

Encoding of Digitised Documents

STEFAN PLETSCHACHER

School of Science, Engineering and Environment

University of Salford, Salford, UK

Submitted in Partial Fulfilment of the Requirements of the Degree of

Doctor of Philosophy

2023

I. Table of Contents

I.	Table of Contents	ii
II.	Figures	viii
III.	Tables.....	xi
IV.	Declaration	xii
V.	Symbols.....	xiii
VI.	Abstract	xv
1	Introduction.....	1
1.1	Motivation.....	1
1.2	Aims	4
1.3	Objectives	5
1.4	Methodology.....	6
1.4.1	Encoding Scheme.....	6
1.4.2	Unstandardised Characters	6
1.5	Structure of the Thesis.....	8
1.6	Summary.....	9
2	Background and Related Work.....	10
2.1	Text Recognition	10
2.2	Representation of Digitised Documents.....	11
2.3	Document Image Analysis.....	14
2.4	Document Encoding Standards and Format Specifications.....	14
2.5	Clustering and Symbolic Compression.....	16
2.6	Summary	17
3	Format Requirements and Design Principles	19
3.1	Requirements.....	19

3.2	Design Principles	20
3.3	The PAGE Format Framework.....	21
3.4	Summary	22
4	Encoding Page Content.....	23
4.1	Root Element	23
4.2	PcGtsType	24
4.3	MetadataType.....	25
4.4	PageType.....	25
4.5	BorderType	26
4.6	CoordsType	27
4.7	PrintSpaceType	27
4.8	ReadingOrderType	28
4.8.1	OrderedGroupType	29
4.8.2	OrderedGroupIndexedType	30
4.8.3	UnorderedGroupIndexedType	31
4.8.4	UnorderedGroupType	32
4.9	LayersType	32
4.10	TextRegionType.....	33
4.10.1	TextLineType.....	35
4.10.2	WordType	36
4.10.3	GlyphType.....	37
4.11	ImageRegionType.....	37
4.12	LineDrawingRegionType.....	38
4.13	GraphicRegionType	39
4.14	TableRegionType	40
4.15	ChartRegionType	41

4.16	SeparatorRegionType	41
4.17	MathsRegionType	42
4.18	NoiseRegionType	43
4.19	FrameRegionType	43
4.20	UnknownRegionType	44
4.21	Summary	45
5	Evolution of the Pagecontent Format Specification	46
5.1	Second Release	46
5.2	Third Release	47
5.2.1	ReadingOrderType	48
5.2.2	TextTypeSimpleType	48
5.3	Fourth Release	49
5.3.1	CoordsType	49
5.3.2	General Region Nesting	50
5.3.3	Additional Region Types	51
5.3.4	TextStyleType	52
5.3.5	Custom and Comments Attributes	53
5.3.6	RelationsType	54
5.3.7	Baseline	55
5.3.8	Further Changes	56
5.4	Fifth Release	58
5.4.1	OCR/Text Variants	58
5.4.2	Attributes	60
5.5	Sixth Release	61
5.5.1	GraphemesType	62
5.5.2	UserDefinedType	64

5.5.3	RolesType.....	65
5.5.4	Further Changes	66
5.6	Seventh Release	67
5.6.1	Additional Region Types	67
5.6.2	LabelsType	69
5.6.3	MetadataItem Type	70
5.6.4	GridType	70
5.6.5	RelationType Update	71
5.6.6	Further Changes	72
5.7	Eighth Release.....	73
5.8	Summary	73
6	Encoding Unstandardised Characters	75
6.1	Adaptive Glyph Clustering	75
6.2	Formalising the Problem.....	76
6.3	Glyph Representation	77
6.3.1	Bitmap Images	78
6.3.2	Feature Vectors	79
6.4	Measuring the Similarity of Glyphs.....	82
6.4.1	Feature Based	83
6.4.2	Template Matching Based	84
6.4.3	A Combined Distance Measure	85
6.5	Clustering Algorithm	87
6.6	Determining an Appropriate Threshold.....	89
6.6.1	Supervised – From Known Examples.....	91
6.6.2	Unsupervised – Using a Surrogate Metric for Error Rate.....	94
6.6.3	Generalisation	97

6.6.4	Calculating the Threshold	99
6.7	Assigning Code Points	100
6.8	Evaluation	102
6.8.1	Datasets	102
6.8.2	Overall Results – Initial Configuration	102
6.8.3	Revisiting the Combined Distance Measure Weights	104
6.8.4	Overall Results – Optimal Configuration	104
6.8.5	Discussion of Results and Benefits	106
6.9	Summary	108
7	Discussion	109
7.1	Success Factors of PAGE	109
7.2	Technical Validity	110
7.3	Format Features.....	110
7.4	Official Tool Support	111
7.5	Datasets	114
7.6	Competitions.....	114
7.7	Impact	115
7.7.1	Citations.....	115
7.7.2	External Projects, Datasets and Tools	117
7.7.3	Influence on Other Formats	118
7.7.4	REF Impact Case Study	119
7.8	Applications of Adaptive Glyph Clustering	119
7.8.1	Training Data Generation / Semi-automatic OCR	120
7.8.2	Faithful Reproduction of Document Images	122
7.9	Summary	124
8	Conclusions and Future Work	125

8.1	Conclusions	125
8.2	Future Work	126
	References	127
	Appendix	140

II. Figures

Figure 1: Document images of different types [1]	2
Figure 2: Documents with regular yet difficult to process textual content	3
Figure 3: Text recognition result for an example of a historic document.....	4
Figure 4: Example page from the dataset of original document scans.....	7
Figure 5: Examples of labelled glyph images.....	7
Figure 6: Overview of the pagecontent schema	24
Figure 7: PcGts (root) element	24
Figure 8: PcGtsType with Metadata and Page elements	25
Figure 9: BorderType	27
Figure 10: CoordsType.....	27
Figure 11: PrintSpaceType	28
Figure 12: ReadingOrderType.....	29
Figure 13: OrderedGroupType	30
Figure 14: OrderedGroupIndexedType	31
Figure 15: UnorderedGroupIndexedType	31
Figure 16: UnorderedGroupType	32
Figure 17: LayersType	33
Figure 18: TextRegionType	34
Figure 19: TextLineType.....	36
Figure 20: WordType	36
Figure 21: GlyphType.....	37
Figure 22: ImageRegionType	38
Figure 23: LineDrawingRegionType.....	38
Figure 24: GraphicRegionType	39
Figure 25: TableRegionType	40
Figure 26: ChartRegionType	41
Figure 27: SeparatorRegionType	42
Figure 28: MathsRegionType.....	42
Figure 29: NoiseRegionType	43
Figure 30: FrameRegionType.....	44

Figure 31: UnknownRegionType	44
Figure 32: PageType with unrestricted number of regions.....	47
Figure 33: Updated ReadingOrderType.....	48
Figure 34: Updated CoordsType.....	50
Figure 35: Abstract RegionType (here extended by ImageRegionType).....	51
Figure 36: AdvertRegionType (ChemRegionType and MusicRegionType accordingly)	52
Figure 37: TextStyleType	53
Figure 38: Custom and comments attributes in RegionType.....	54
Figure 39: RelationsType	55
Figure 40: Baseline as defined in TextLineType	56
Figure 41: TextEquivType (as used on word level)	59
Figure 42: GraphemesType	62
Figure 43: GraphemeType	63
Figure 44: NonPrintingCharType	63
Figure 45: GraphemeGroupType.....	64
Figure 46: UserDefinedType	65
Figure 47: RolesType	65
Figure 48: MapRegionType.....	67
Figure 49: CustomRegionType.....	68
Figure 50: LabelsType	69
Figure 51: MetadataItem.....	70
Figure 52: GridType	71
Figure 53: RelationType.....	71
Figure 54: Adaptive Vector Quantisation algorithm for glyph clustering	88
Figure 55: Compression rate over increasing distance measure threshold.....	92
Figure 56: Error rate over increasing distance measure threshold.....	93
Figure 57: Maximum intra cluster distance and error rate over increasing threshold.....	96
Figure 58: Correlation between maximum intra cluster distance and error rate.....	98
Figure 59: Close-up of the interval relevant to the threshold selection problem.	99
Figure 60: Compression and error rate for w_t : 80% and w_f : 20% (initial configuration) ...	103
Figure 61: Compression and error rate for w_t : 90% and w_f : 10% (optimal configuration)	105
Figure 62: Editing a Newspaper Page in Aletheia [11]	112

Figure 63: PRImA Performance Evaluation Tool [73]	112
Figure 64: Citations of the Original PAGE Format Framework Paper [90]	116
Figure 65: Basic Glyph Segmentation Tool [116]	121
Figure 66: Glyph Labelling Interface [116]	121
Figure 67: Original Document Image (left) and Prototypes of the Document Specific Alphabet (right) [116]	122
Figure 68: Visual Quality of Vectorised Glyphs for Increasing Abstraction Level [68]	123
Figure 69: Rendered Output (Based on a Document Specific Alphabet and Font)	123

III. **Tables**

Table 1: Example features for distinguishing different character classes.....	79
Table 2: Datasets used for evaluation	102
Table 3: Compression and error rate for different combinations of weights	104
Table 4: Manual labelling effort – proposed approach vs no clustering.....	106
Table 5: Manual labelling effort – proposed (adaptive) vs fixed threshold approach.....	107

IV. Declaration

I am the main author of the PAGE Format Framework and the Adaptive Glyph Clustering approach as presented in this work. I have received guidance and feedback from my supervisor and colleagues at the Pattern Recognition and Image Analysis (PRImA) Research Group to the extent common in scientific discourse.

The software tools described in section 7.4 were implemented by Christian Clausner as main developer with contributions on conceptual and algorithmic design level as well as testing from my side. His work on Aletheia as reference implementation for PAGE has provided input to subsequent releases of the format in particular with regard to graphemes and semantic labels.

The creation of datasets as presented in section 7.5 was a joint effort of current and past members of the PRImA Research Group. Similarly, competitions hosted by PRImA as presented in section 7.6 involved multiple members of the group as indicated through the authorship of the corresponding publications.

V. Symbols

In	Input data
$G(i)$	Glyph image of glyph i
n	Number of glyphs in input data
$P(x, y)$	Pixel
xd_i	Width of glyph image $G(i)$
yd_i	Height of glyph image $G(i)$
$P_{bw}(x, y)$	A Pixel in a black-and-white image
$P_g(x, y)$	A Pixel in a greyscale image
φ	Feature extraction function
F	Feature vector
F_j	A single feature (component j of feature vector F)
m	Feature space dimension
φ_j	A feature extraction method (for one particular feature)
δ	Distance function
δ_f	Feature based distance function
δ_t	Template matching based distance function
$P_s(x, y)$	A Pixel in the search image
$P_t(x, y)$	A Pixel in the template
xd_t	Template width
yd_t	Template height
xd_s	Search image width
yd_s	Search image height
c	Index variable ranging through x values in a template
d	Index variable ranging through y values in a template
k	Index variable ranging through x values of positions in a search image
l	Index variable ranging through y values of positions in a search image
δ_c	Combined distance function
w_t	Weight for template matching based error
w_f	Weight for feature based distance
s	Scaling factor for feature based distance
χ	A clustering function
Cl	A set of clusters (a clustering result)
C	One specific cluster
r	Number of clusters in a set of clusters
χ_{AVQ}	Adaptive Vector Quantisation clustering function
t	Clustering threshold
Cl_{res}	A specific clustering result
Cl_{gt}	Perfect clustering (i.e. ground truth)
Cl_{val}	A valid clustering (i.e. free of misclassifications but not necessarily optimal)
r_{res}	Cardinality of a specific clustering result (i.e. number of clusters)
r_{gt}	Cardinality of the perfect clustering result (i.e. number of clusters as per ground truth)

t_a	Appropriate clustering threshold
acc	Accuracy of a clustering result
A	Set of correctly clustered glyphs
err	Error rate of a clustering result
$dist_{IC}$	Intra cluster distance
$dist_{AIC}$	Average intra cluster distance across all clusters
$dist_{MIC}$	Maximum intra cluster distance across all clusters
$corr$	Sample correlation coefficient

VI. Abstract

This work presents solutions to the multifaceted problem of document encoding in the context of digitisation processes. It addresses aspects related to structure and layout on page level as well as mechanisms for representing underlying textual information.

The main contribution of this work is a format specification for page content, embedded in the PAGE (Page Analysis and Ground truth Elements) format framework. This page content format specification has seen wide uptake in the community and reached such popularity that it is now commonly referred to by the name of the whole framework – as PAGE or PAGE XML. The specification is described in detail from its inception through multiple iterations of improvements to the very mature state it has evolved into. A discussion of success factors documents the development of an entire infrastructure surrounding PAGE as well as its influence on other formats and the impact it has had on the research landscape.

The second and smaller contribution that is presented here concerns the problem of encoding unstandardised characters that are practically unrecognisable by current off-the-shelf OCR (Optical Character Recognition) systems. A novel approach based on Adaptive Glyph Clustering is described with the specific characteristic that it can run without user intervention by estimating required parameters automatically from the input. The usefulness of the approach is demonstrated through practical applications related to training data generation/semi-automatic OCR and faithful reproduction using document specific alphabets and fonts.

