontent/2016-07-15/pagecontent.xsd

All objects (regions, groups etc.) are identified with an ID which has to be unique within the whole XML file.

Main Structure

```
<?xml version="1.0" encoding="UTF-8"?>
<PcGts xmlns="http://schema.primaresearch.org/PAGE/gts/pagecontent/2019-07-15"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://schema.primaresearch.org/PAGE/gts/pagecontent/2019-07-15
        http://schema.primaresearch.org/PAGE/gts/pagecontent/2019-07-15/pagecontent.xsd">
        <Metadata>…</Metadata>
        <Page imageFilename="SimplePage.png" imageWidth="800" imageHeight="600">
                <ReadingOrder>…</ReadingOrder>
                <TextRegion>…</TextRegion>
                …
        </Page>
</PcGts>
```

Metadata

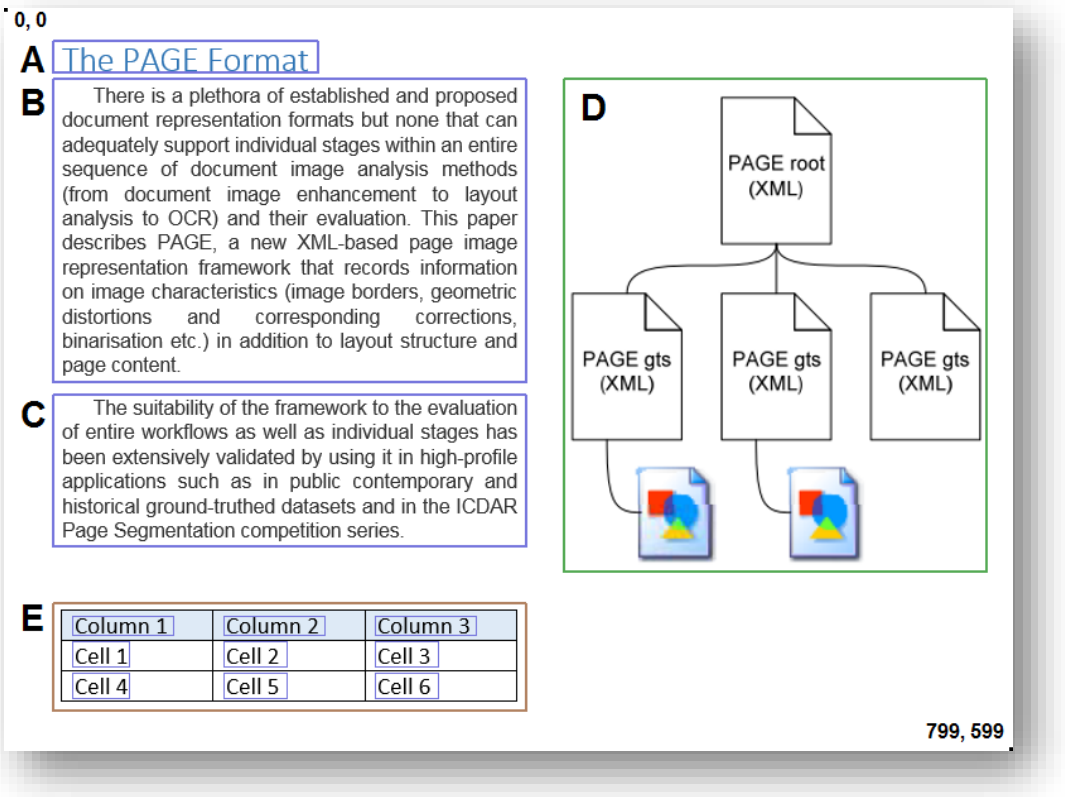
Various attributes regarding the PAGE file.

```
<Metadata>
        <Creator>Me</Creator>
        <Created>2017-05-03T10:20:47</Created>
        <LastChange>2017-05-03T10:27:21</LastChange>
</Metadata>
```

Regions

A region reflects a physical object on a page. Regions are defined by their type, outline (polygon), and attributes.

Following types are supported: TextRegion, ImageRegion, GraphicRegion, ChartRegion, LineDrawingRegion, SeparatorRegion, TableRegion, MathsRegion, ChemRegion, MusicRegion, AdvertRegion, NoiseRegion, UnknownRegion.



A

```
<TextRegion id="r0" type="heading">
        <Coords points="25,30 25,55 235,55 235,30"/>
        <TextEquiv>
                <Unicode>The PAGE Format</Unicode>
        </TextEquiv>
</TextRegion>
```

B

```
<TextRegion id="r1" type="paragraph">
        <Coords points="25,60 25,300 400,300 400,60"/>
        <TextEquiv>
                <Unicode>There is a plethora …</Unicode>
```

```
        </TextEquiv>
</TextRegion>
```

C

```
<TextRegion id="r2" type="paragraph">
        <Coords points="25,310 25,430 400,430 400,310"/>
        <TextEquiv>
                <Unicode>The suitability of …</Unicode>
        </TextEquiv>
</TextRegion>
```

D

```
<GraphicRegion id="r4">
        <Coords points="430,60 430,450 765,450 765,60"/>
</GraphicRegion>
```

E

```
<TableRegion id="r3" lineSeparators="true">
        …
</TableRegion>
```

Nested Regions

Regions can have sub-regions (nested regions). Examples are table cells or text in figures.

For the table region from above the XML looks like follows:

```
<TableRegion id="r3" lineSeparators="true">
        <Coords points="25,475 25,560 400,560 400,475"/>
        <TextRegion id="r5" type="paragraph">
                <Coords points="40,485 40,500 120,500 120,485"/>
                <TextEquiv>
                        <Unicode>Column 1</Unicode>
                </TextEquiv>
        </TextRegion>
        <TextRegion id="r6" type="paragraph">
                <Coords points="160,485 160,500 240,500 240,485"/>
                <TextEquiv>
```

```
                <Unicode>Column 2</Unicode>
            </TextEquiv>
        </TextRegion>
        <TextRegion id="r7" type="paragraph">
            <Coords points="280,485 280,500 360,500 360,485"/>
            <TextEquiv>
                <Unicode>Column 3</Unicode>
            </TextEquiv>
        </TextRegion>
        <TextRegion id="r8" type="paragraph">
            <Coords points="40,505 40,525 85,525 85,505"/>
            <TextEquiv>
                <Unicode>Cell 1</Unicode>
            </TextEquiv>
        </TextRegion>
        …
</TableRegion>
```

Reading Order

The reading order describes the logical order of text regions. It can have groups and sub-groups which can contain either ordered or unordered references to regions. The example page has a very simple sequential reading order.

```
<ReadingOrder>
        <OrderedGroup id="ro357564684568544579089">
            <RegionRefIndexed regionRef="r0" index="0"/> A
            <RegionRefIndexed regionRef="r1" index="1"/> B
            <RegionRefIndexed regionRef="r2" index="2"/> C
        </OrderedGroup>
</ReadingOrder>
```

Text Line Objects

Text line objects are sub-elements of TextRegion. Each text line is defined by it's bounding polygon, optional attributes, and the text content. The text content can be stored simultaneously in the text region and in the text line objects of a text region. If you choose one over the other or fill both depends on the use case.

```
<TextRegion id="r0" type="heading">
        …
        <TextLine id="l0">
                <Coords points="25,30 25,55 235,55 235,30"/>
                <TextEquiv><Unicode>…</Unicode></TextEquiv>
        </TextLine>
        …
</TextRegion>
```