1 Introduction

In today's connected world there is an expectation that information of all types can be easily accessed online. For physical documents like books, newspapers and reports this is not possible per se. Major digitisation efforts try to address this problem by converting traditional media into electronic counterparts. Digitisation processes involve scanning of physical documents and subsequently the creation of a digital version. Image data alone allows for simple applications such as archives for browsing and viewing documents but is limited in terms of more advanced features like full-text search. Proper interaction with the actual content of digitised documents requires transformation into a computer accessible format first. This can either be done manually or with the help of text recognition software. In both cases there is a need for a specification that governs the electronic representations that are to be created.

The development of a comprehensive encoding scheme for digitised documents is the subject of this work.

1.1 Motivation

The magnitude of the problem of devising a universal encoding scheme becomes very clear when looking at the great variety of document types that can be found. Figure 1 shows only a relatively small selection of scanned document images and yet it is obvious that information in numerous presentation forms and on various levels has to be considered. This effect is compounded by all the different ways in which users may want to interact with the material such as accessing it on devices with small screens, carrying out full text search or asking questions about the content, only to mention a few.

Digital libraries and related applications need accurate data to support advanced use scenarios. Equally, precisely defined datasets are essential for research into new methods and the development of systems that can then feed into the aforementioned applications. To support research and development tasks, datasets need to include ground truth data in addition to scanned document images. Ground truth, sometimes called gold standard, is the ideal result or expected output related to a computational problem. As such it is of

paramount importance for training of new methods as well as objective performance evaluation to gauge progress.



Figure 1: Document images of different types [1]

Existing solutions to the problem of encoding digitised content on page level (as the basic unit of output from document imaging/scanning processes) have too many limitations such as lack of detail or missing features, exposing the need for a more tailored solution in order to support the wide range of use scenarios.

Pica Black.

And be it further enacted by the Authority aforesaid, That all and every of the said Etchiquer Bills to be made forth by virtue of this Act, or so many of them as shall from a B C D E F G H I K L M N O P Q R S T

Brevier Black.

And be it further enacted by the Authority aforesaid, That all and every of the said Etchiquer Bills to be made forth by virtue of this Act, or so many of them as shall from a B C D E F G H I K L M N O P Q R S T

Pica Roman.

And be it further enacted by the Authority aforesaid, That all and every of the said Etchiquer Bills to be made forth by virtue of this Act, or so many of them as shall from a B C D E F G H I K L M N O P Q R S T

Hebrew without Points.

Direction des D. ad.

And be it further enacted by the Authority aforesaid, That all and every of the said Etchiquer Bills to be made forth by virtue of this Act, or so many of them as shall from a B C D E F G H I K L M N O P Q R S T

To any human being, the above images will easily be discernible as instances of printed documents with characteristic typefaces (fonts) and layouts, even without knowledge of the particular languages and scripts. Furthermore, to anyone with expert knowledge in Document Image Analysis and Recognition it may also appear that the chosen examples are extremely hard if not impossible to recognise/process using current state-of-the-art systems. A major problem, for instance, is the presence of characters and symbols which are unknown to the recognition system. In such cases, one might obtain no result at all or worse, wrong results in the form of whatever characters the particular system knows and deems closest. Figure 3 shows this problem very impressively by overlaying the automatically recognised text¹ on top of the original document image.

¹ Using the Tesseract text recognition engine.

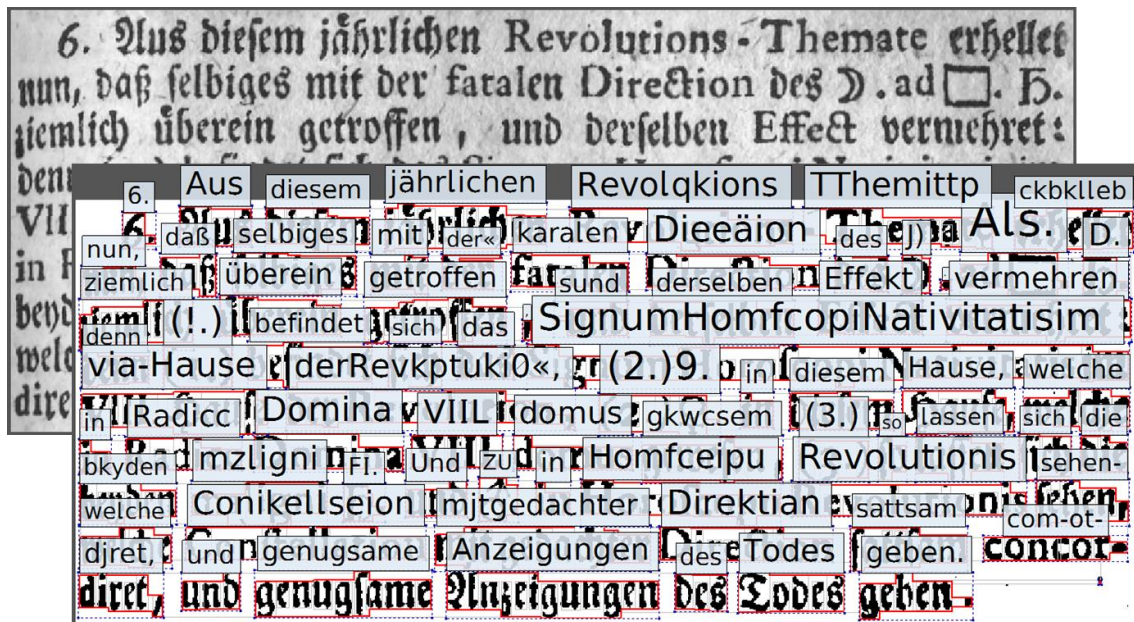


Figure 3: Text recognition result for an example of a historic document

This demonstrates the need for more generic approaches that are independent of OCR to identify potentially unstandardised characters with a view of encoding them properly and adding official support in the future. Given the volumes of material concerned it is important that the ability to fully automate processes is addressed.

1.2 Aims

The overall goal of this work can hence be described as finding a solution to the problem of encoding digitised documents. Other formats and approaches exist but have limitations as will be shown further on.

In particular, this work aims at a comprehensive format specification that can adequately represent information present in document images at the page level, to be stored in a computer accessible form. A holistic approach must be taken in order to cover various types of documents as well as support associated applications and tasks.

A secondary goal is the development of an approach that can automatically identify unstandardised characters which cannot be recognised using current OCR technology. Such an approach can then aid text encoding of content that could otherwise not be sufficiently represented.

1.3 Objectives

Translating the goals from above leads to two major strands of work with individual objectives.

1. Development of an encoding scheme for content on page level
 - Establish current state-of-the-art to inform further work (literature review)
 - Identify requirements on a new encoding scheme
 - Define design principles for a concrete format implementation
 - Implement the format specification
 - Document the format specification
 - Release the format specification
 - Iteratively improve the format specification and release updated versions
 - Assess the success/impact of the format specification
2. Development of an approach to automatically identify and encode unstandardised characters
 - Establish current state-of-the-art to inform further work (literature review)
 - Formalise the problem
 - Create a dataset for development and validation
 - Devise ideas for a possible solution
 - Empirically test the validity of possible solutions
 - Validate the ability to generalise
 - Synthesise results and propose a recommended approach
 - Evaluate overall results of the recommended approach using representative datasets
 - Demonstrate usefulness of the proposed approach through concrete application scenarios

The intended outcome of Strand 2 should be a recommended approach with evidence of its feasibility in the sense of a proof-of-concept. It is explicitly not intended to create an optimised method that is ready for integration into a production environment. Also excluded is large-scale evaluation which would go beyond the available resources.

1.4 Methodology

The approach followed in this work can to some extent already be deduced from the objectives above. There are two main strands related to a general encoding scheme and the handling of unstandardised characters. Both start with literature-based investigations to establish current state-of-the-art and to confirm the assumed gaps this work is aiming to fill. The individual approaches will then diverge for the two problems.

1.4.1 Encoding Scheme

Development of a new format specification is a combination of domain specific research and more general software engineering. While requirements need to be first gathered with regard to the field of document digitisation it is then necessary to derive design principles to match relevant technologies and to follow software engineering standards.

The implementation of the actual format specification will be executed as an iterative process with a first release and later updates based on feedback and experience from use.

Evaluation of the outcomes will be of a qualitative nature given the open character of the addressed problem. The criteria that will be considered are related to the adherence to existing technical standards and the validity of source code as well as to general success and uptake of the format specification. To this end it will be attempted to gather as much evidence of real-world impact as is realistically possible.

1.4.2 Unstandardised Characters

The second problem is of a more theoretical nature and more deeply invested in computer science. The starting point is the formalisation of the problem, introducing the concept of glyph clustering. Empirical studies involving a dataset of glyph images from real-world documents are then carried out to observe the behaviour of characteristic metrics. In the next stage, a potential solution fitting the observations is to be formulated. The ability of this solution to generalise is to be confirmed using additional datasets. This form of validation is of quantitative nature, using corresponding metrics such as distance measures, error rate, compression and correlation. A full evaluation of the proposed method with regard to achievable overall results, expressed through error and compression rates as well as possible gain over an traditional approach, will be presented on representative datasets (full details are provided in Section 6).

The usefulness of the finally proposed solution is to be investigated through real-world applications. This will serve as further qualitative validation.

The main dataset used during development is derived from a collection of scanned historical documents (originals). In the following it will be referred to as German Fraktur (o). Figure 4 shows an example page from this dataset.

Nach vorgesehener Tabell ist bald im Anfang des Decembris, und dann sub medium, der Mercurius über 30. Zoll/und zwar dorten um $1\frac{1}{2}$ Linien gestiegen; hingegen war sein tieffester Stand just am H. Christ-Tag/nehmlich 29. 1. der uns dann das am 2. und 3. Seyertag erfolgte stürmische Schnee-und Regen-Wetter ganz sicher prognosticirte. Im Thermometro kam der Liquor am tieffesten am 5. und zwar auf 45. an welchem Tag des Morgens ein gar starcker Reiffen gefallen. Die folgende Tage bis zum 14. waren von passabler Kälte/von dar aber nahm selbige immerzu ab/wie wir denn den 13. mehr nicht als 15. gradus frigoris gehabt/die auch bis auf den 26. nicht höher als auf 27. gestiegen; Dannenhero kein Wunder/wann wir in den H. Weyhenacht-Ferien gar gelind/ und wie es das Barometrum auch anzeigte/ meistens unfreundlich und regnerisches Wetter gehabt/ auffser am Heil. Christ-Tag/da sich des Nachmittags der lieblichste Sonnenschein gezeigt.

Figure 4: Example page from the dataset of original document scans

Pages like this were segmented into individual glyphs by annotating corresponding region outlines using Aletheia (a document image analysis software developed by the PRIMA Research Group, details in Subsection 7.4). After entering the actual text, individual glyph images (snippets) could be exported together with their true character class as labels. Figure 5. shows such labelled examples which constitute ground truth for the considered problem.



Figure 5: Examples of labelled glyph images

This process of creating datasets with ground truth information leads to representative results but is unfortunately very time-consuming. In order to generate larger amounts of

test data for validation, the idea of synthetic examples can be utilised. Synthetic images can be produced from encoded text in born-digital documents by rendering the actual glyphs using a variety of scripts and fonts. Ground truth in the form of character class labels can then be automatically obtained from the electronic text. To this end, additional synthetic datasets were created:

- Greek (s)
- Arabic (s)
- German Fraktur (s)
- Cyrillic (s)
- Hungarian Runes (s)
- Georgian (s)
- Latin Arial (s)
- Thai (s)

Even though the used characters in these synthetic datasets are not actually "unstandardised" as per the studied problem, they can still be used in lieu of more originals from scans as they will have been unstandardised at some point in the past (before standards such as ASCII and Unicode emerged).

1.5 Structure of the Thesis

This thesis is organised in nine chapters which are followed by references and an appendix. Background and Related Work are discussed directly following this Introduction. Format Requirements and Design Principles are presented in Chapter 3, followed by Chapter 4 Encoding Page Content which describes the implementation of the format specification in detail. Chapter 5, titled Evolution of the Pagecontent Format Specification, addresses further iterations of the format. The issue of unstandardised characters is then covered in Chapter 6. Chapter 7 contains the overall Discussion with a focus on success, impact and application of the findings. Chapter 8 wraps the work up with final Conclusions and Future Work.

1.6 Summary

Digitisation of existing physical documents has been identified as an important issue. Linked to it is the question of how digitised documents should be encoded with an appropriate level of detail and accuracy to facilitate a wide range of application scenarios. In addition, there is the question of how previously unknown character sets may be handled. Two strands of work with individual objectives were defined to provide answers to these questions. The overall methodology that will be followed in these two strands of work was presented, including a discussion of related validation approaches and datasets.

2 Background and Related Work

The general theme of this work is about transforming text contained in images into a machine-readable (encoded) representation. In the wider sense, solutions to this kind of problem can already be found in US patents of the early 19th century related to reading aids for the blind and input devices for telegraphy [2]. The development of text recognition systems in the more modern sense (i.e. for processing information), however, started only with the advent of computer technology and scanning devices in the 1950s [3]. Today, software for automatic text recognition is commonly referred to as OCR or Optical Character Recognition.

The broad field of document encoding, specifically with regard to digitised documents, will be discussed in the following. Starting with text recognition approaches and underlying technologies, it will be moved on to different ways for representing digitised documents. Aspects to be considered in representation formats will be derived from document image analysis methods before investigating concrete standards and specifications. Finally, the issue of clustering and symbolic compression which is of particular relevance to handling previously unstandardised characters will be discussed.

2.1 Text Recognition

Originally, the term OCR stood for the actual problem of assigning a predefined code (maintained in code tables covering specific alphabets) to a pixel pattern representing a single character in a digital image. Following [4], mainly two general approaches have emerged for this task.

The first is based on comparing the input pixel pattern, which is to be recognised, with stored pixel patterns (also called prototypes) for all supported characters. For every combination a matching or conversely an error value is calculated and then the character code of the prototype with the best match (lowest error) is assigned. The calculation of the matching/error value typically involves counting the number of co-occurring/differing pixels in input and prototype image. For colour and greyscale images this can also be extended to factor in the magnitude of deviation. Advantages of this approach are its simplicity and the fact that it can produce good results for specifically trained fonts. On

the negative side, it has to be stated that this approach cannot handle documents with mixed font types, sizes and effects very well, due to the requirement of having a prototype for any combination of suchlike font peculiarities ready.

The second main approach is based on universal characteristics (called features) rather than specific pixel patterns in order to achieve independence from concrete fonts. Such features can be characteristics which are quite obvious to the human eye, like the number of components (e.g. the letter “i” being composed of two), intersection points (as in the letter “X”) or of a less evocative and more mathematical nature (such as the frequency of a specific pixel texture). The recognition task can then be formulated as a classification problem related to objects represented by their multi-dimensional feature vectors.

Successful solutions to this problem can be based on various machine learning approaches such as neural networks, decision trees, support vector machines etc. trained to separate instances of characters that belong to different classes. There can be a one-versus-rest classifier for every character class or one integrated multi-class classifier (which is less common). Training (the learning phase) of a classifier involves the finding of an optimal model that operates on the features. Typically, this is based on a number of examples with known character class (training data) and the goal is to arrive at a model that allows correct classification of all elements in the training data as well as the ability to generalise (i.e. correctly classify new data as well) which is to be confirmed with the use of separate test data. During the actual character recognition (classifier application) the trained model is applied on the features vectors of input characters to achieve the desired allocation into the different character classes.

Over time the usage of the term OCR has changed and nowadays it is typically used as a synonym for whole Document Image Analysis and Recognition systems (sometimes also referred to as Reading Systems), incorporating a range of pre-processing, segmentation, layout analysis, classification, recognition and post-processing steps [5], [6], [7], [8].

2.2 Representation of Digitised Documents

As an alternative to full OCR, there are also systems focusing on the imaging and/or management aspects of scanned document without employing any form of further analysis or recognition. While image servers and basic digital library systems are very

useful for making content at least publicly available in a form that can be consumed by human readers there are numerous limitations inherent to such systems (like the lack of text search features, no means for re-formatting and no way of re-purposing content). In [9] it is for instance argued that “... serious obstacles prevent merely imaged documents within DLs [Digital Libraries] from playing all the useful roles that encoded documents do, or even many of the roles that their original physical embodiments did.” Nevertheless, such systems have their place as OCR is far from being perfect and does not always produce usable results, especially on documents with complex layouts and content, previously unseen fonts, scripts, and character sets. Evidence for this is given in [10] where OCR evaluation results are presented for Europeana Newspapers, one of the most recent, truly large-scale digitisation projects [11]. With the nature of the material being historical newspapers, the reported figures paint a very good picture of how current state-of-the-art OCR systems can, on the one hand, produce fairly usable results (in the region of 90% accuracy with regard to bag of words² evaluation) for modern and well supported languages and fonts but, on the other hand, almost fail on older and more uncommon material (e.g. with a bag of words accuracy as low as 36% for Yiddish material which is virtually unusable in any use scenario requiring full text)³.

A way forward for documents which are only in part recognisable can be a hybrid approach. One such solution is supported by Adobe’s PDF format in the form of partially searchable documents [12]. The idea behind the format is to store information in two layers – one containing the scanned image which is presented to the reader and a second hidden text layer which might not be complete and/or may contain errors in the recognised text. This way a (partial) search can be enabled while recognition errors are not (immediately) apparent to the reader. Clever as the approach is, it is mostly masking the underlying imperfections. It still presents a general improvement in handling documents with hard-to-recognise content but systems that go further by leveraging regular structures and patterns without necessarily having to recognise all content first would be more desirable.

² Bag of words evaluation is a slightly less-strict approach which disregards the precise order of words by “putting all words in a bag” and only checking if all instances were correctly found.

³ In [10] it is argued that at least 70% bag of words accuracy is required for the material to be used in real-world applications like a digital library system offering key word search.

In [13] the view is taken that “OCR results should be viewed as an ‘annotation’ of the document image, not as the definitive representation” and that the various steps along the way to a fully encoded document can add value and open new ways to interact with it – especially if OCR cannot achieve acceptable recognition rates. It is for instance advocated that properly segmented word snippets, even though not fully recognised, can be used to reflow text and therefore support different layouts for specific target viewing devices.

Such a system, specifically designed for transforming content to be displayed on small handheld devices, is described in [14]. While being a step in the right direction there are a number of limitations associated with the presented system. Firstly, the system does not achieve a truly encoded representation (i.e. a code assigned to every character) which would allow for more sophisticated editing and/or searching of content. Moreover, graphical reproduction is achieved via word image snippets (bitmaps) taken from the original which limits the potential for producing new layouts and reflowing (e.g. no true hyphenation possible) as well as improvements in terms of accessibility (such as unlimited zooming levels which vectorised formats would enable). Also, the authors do not propose a solution to the problem of clustering word/glyph prototypes which would be required in order for the system to run autonomously (without any user input) in a real-world application and to achieve a high level of symbolic compression of the data (as opposed to storing bitmap images for every word/glyph instance).

A system similar to the approach presented above, yet specifically targeted at generating output formats for e-book readers, is described in [15]. The authors write, however, that “Future improvements of the approach should consider automatic solutions to fix the above values [parameters related to the clustering method] on the basis of statistics computed from the connected components in the pages to be converted.” So again, with regard to the clustering problem, no solution is proposed to ascertain the relevant parameters automatically (based on the input data), thus not allowing this approach to work completely independently.

2.3 Document Image Analysis

Document image analysis systems depend on numerous individual processing steps which have relevance to the general problem of document encoding and in particular what information should be included in a format specification.

Segmentation is a crucial task in almost any image analysis system as it extracts regions of interest from an input image in order to guide further processing. The nature and level of the segmentation approach depends on the envisaged application. For document images it is common to carry out segmentation on region [6], [16], [17], text line [18], word [19] and character/glyph level [20]. In terms of the underlying approaches there are several types of methods. Top-down methods start from the whole image and depending on certain criteria split it further into smaller segments [21], [22]. Conversely, bottom-up algorithms start from smaller entities (typically pixels and/or connected components) and aggregate them into more complex structures such as characters, words, text lines etc. [23], [24], [25]. Hybrid methods can then be seen as a combination of top-down and bottom-up [26]. Another hybrid approach, which is described as global, analyses the background in order to segment objects on the page [27].

Region classification is often linked to segmentation (sometimes jointly referred to as page layout analysis) as its main purpose is to label coherent regions in terms of their content type and/or function. A solution to the problem of identifying text regions, which are of particular interest for the proposed work, is presented in [28].

Established activities, such as the ICDAR⁴ Page Segmentation Competition series, show how the state-of-the-art has developed and that current systems can achieve good success rates (above 90%) even on historical documents [29], [30], [31], [32].

2.4 Document Encoding Standards and Format Specifications

With regard to the first strand of work considered in this thesis, there are only few established formats related to text recognition and encoding in general use. PDF [12], as was already mentioned before, is typically used for a fixed/final representation in archiving and/or digital library scenarios.

⁴ International Conference on Document Analysis and Recognition

The ALTO format, maintained by The Library of Congress [33], is perhaps the most-widely used standard for storing OCR results and driving publication systems with more sophisticated features (such as text search with highlighting of matching sections). At the time when the work presented here was started, ALTO was limited in terms of structural levels (no representation of characters or glyphs) and not equipped to express intricate logical structures such as complex reading orders as they occur in newspapers. As will be shown later, the work presented in this thesis has in fact influenced the development and future releases of ALTO.

Another format in the realm of OCR systems is hOCR [34]. Being a micro-format, it mostly aims at output presentation, in particular the capability to be displayed in a browser without further conversion. As a consequence, it has a rather reduced document model which limits its usability for further applications.

TEI (Text Encoding Initiative) [35] is a format that must naturally also be mentioned in relation to the encoding of text. It has wide applications in manually curated collections especially in the field of digital humanities. Its main background is related to the study of corpora and thus the actual content structure can be considered the driving force. For the representation of digitised documents, however, it is more natural to follow the physical appearance, layout and structure of content elements as they appear on a printed page.

Specifically targeted at semantic information is DoCo (Document Components Ontology) [36]. It provides a structured vocabulary so that structural and rhetorical document components can be described. Due to its main purpose of expressing the meaning of content elements it does not offer features to describe the information relevant at lower levels of a digitisation process such as physical layout and attributes.

DJVU [37] was developed by AT&T in the late nineties and specifically designed for the storage of digitised documents. It featured advanced compression and graphical reproduction capabilities but did not define a detailed content model. Today, development of the format is discontinued.

CoCo [38] is a specific format for the representation of ground truth related to layout analysis, based on JSON (JavaScript Object Notation). It follows a very minimalistic approach and only supports simple bounding boxes for region outlines.

Another format that is relevant in the wider sense is SVG (Scalable Vector Graphics) [39] which is typically used for displaying graphics in browsers. As such it is not specific to OCR or digitised documents but may be used as a convenient output format due to its widespread support.

2.5 Clustering and Symbolic Compression

The second strand of work considered in this thesis will investigate ways for identifying unstandardised characters in an input stream. This requires the set of unique instances (i.e. classes) to be ascertained first. Once found, the input can be encoded solely by means of referencing a prototype for each class. This follows the basic idea of symbolic compression as presented in [40]. Crucial for this to work is the ability to cluster the set of all glyphs into homogeneous groups (i.e. the different character classes). As such, “Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters).” [41].

Many different clustering approaches have been proposed over the years, including K-means, k-Nearest Neighbours, Simulated Annealing, Self-Organising Maps, and Genetic Algorithms [42], but the problem of determining the right number of clusters (or internal parameters that will result in the right number of clusters in the output) is intrinsic to all of them and can typically only be solved on application level, i.e. related to a concrete use/data scenario. For example, related to K-means it has been stated that “Automatically determining the number of clusters has been one of the most difficult problems in data clustering. Most methods for automatically determining the number of clusters cast it into the problem of model selection. Usually, clustering algorithms are run with different values of K [number of clusters]; the best value of K is then chosen based on a predefined criterion.” [43]. A particular difficulty related to the problem considered in this work is the requirement to achieve clustering results that are completely free of misclassifications. Existing approaches for estimating the right number of classes (or, depending on the

approach, the right internal parameters) that try to minimise (rather than strictly avoid) errors within clusters will therefore not be applicable.

Also, the particular problem of glyph clustering will lead to a potentially huge number of elements to be processed, prohibiting any extensive search/brute force approaches. A way forward may be the application of a method derived from vector quantisation [44], [45] which allows for sequential processing of input elements. This, however, requires a specific similarity measure and a new approach for ascertaining/tuning the threshold parameter related to starting a new cluster as the algorithm progresses.

For the representation of document specific fonts, as will be discussed later in a possible application scenario of the work on encoding unstandardised characters, it is necessary to look also into methods for glyph vectorisation.

Vectorisation in this context is the conversion of objects contained in bitmap⁵ images into a mathematical description of their shapes (based on vectors). Numerous methods for the vectorisation of drawings and images have been proposed in the literature [46]. If applied only to properly segmented images of glyphs rather than any type of content (preceding segmentation and classification steps have to make sure of this), it is possible to tune the vectorisation method specifically to this task and thus obtain good results [47], [48], [49]. For integration into a proof-of-concept application scenario there are various existing tools that can be readily used [50], [51].

2.6 Summary

Digitisation of physical documents has become a very important topic for which various solutions with varying degrees of complexity and features have been developed over the years. The different approaches that have emerged, such as recognition based, imaging based as well as hybrid were discussed with a deeper analysis of related standards and format specifications.

With regard to the first strand of work as per the project objectives, it can be concluded that while many of the presented formats have certain features that are relevant to the problem addressed in this work, there is none that would support all requirements for

⁵ Also referred to as raster or pixel map

encoding of digitised documents as set out in Section 3.1 of the following chapter. In particular, there is a general lack of precision, level of detail and attributes specific to page content as is required for a sophisticated format capable of storing ground truth.

As for the second strand of work, existing approaches for glyph clustering were discussed as basis for the development of an approach to automatically identify unstandardised characters in digitised documents. However, as there is no specific solution readily available that would fit the requirements arising from the problem at hand, it will be necessary to develop a new approach that is informed by existing work but provides a solution to the application specific challenges.

3 Format Requirements and Design Principles

The solutions presented in later parts of this work are based on a continuous and thorough analysis of the requirements on a format for accurate and comprehensive encoding of digitised documents which was stated the main aim of this work. Design principles had to be derived accordingly to guide the development process with regard to implementational choices.

In the following it is described what specific requirements were gathered and from which sources they originated. The design principles that were guiding the actual development of the new format are then discussed. Finally, it is explained, how the new format specification fits into a bigger framework.

3.1 Requirements

Requirements for the format under development were gathered from a number of different sources. Prior work such as documented in [52] was first of all investigated. Most valuable input came from technical and library partners in the large-scale digitisation projects IMPACT [53], Succeed [54] and ENP [11]. In addition, experience gathered through implementation of a reference software, using the specification developed within the work presented here as native storage format, was translated into further requirements (see section 7.4 for tool support).

The following high-level requirements were initially identified:

- Support information storage for a wide range of digital library applications (OCR output)
- Support information storage for performance analysis applications (ground truth)
- Support information storage for training of document image analysis methods (ground truth)
- Support contemporary as well as historical documents
- Support a wide range of methods such as
 - Segmentation
 - Layout analysis
 - Region classification

- Text recognition
- Border removal
- Image deskewing
- Support a wide range of content elements with corresponding attributes
- Support precise region outlines in the form of arbitrary shapes
- Support text information on all relevant levels
- Implement ability to represent complex reading orders
- Implement a layers concept to support disambiguation of overlapping regions
- Allow customisation and application specific attributes

A continuous backlog of all low-level requirements, translated into corresponding feature requests with priority levels and comprehensive descriptions was kept in Mantis, an open-source issue tracker. The full list of requirements had hundreds of entries and would go beyond the scope of this thesis.

3.2 Design Principles

The following design principles were established to guide the actual format implementation:

- Document instances are to be stored in XML (eXtensible Markup Language) [55].
- The format specification is to be implemented as an XML Schema using XSD (XML Schema Definition).
- Namespace and format version are to be directly linked. Any changes affecting backwards compatibility are to be implemented under a new namespace. Backwards compatibility is thereby to be considered from both sides – document instances as well as software.
- The namespace is to indicate the publication/release date.
- Schema files are to be publicly hosted at <https://www.primaresearch.org>.
- Any new versions of the format are to be accompanied by detailed release notes, outlining steps required for migration.
- The first and foremost goal of the format is to record layout information and text at page level. While comprehensive support for a wide range of region types is

aspired, it is accepted that there will be cases necessitating additional information to be linked (via external mechanisms).

- Every element is to be addressable i.e. have an ID attribute (required or optional depending on context).
- All regions of interest (representing an object with physical dimensions) are to be expressed as flexible shapes via polygons or polylines defined by a list of points (as opposed to limiting the representation to bounding boxes for instance).
- The hierarchy of text content elements are to cover all major levels i.e. text regions, text lines, words and glyphs and it should be possible to store encoded text independently on every level.
- Recognised standards for text encoding are to be followed (Unicode).
- Recognised standards for language and script names are to be followed (ISO).
- The format specification is to define an explicit entry point i.e. exactly one root element.
- XSD type definitions are to be used for reusable components to avoid redundancy and achieve a clear code structure.
- XML element structure and logical document structure are to be kept separate. The element structure is to follow the physical layout of the page. Logical structure (such as reading order) is to be expressed via region references.
- A comprehensive documentation is to be maintained within the source code (using XSD annotation and documentation elements).

3.3 The PAGE Format Framework

The format specification for page content as presented in the following chapters is embedded in a wider structure – the PAGE (Page Analysis and Ground truth Elements) format framework [56]. From the start, the PAGE format framework included a *root* specification for representing the general structure of data files related to specific document analysis workflows/performance evaluation experiments as well as four task-specific *gts* (ground truth and storage) data formats corresponding to different processing steps and methods. In a later release a third category *eval* (for data formats linked to evaluation approaches and tools) was added.

The base URL for all schemas is: <https://www.primaresearch.org/schema/PAGE>. The schema location and namespace are kept in sync for all releases. This is to aid XML parsers that can auto-fetch XSD files for validation. In 2017 PAGE was also released on GitHub to open it up for input from the community: <https://github.com/PRImA-Research-Lab/PAGE-XML> (under Apache License 2.0)

3.4 Summary

This chapter has laid the foundations for the work on implementing the proposed format which will be presented in the following. Features of the new format will be based on the requirements that were gathered. Concrete design decisions and implementational solutions will follow the principles that were developed here. This is expected to lead to a stable and widely usable format specification for encoding of digitised documents which was stated the main aim of this work.

4 Encoding Page Content

The following describes the implementation of the pagecontent format specification in the form of an XML schema definition. It was first released in 2009 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2009-03-16>. Its full specification as well as subsequent releases are attached as Appendix for reference. The implementation follows the general design principles as described in Section 3.2.

First, the root element as entry point to the larger structure is introduced. This is followed by the all the types that define the various features implemented by the format. This ranges from general metadata, elements representing the physical as well as logical structure, all the way to the specific region types with their extensive ranges of attributes used for describing content in digitised documents.

4.1 Root Element

The pagecontent schema defines exactly one element at the root level (to be used by all pagecontent XML instances) with all further element and attribute definitions organised in appropriate types (see Figure 6). Internally, the namespace prefix *pc* (short for pagecontent) is used to distinguish the application specific types from general XML Schema syntax elements.

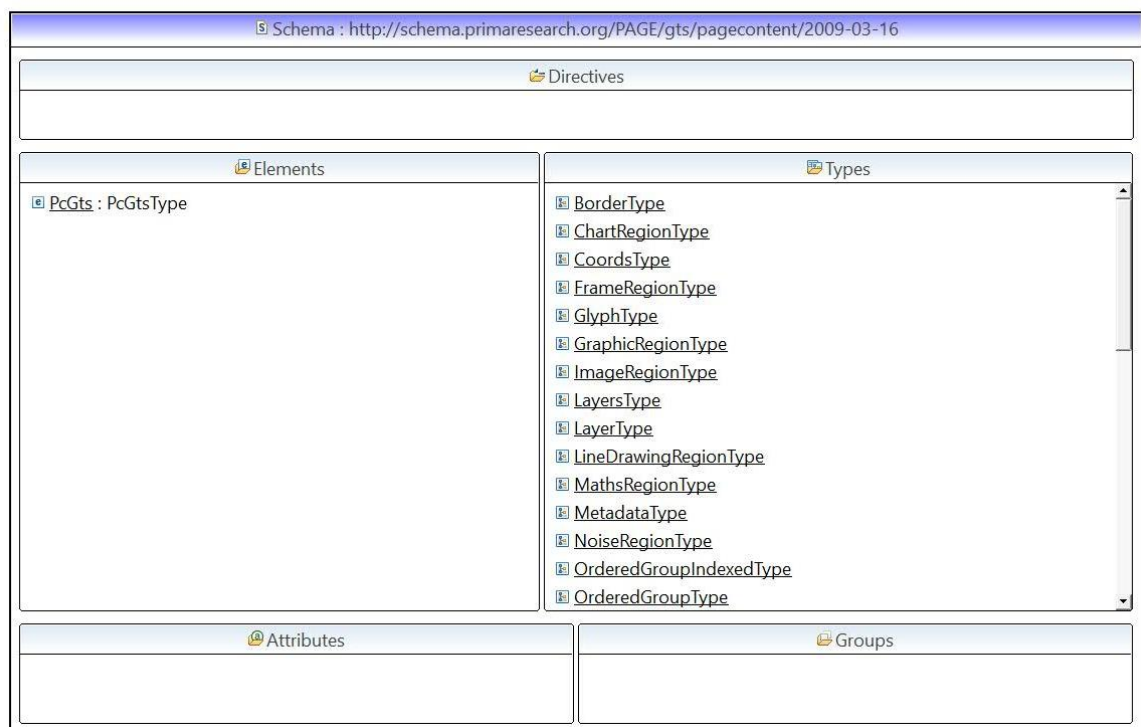


Figure 6: Overview of the pagecontent schema

Every pagecontent instance starts with a PcGts element at root level as shown in Figure 6. The name is derived from “Page content” in the “Ground truth and storage” section of the format framework.

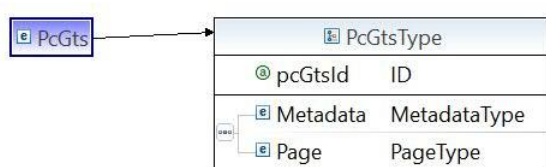


Figure 7: PcGts (root) element

The actual definition of the PcGts element is given in PcGtsType as per Figure 7.

4.2 PcGtsType

PcGts has one optional attribute “pcGtsId” to store a document id. This is followed by a sequence of exactly two child elements: Metadata and Page (Figure 8).

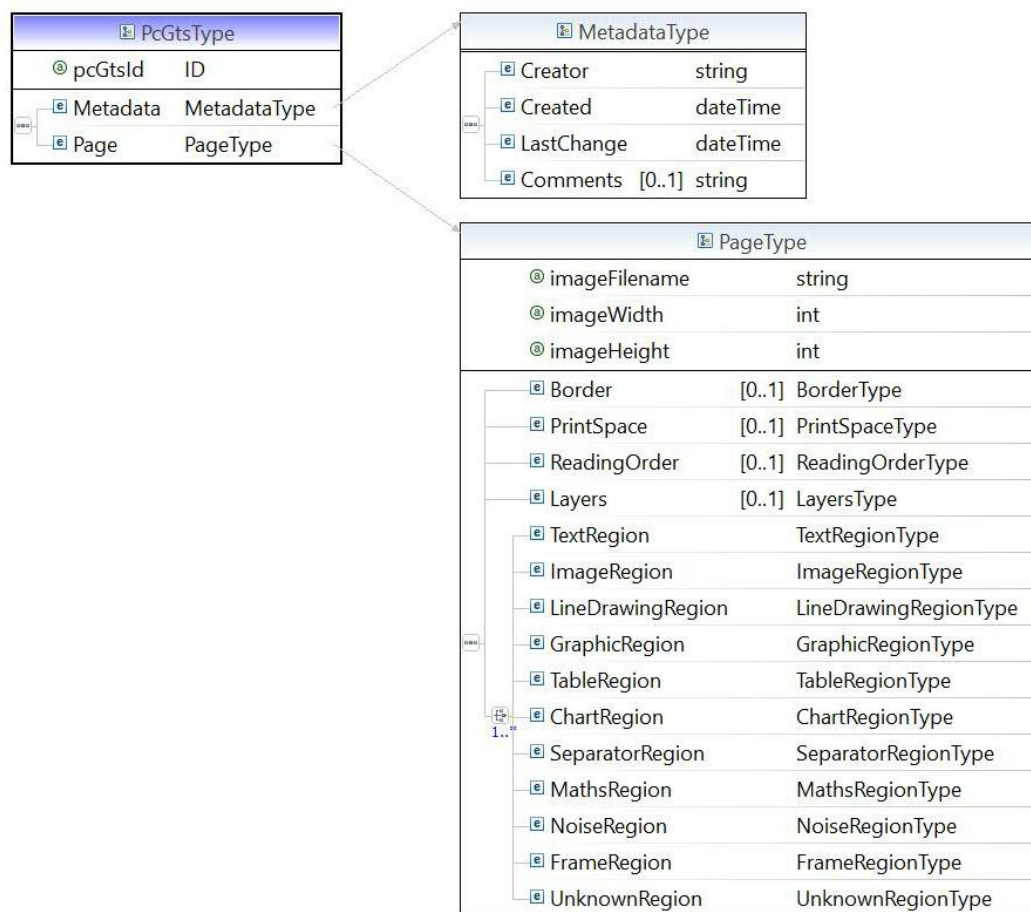


Figure 8: PcGtsType with Metadata and Page elements

Metadata is for storing additional information about the document and Page contains all actual content such as layout information and recognised text.

4.3 MetadataType

The metadata section in the first release is very brief and only covers the very basics. This follows the idea that more detailed information would be stored in dedicated (external) files using appropriate standards for metadata representation. The elements Creator (type string), Created (type dateTime) and LastChange (type dateTime) are mandatory (exactly one), while Comment (type string) is optional.

4.4 PageType

Page elements have three attributes which are all required. “imageFilename” is used to store the name of the reference image (to which all information in the pagecontent instance relates). “imageWidth” and “imageHeight” are used to record the dimensions of

the reference image and thus the area of the pagecontent instance. While these could also be derived from the actual image it is advantageous to store them explicitly in case the (external) reference image is not accessible.

The actual elements inside a Page element are organised as a sequence of Border, PrintSpace, ReadingOrder and Layers which are all optional, followed by one or more content regions out of the available pool of region types:

- TextRegion
- ImageRegion
- LineDrawingRegion
- GraphicRegion
- TableRegion
- ChartRegion
- SeparatorRegion
- MathsRegion
- NoiseRegion
- FrameRegion
- UnknownRegion

The particular types with their individual structure and attributes are discussed in the following.

4.5 BorderType

Scanned document images often contain parts that do not belong to the actual page due to the hardware and setup involved. The Border element allows to explicitly define a polygon that separates any background that may be visible in a scanned image from the true page area. The border outline is stored as a polygon defined by coordinates as per CoordsType (Figure 9).



Figure 9: BorderType

The feature may be used by image enhancement tools for background removal and/or as ground truth for evaluation tools measuring the performance of background detection/border removal methods.

4.6 CoordsType

Various content elements defined in the pagecontent format specification depend on geometric information. The basic type to encode shapes such as polygons or lines is given in CoordsType (Figure 10) and allows for any number of individual Point elements to be contained. The interpretation of the shape with regard to closed (polygon) or open (polyline) follows from the context in which it is used. Region outlines for instance are considered to be closed i.e. the last point being connected to the first.

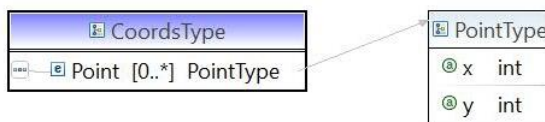


Figure 10: CoordsType

Individual Point elements are defined by two required attributes “x” and “y” of type integer. This first implementation is based on explicitly storing points in their own elements which allows direct access through XML mechanisms (such as XPath expressions) but can be memory intensive for complex shapes. The representation of coordinates has been changed in a future release in favour of a more memory-efficient solution (see following sections).

4.7 PrintSpaceType

A Page element may have an optional PrintSpace child element as defined by PrintSpaceType (Figure 11). Its structure is identical to BorderType but the associated meaning differs.



Figure 11: PrintSpaceType

Print space determines the effective area on a printed page. Typically, its size is equal for all pages of a book with some exceptions such as title pages or multipage pictures. The print space contains all living elements like main body, footnotes, headings, and running titles. Outside of the print space are typically elements such as page numbers (unless part of a running title), marginals, signature marks and preview words.

4.8 ReadingOrderType

The order of content objects in a document image is a very important aspect related to text recognition and correct serialisation of elements. While potentially trivial in book pages, it can be highly complex in other types of documents such as newspapers or magazines.

The starting point for the implementation of a reading order structure is that the order of elements in an XML instance is not reliable as it is not guaranteed that each parser will read in siblings in the same order. Also, consecutive elements in an XML instance could only be used to express simple sequences and not more intricate structures. Therefore, the pagecontent format specification introduces an explicit reading order definition with advanced structuring mechanisms.

A Page element may have an optional ReadingOrder child element as defined by ReadingOrderType (Figure 12). This is a purely logical structure which works with grouping elements and region references.

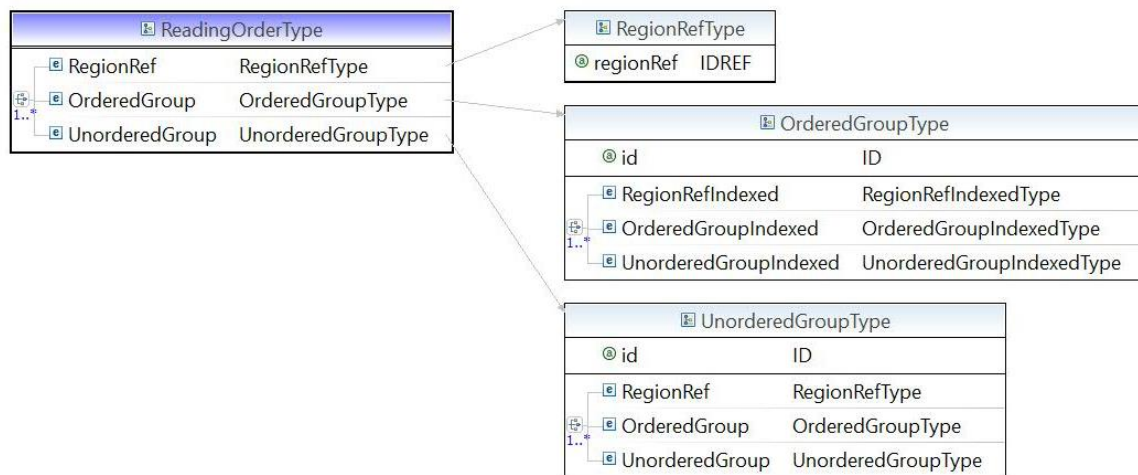


Figure 12: ReadingOrderType

RegionRef elements have one required attribute “regionRef” of type IDREF (i.e. a reference to an existing ID in the current XML instance). The referenced ID is expected to belong to a region element. As default at page level, without further grouping, referenced regions are considered to be unordered on the page (in the same way as defined in the UnorderedGroupType).

If a concrete order between elements is to be expressed, then they must be included in an ordered group. Groups may contain further groups to create nested hierarchies (recursive definition).

4.8.1 OrderedGroupType

OrderedGroupType (Figure 13) defines a container for explicitly ordered element references. An OrderedGroup element is required to have an “id” attribute of type ID and may have any number of RegionRefIndexed child elements. If nesting is required, then it can also have OrderedGroupIndexed and UnorderedGroupIndexed child elements.

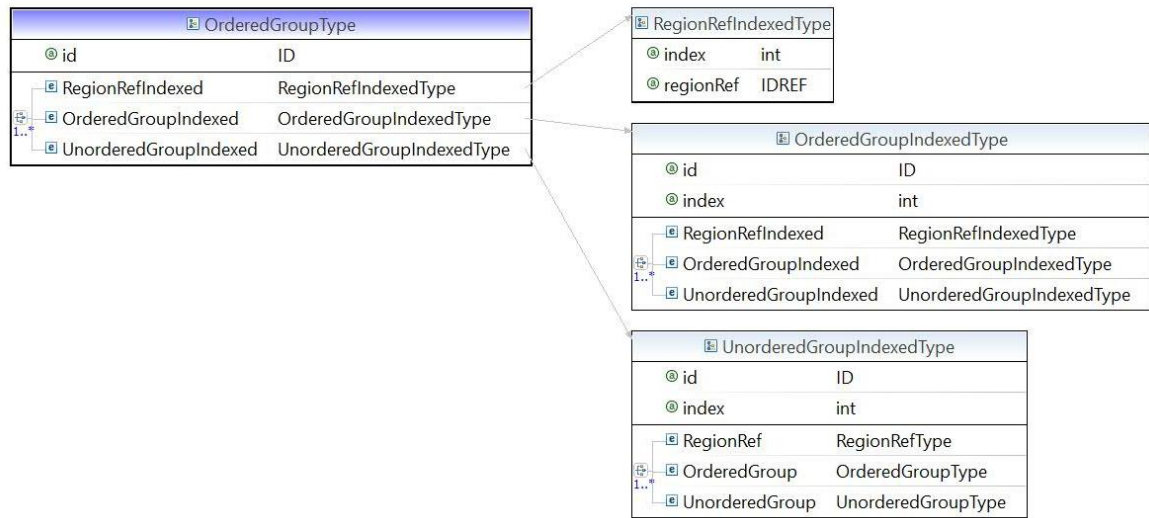


Figure 13: OrderedGroupType

The differentiation between indexed and non-indexed elements is necessary to enforce the use of an “index” attribute for recording positions where required. Elements with smaller values in “index” are interpreted as coming before elements with greater values. All child elements inside an ordered group must be of an indexed type so as to express their explicit reading order.

4.8.2 OrderedGroupIndexedType

OrderedGroupIndexed elements are containers for references of ordered elements. At the same time, they are members of (i.e. inside) an ordered group above with their position defined via the required “index” attribute.

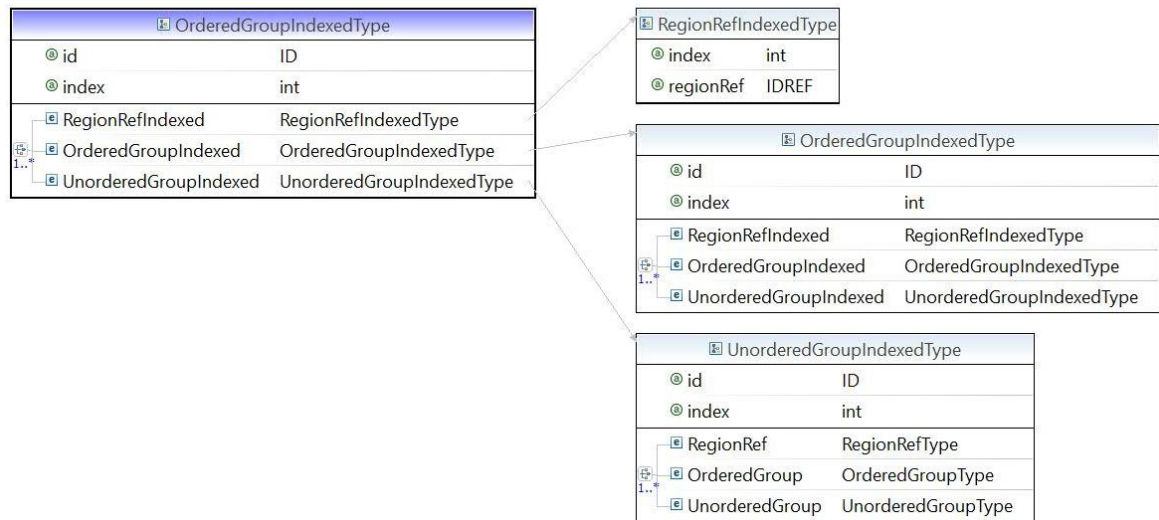


Figure 14: OrderedGroupIndexedType

The structure of OrderedGroupIndexedType (Figure 14) is identical to OrderedGroupType (Figure 13) with the exception of the additional “index” attribute.

4.8.3 UnorderedGroupIndexedType

UnorderedGroupIndexedType (Figure 15) defines the second group element type allowed inside an ordered group. Like the ordered counterpart it has a required “index” attribute to indicate its position.

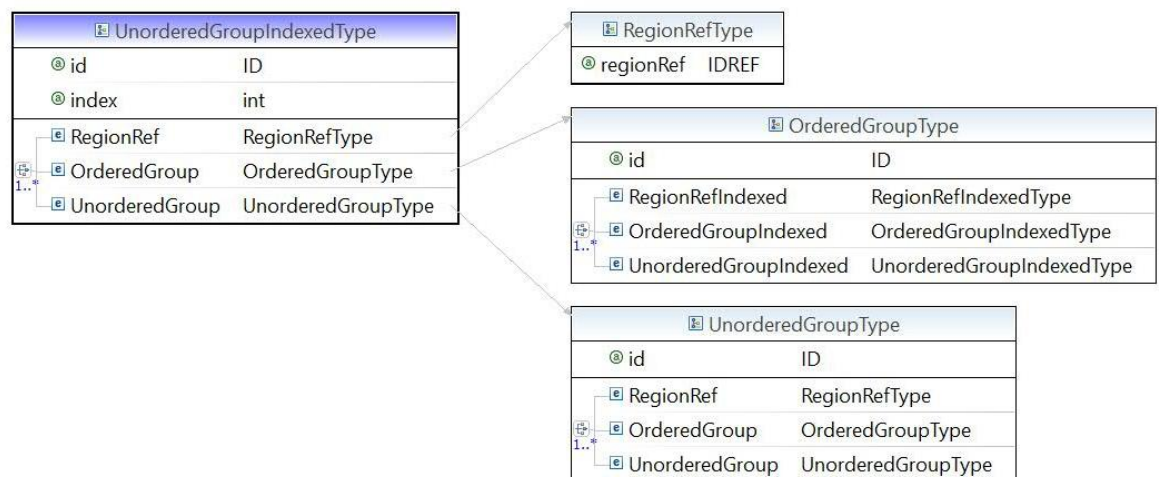


Figure 15: UnorderedGroupIndexedType

The difference is now that elements contained inside are treated as unordered i.e. must be of a non-indexed type.

4.8.4 UnorderedGroupType

UnorderedGroupType (Figure 16) completes the four possibilities arising from the two group and member types each. As such it represents a member of an unordered group that itself also contains unordered members inside.

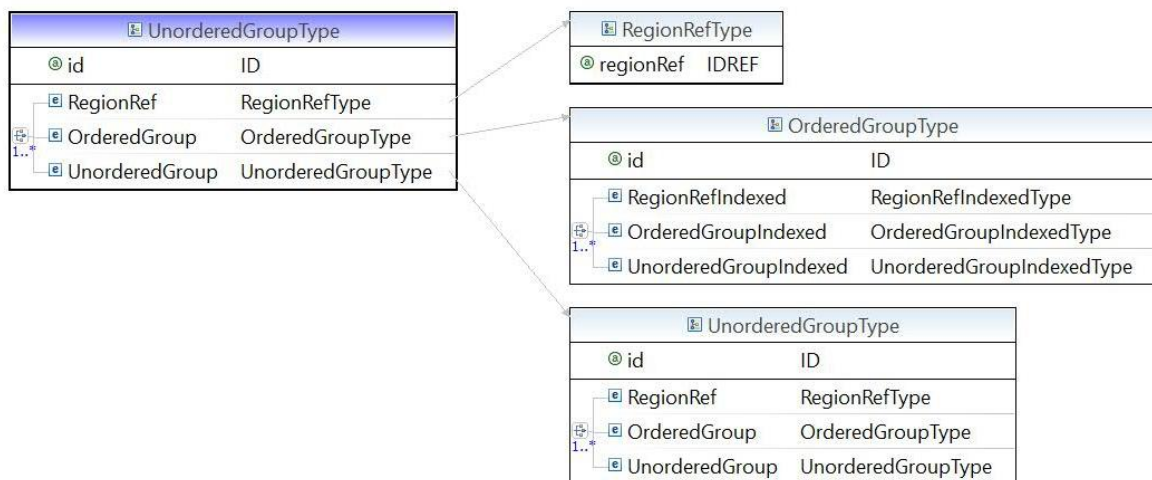


Figure 16: UnorderedGroupType

This type does not involve any index values and resembles the same general structure as ReadingOrderType (which is unordered by default) with the addition of an id which allows it to be placed inside a complex hierarchy of groups and regions.

4.9 LayersType

Layers are a mechanism that can be used for disambiguation of regions that overlap in the two-dimensional representation of standard bitmap images (x/y-plane) by introducing a third dimension.

A Page element may have an optional Layers child element as defined by LayersType (Figure 17). As reading order, this is a purely logical structure (without any direct graphical representation) that works with references but does not allow further grouping and is thus less complex.

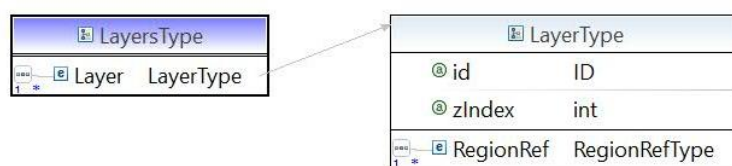


Figure 17: LayersType

If present, the Layers element may contain one or more Layer elements as defined by LayerType. Of particular interest here is the “zIndex” attribute which is used to express a third dimension i.e. stack order of elements. Its interpretation follows the conventions of other formats such as CSS (Cascading Style Sheets). An element with a greater zIndex is considered to be in front of another element with lower zIndex, potentially occluding it (where foreground pixels are actually present). RegionRef elements use the same type as defined before in the context of reading order structures.

4.10 TextRegionType

The first in the list of available region type elements is TextRegion. In many use scenarios related to OCR this will be the most important one in that it can provide a container for all textual information. The TextRegionType (Figure 18) defines various attributes that are specific to text regions and allows the actual encoded text on region level to be recorded in a TextEquiv element (the name being derived from text equivalent). The region outline is to be stored using a Coords element as introduced before. From a structural point of view, text regions can be described further by explicitly providing (any number of)

TextLine elements as children.

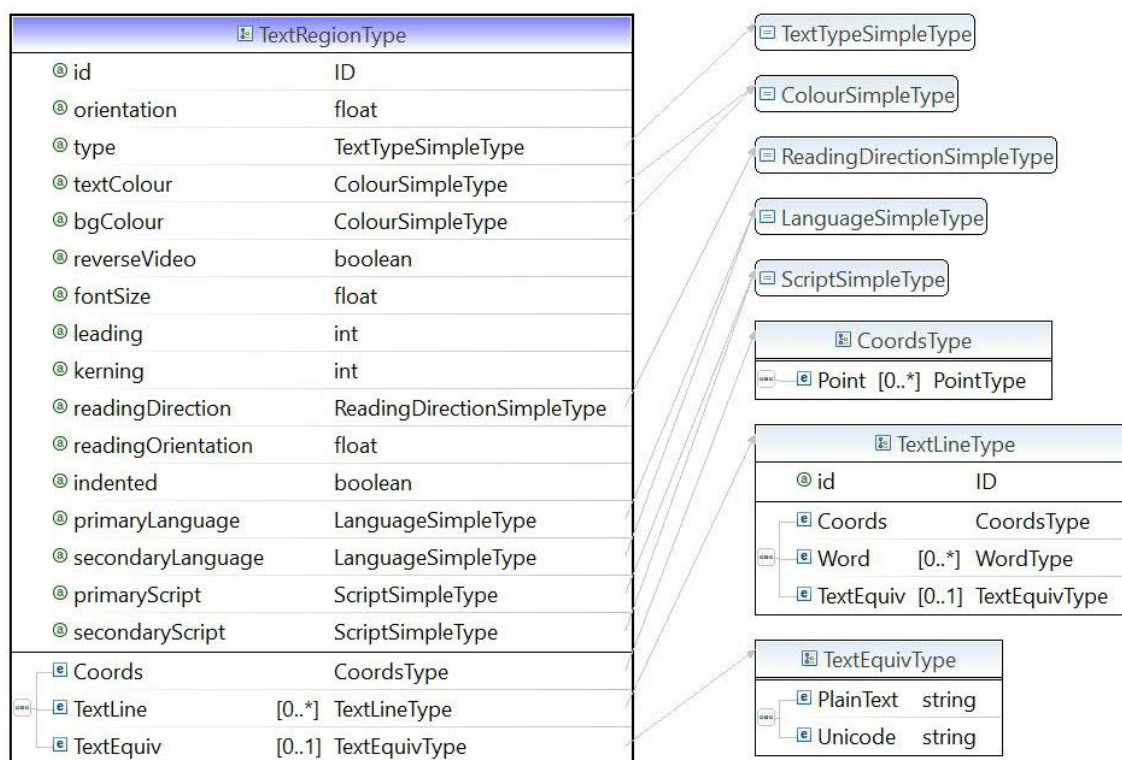


Figure 18: TextRegionType

TextEquiv has two child elements: PlainText and Unicode. PlainText may be used for storing text in a simplified form (ASCII or extended ASCII as mostly used for typing) following specific conventions. This is to acknowledge common practice where text such as special characters may be transliterated when being entered. Typical examples are ligatures which are technically one glyph but are often entered as two separate entities (ſt vs st). As such practice may differ significantly between users and institutions it is recommended to use the Unicode element for a more faithful representation. The expectation is that text stored in the Unicode element only makes use of the appropriate code points as defined in the Unicode standard. The scenario where a glyph is not (yet) defined in Unicode is specifically discussed in Chapter 6.

TextRegionType defines the following attributes to describe the individual characteristics:

- id – required, type ID, for referencing regions
- orientation – optional, type float, individual skew angle of the region

- type – optional, possible values
 - paragraph
 - heading
 - caption
 - header
 - footer
 - page-number
 - drop-capital
 - credit
 - floating
- textColour, bgColour – optional, enumeration type of named colours
- reverseVideo – optional, type Boolean, indicating reversed foreground/background colours
- fontSize – optional, type float
- leading – optional, type int, space in points between lines
- kerning – optional, type int, space in points between characters
- readingDirection – optional, possible values
 - left-to-right
 - right-to-left
 - top-to-bottom
 - bottom-to-top
- readingOrientation – optional, type float, clockwise angle
- indented – optional, type Boolean, indicates whether a region is indented
- primaryLanguage, secondaryLanguage – optional, enumeration type of language names
- primaryScript, secondaryScript – optional, enumeration type of script names

4.10.1 TextLineType

If a greater level of detail is required, then it is possible to express the inner structure of text regions via TextLine elements as defined in TextLineType (Figure 19).

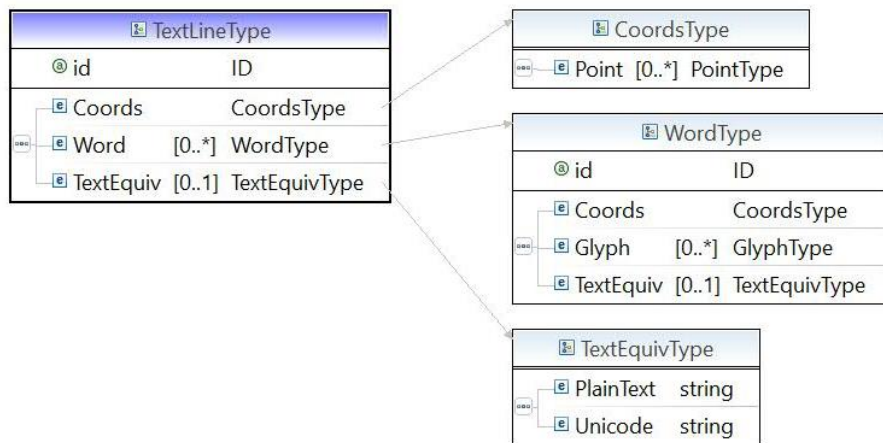


Figure 19: TextLineType

Each TextLine element must have an “id” attribute and a Coords child element representing its outline. Following the same approach as before, an optional TextEquiv element can be used to store the actual text content on text line level.

TextLine elements may have any number of Word elements as children.

4.10.2 WordType

Progressing deeper in the hierarchical structure of text components, corresponding Word elements can be used to store encoded text and outlines on word level.

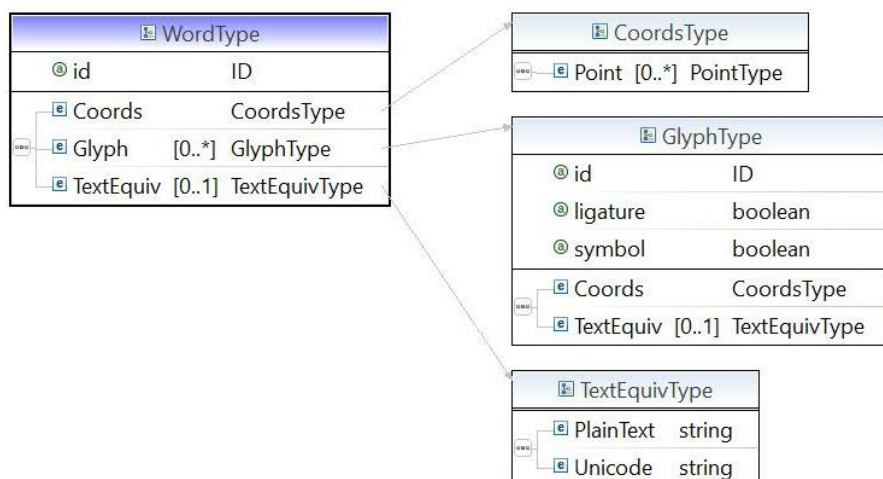


Figure 20: WordType

The structure given in WordType (Figure 20) is the same as that of TextLineType (Figure 19) with the only difference that Glyph elements are now allowed as children representing the level below.

4.10.3 GlyphType

Glyph elements use the same child elements for outline and actual text as the other text related types. In addition to the common features, GlyphType (Figure 21) adds two more Boolean attributes to describe glyphs further. The optional “ligature” attribute can be used to indicate that the current glyph is a ligature i.e. a typographic unit composed of more than one base character. Similarly, the optional “symbol” attribute can be used to indicate that the current glyph is a symbol i.e. a character not representing a simple letter or number.

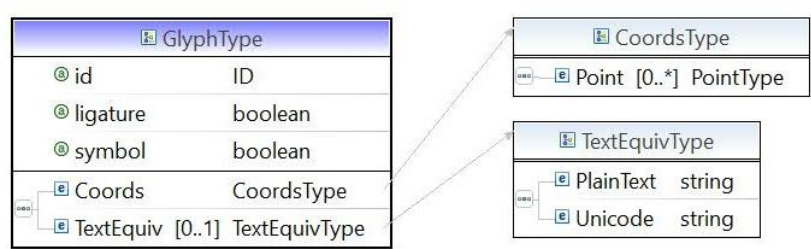


Figure 21: GlyphType

Glyph elements constitute the lowest text hierarchy level in this first pagecontent release. This is revisited in a later version of the pagecontent format specification.

4.11 ImageRegionType

Next in the list of available region type elements is ImageRegionType (Figure 22). ImageRegion elements may be used to indicate areas on a page representing image information such as photographs. An image is considered to be more intricate and complex than a graphic. As defined by all region types, ImageRegion must have an “id” attribute and outline information given in a Coords element.

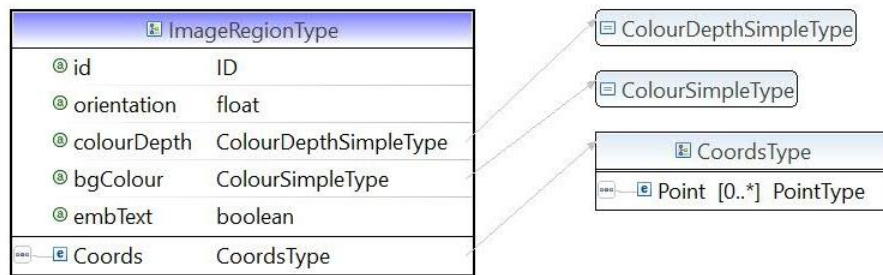


Figure 22: ImageRegionType

ImageRegionType defines the following specific attributes:

- orientation – optional, type float, individual skew angle of the region
- colourDepth – optional, allowed values: bilevel, greyscale, colour (this is only intended as a general indication with regard to the three main types, not a concrete bit depth)
- bgColour – optional, enumeration type of named colours
- embText – optional, type Boolean, indicates whether the image also contains text

4.12 LineDrawingRegionType

LineDrawingRegion elements as defined by LineDrawingRegionType (Figure 23) can be used for (potentially colour) illustrations and drawings that typically do not contain solid areas. The general structure is as ImageRegionType but with partially different attributes.

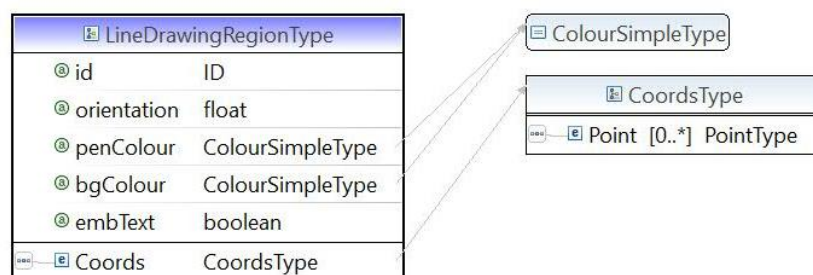


Figure 23: LineDrawingRegionType

LineDrawingRegionType defines the following specific attributes:

- orientation – optional, type float, individual skew angle of the region

- penColour, bgColour – optional, enumeration type of named colours
- embText – optional, type Boolean, indicates whether the line drawing also contains text

4.13 GraphicRegionType

GraphicRegion elements as defined by GraphicRegionType (Figure 24) may be used for content parts that have a graphic representation and are of relatively low complexity in comparison with images. Examples are company logos, letterheads as well as some other visible effects resulting from interaction with the paper (such as stamps or punch-holes).

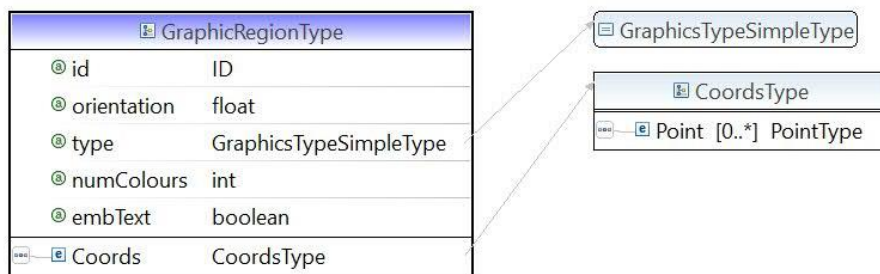


Figure 24: GraphicRegionType

Specific attributes defined in GraphicRegionType:

- orientation – optional, type float, individual skew angle of the region
- type – optional, possible values
 - logo
 - letterhead
 - handwritten-annotation
 - stamp
 - signature
 - paper-grow
 - punch-hole
 - other
- numColours – optional, type int, approximation of colours used in the region
- embText – optional, type Boolean, indicates whether the graphic also contains text

4.14 TableRegionType

TableRegion elements may be used to indicate areas representing tabular data. Rows and columns may or may not have separator lines – these lines are not considered separator regions as specified below. TableRegionType (Figure 25) follows the common region definition with some additional attributes allowing for a very basic structural description.

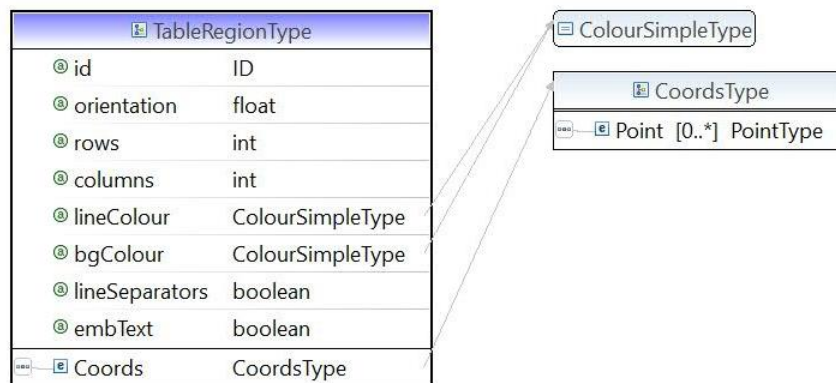


Figure 25: TableRegionType

Specific attributes defined in TableRegionType:

- orientation – optional, type float, individual skew angle of the region
- rows – optional, type int, number of rows present
- columns – optional, type int, number of columns present
- lineColour, bgColour – optional, enumeration type of named colours
- lineSeparators – optional, type Boolean, indicates if visible lines are present
- embText – optional, type Boolean, indicates whether the table also contains text (which is very likely for this type of region but not guaranteed)

In this first release, table regions are mostly defined as placeholders (outline plus attributes). Rows and columns are defined as optional elements to accommodate use scenarios where the inner structure is not known (recognised/annotated). Tables were revisited in a following release as more requirements become apparent.

4.15 ChartRegionType

ChartRegion elements as defined by ChartRegionType (Figure 26) are intended for content parts relating to graphical representations of data or functions such as charts or graphs.

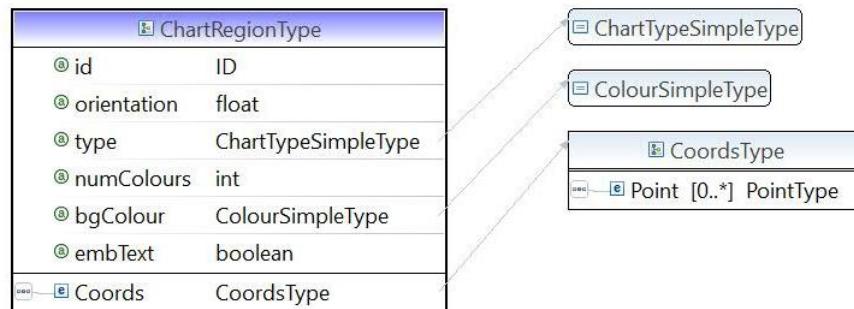


Figure 26: ChartRegionType

Specific attributes defined in ChartRegionType:

- orientation – optional, type float, individual skew angle of the region
- type – optional, possible values
 - bar
 - line
 - pie
 - scatter
 - surface
 - other
- numColours – optional, type int, approximation of colours used in the region
- bgColour – optional, enumeration type of named colours
- embText – optional, type Boolean, indicates whether the chart region also contains text

4.16 SeparatorRegionType

SeparatorRegion elements as defined by SeparatorRegionType (Figure 27) may be used to describe graphical cues indicating logical separation of other content parts. Typical examples are vertical lines between columns and horizontal lines between paragraphs.

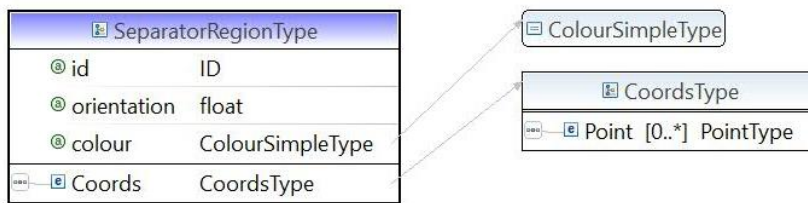


Figure 27: SeparatorRegionType

Specific attributes defined in SeparatorRegionType:

- orientation – optional, type float, individual skew angle of the region
- colour – optional, enumeration type of named colours

4.17 MathsRegionType

MathsRegionType (Figure 28) defines MathsRegion elements which can be used to describe self-contained content parts representing equations and mathematical expressions.

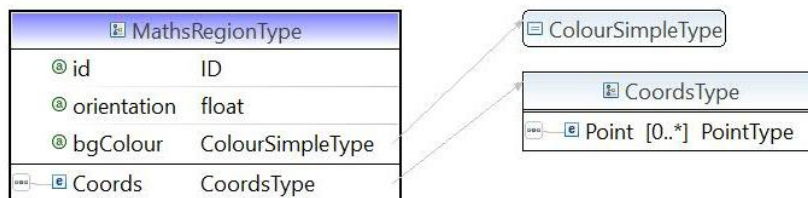


Figure 28: MathsRegionType

Specific attributes defined in MathsRegionType:

- orientation – optional, type float, individual skew angle of the region
- bgColour – optional, enumeration type of named colours

At this point, maths regions are defined as placeholders (outline plus attributes). General nesting of regions as introduced in a later release will allow for the inner structure and content to be described as well.

4.18 NoiseRegionType

NoiseRegion elements provide a mechanism for indicating areas on a page with some form of visual representation but no real information as manifested in various types of artifacts relating to either the physical document or the scanned image. The main purpose of recording noise regions is to exclude them from further processing and/or remove them as part of general image enhancement.

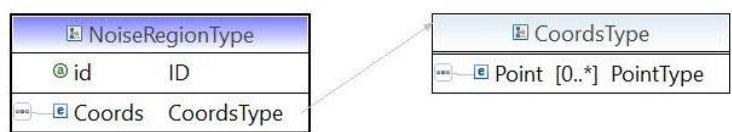


Figure 29: NoiseRegionType

NoiseRegionType (Figure 29) does not define any specific attributes beyond “id” which is shared by all regions.

4.19 FrameRegionType

FrameRegion elements are the only regions that allows nesting of other regions. In this first release, recursive nesting is only possible with more FrameRegion elements. Other types of regions that are included will automatically result in leaf nodes (i.e. prevent further nesting, not having the ability themselves). Typical use cases are decorative frames surrounding other regions or graphical elements for grouping multiple regions.

FrameRegionType (Figure 30) defines the common region features (id and outline) and in addition allows any number of the available region types as child elements.

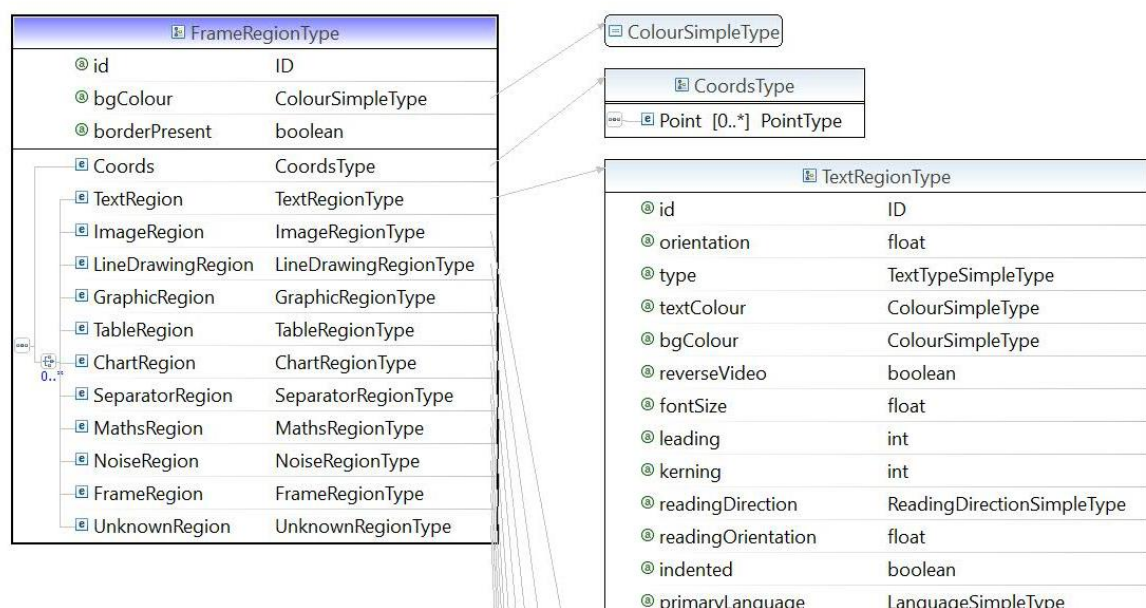


Figure 30: FrameRegionType

Specific attributes defined in FrameRegionType:

- bgColour – optional, enumeration type of named colours
- borderPresent – optional, type Boolean, indicates whether a visible border is present

The issue of nested regions will be revisited in a future version of the format.

4.20 UnknownRegionType

The last region type introduced in this first pagecontent format specification is intended to cover any cases that have not been specifically considered and/or situations where the region type cannot be ascertained.

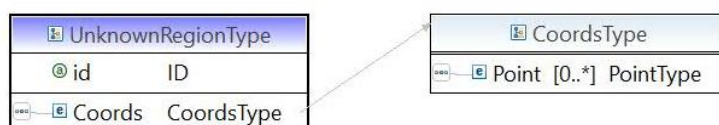


Figure 31: UnknownRegionType

UnknownRegionType (Figure 31) does not define any specific elements or attributes beyond what is shared by all regions.

4.21 Summary

This chapter described the initial implementational solution to the first strand of work on developing an encoding scheme for content on page level. It was published as the first version of the pagecontent format specification in the form an XML schema definition. The general structure, logical constructs as well as all specific region types with their numerous attributes were presented. The detailed sources of this (and the following) iterations of the format specification are attached in the Appendix.

5 Evolution of the Pagecontent Format Specification

Following the first version of the pagecontent format specification, further improvements and additions were made and published in subsequent releases. In the first instance, these were informed by the experiences gathered within the PRImA Research group [57] in relation to development of software tools that use the format natively. PRImA Research, of which the author of this thesis is a member, is located at the University of Salford (UK) and focuses on topics related to document analysis and recognition. Further input on additional features and requirements were obtained through the involvement in major projects related to digitisation efforts such as IMPACT [53], SUCCEED [54], ENP [11] and Census 1961 [58].

5.1 Second Release

The second pagecontent format specification was released in 2010 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-01-12>. Although this release can be seen as a minor update it was still given a new (and unique) namespace to avoid potential problems with backwards compatibility.

The main change in this version is related to a limitation that was discovered in cases where the pagecontent format was to be used for storing page border or print space information only. This can for instance apply to methods solely targeted at background removal or print space detection without the need for any further document layout information.

The simple solution to address this problem was to define the number of content regions as completely unconstrained i.e. `[0..*]` rather than requiring at least one region in a valid pagecontent document (Figure 32).

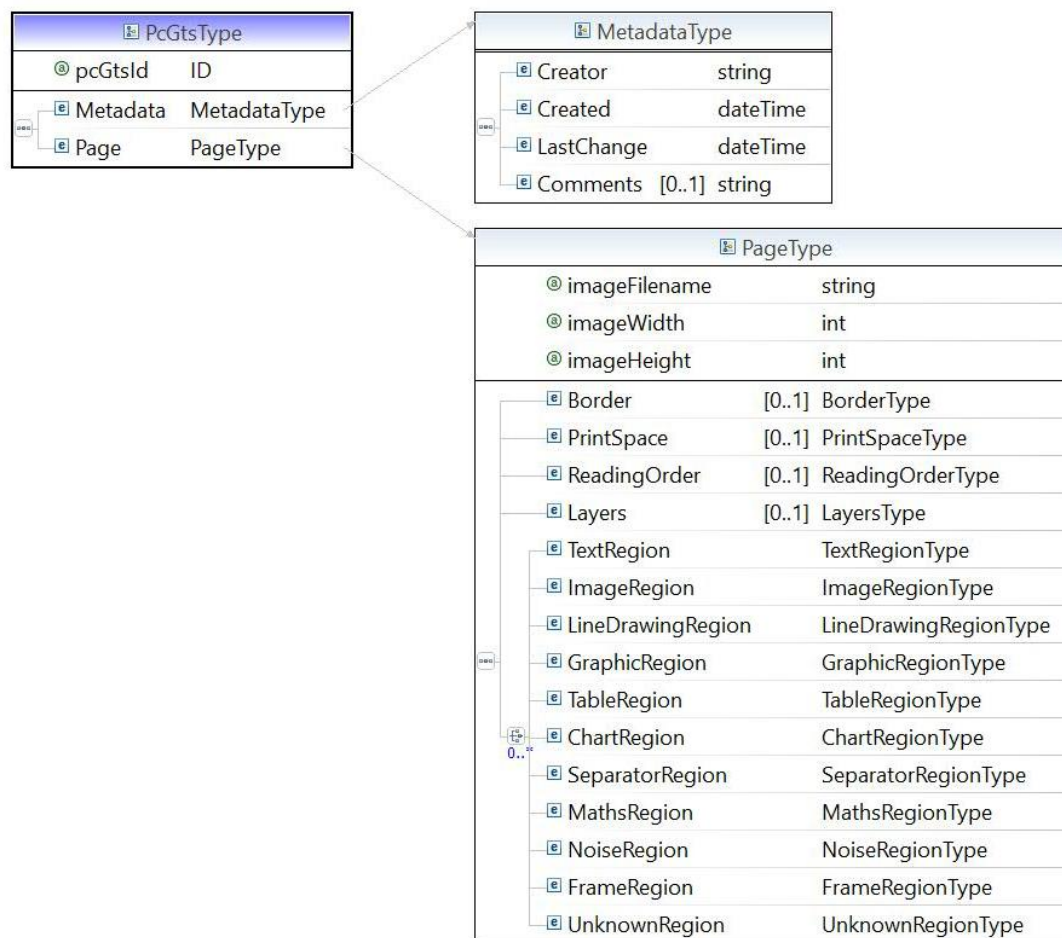


Figure 32: PageType with unrestricted number of regions

Converting existing documents to the new format only requires changing the namespace and corresponding schema location given that the old format is a subset of the new one.

In software tools the new schema needs to be explicitly implemented as the previous assumption of finding at least one region in a valid pagecontent files is no longer valid.

5.2 Third Release

The third pagecontent format specification was released a short time later in 2010 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-03-19>. This release can be considered a service update containing smaller improvements following more intensive use of the format.

5.2.1 ReadingOrderType

The original reading order definition started from the convention that any regions that were referenced as direct children of the ReadingOrder element were implicitly considered unordered. This was intended for convenience of use but later deemed a potential source of confusion as well as making the schema definition unnecessarily complex. By removing this special case, ReadingOrderType (Figure 33) was simplified and made more explicit at the same time. The new format now prescribes exactly one child element for page level ReadingOrder, denoting the nature of the contained objects (OrderedGroup or UnorderedGroup). Actual region references (RegionRefIndexed, RegionRef) are now only allowed inside explicit group elements.

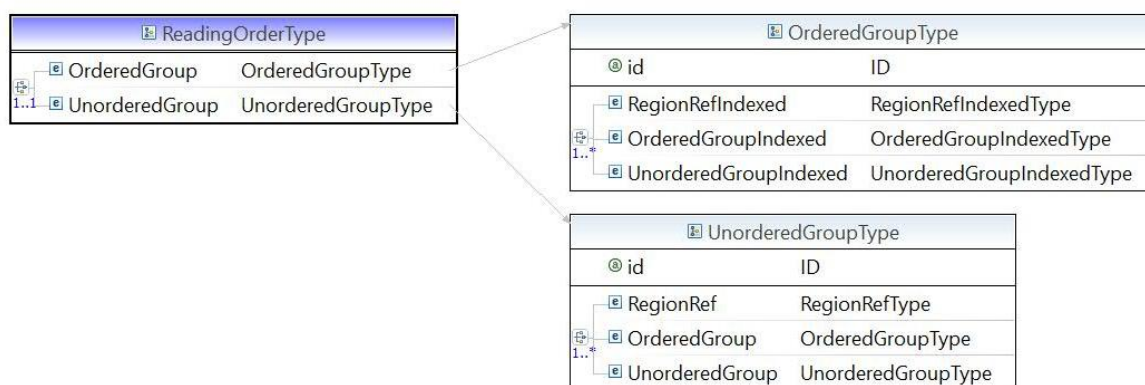


Figure 33: Updated ReadingOrderType

To convert existing documents and/or to achieve the old behaviour, it is always possible to introduce an UnorderedGroup element as child of ReadingOrder and then to place the existing structure underneath.

Tool support needs to respond to this slight change in structure but going forward may be easier to handle as there is now going to be a regular structure throughout and no special case is to be dealt with anymore.

5.2.2 TextTypeSimpleType

The initial format specification provided a number of predefined (sub) types for text regions. Following input from libraries and technical partners involved in the IMPACT project, more values were added to the controlled vocabulary of TextTypeSimpleType:

- signature-mark
- catch-word
- marginalia
- footnote
- footnote-continued
- TOC-entry

This particular addition does not require any changes/conversion of existing pagecontent documents as all previous values which may have been used before are carried over.

Tool support will have to consider this change in type constraints as these additional values may now appear in new documents.

5.3 Fourth Release

The fourth release marks a major update and milestone reflecting additions and improvements associated with work carried out within the scope of the Europeana Newspapers project as well as feature requests from the wider research community (for instance relating to handwritten documents). It was published in 2013 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2013-07-15>. Due to the extent of the changes, it is not backwards compatible with previous versions. This requires existing documents to be converted and tools to be updated in order to take advantage of the new specification.

5.3.1 CoordsType

The first change is related to a more efficient representation of coordinates as used for region outlines (polygons). Before, point elements were stored as children of Coords. While this allowed direct access to coordinates from within the XML document (e.g. via simple XPath expressions) it also required significant memory for the explicit element structure. The new solution makes use of a single points attribute attached to the Coords element as defined in an updated CoordsType (Figure 34).

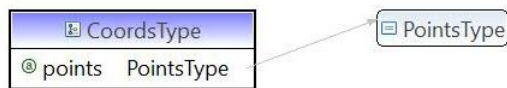


Figure 34: Updated CoordsType

The attribute follows a new PointsType with the purpose of constraining the string pattern to sequences of comma separated x,y coordinate pairs with at least two points (i.e. at least one line segment) using a regular expression of $([0-9]^+, [0-9]^+)+([0-9]^+, [0-9]^+)$.

This change requires existing documents to be converted as well as tools to be updated to support the new point representation. In turn, this will reduce the storage space of documents as well as the memory needed by tools due to a smaller DOM (Document Object Model) structure.

5.3.2 General Region Nesting

As a further major change, this release introduces the ability for all regions to be nested. Before, this feature was limited to FrameRegion elements. General nesting is crucial for detailed and accurate representation of deeply structured elements such as tables.

In terms of the pagecontent schema this is now implemented through a new abstract RegionType (Figure 35) from which all specific region types inherit. This includes the ability to have any number of (sub) regions as child elements. The use of an abstract RegionType has additional benefits such as a better structured and more easily readable schema with less repetition by separating common and individual region features.

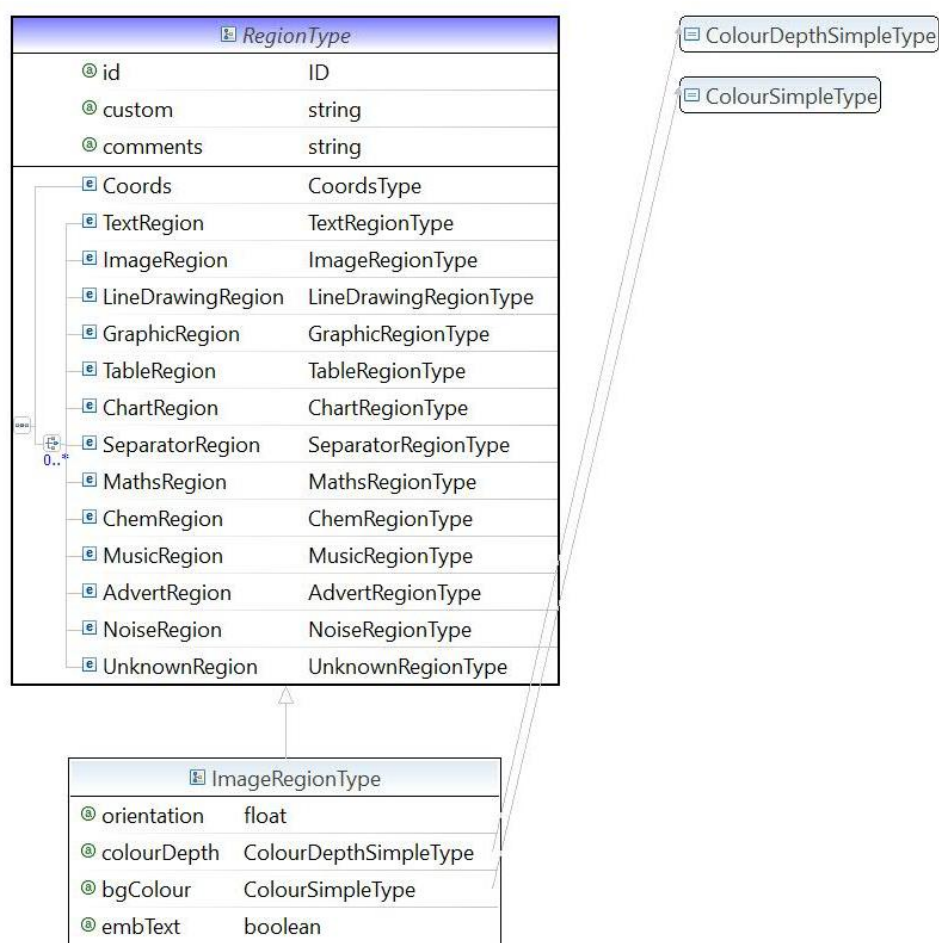


Figure 35: Abstract RegionType (here extended by ImageRegionType)

The dedicated type of frame regions from before has been removed as it is not needed any longer (with basically every region being able to function as a frame). In terms of migrating documents from previous format versions it is recommended to convert FrameRegion elements to GraphicRegion with type attribute “frame”.

5.3.3 Additional Region Types

The new format specification also introduces three new general region types to indicate the presence of adverts (Figure 36), chemical formulas and musical notation. Apart from the inherited region features all three define additional attributes for orientation and background colour.

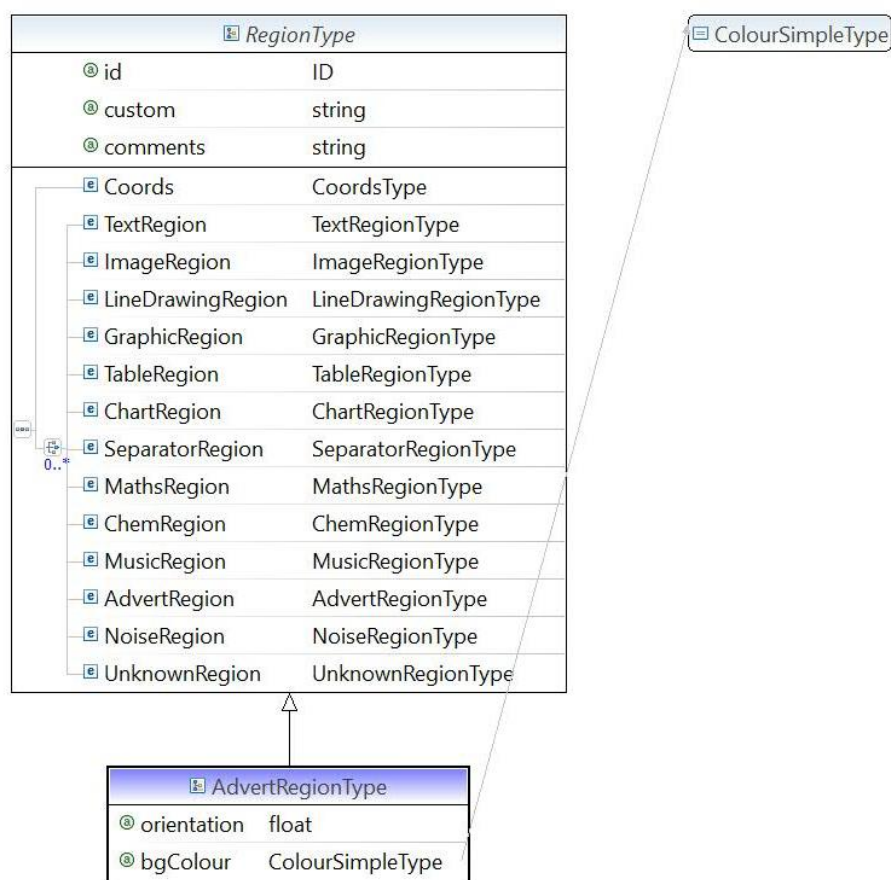


Figure 36: AdvertRegionType (ChemRegionType and MusicRegionType accordingly)

These additional region types do not necessitate any changes when converting existing documents but require tool support if they are to be utilised.

5.3.4 TextStyleType

Previous versions of the pagecontent format allowed storing of text style information only on region level (as attributes of TextRegion). This is now extended to TextLine, Word and Glyph via a dedicated TextStyle element. Besides the finer granularity this also means a better structured schema where text style attributes are kept separate from other information. Any existing text style related attributes were removed from TextRegionType and moved into TextStyleType (Figure 37). Also, several new attributes (such as subscript, superscript etc.) were added in response to feature requests that were collected over time.

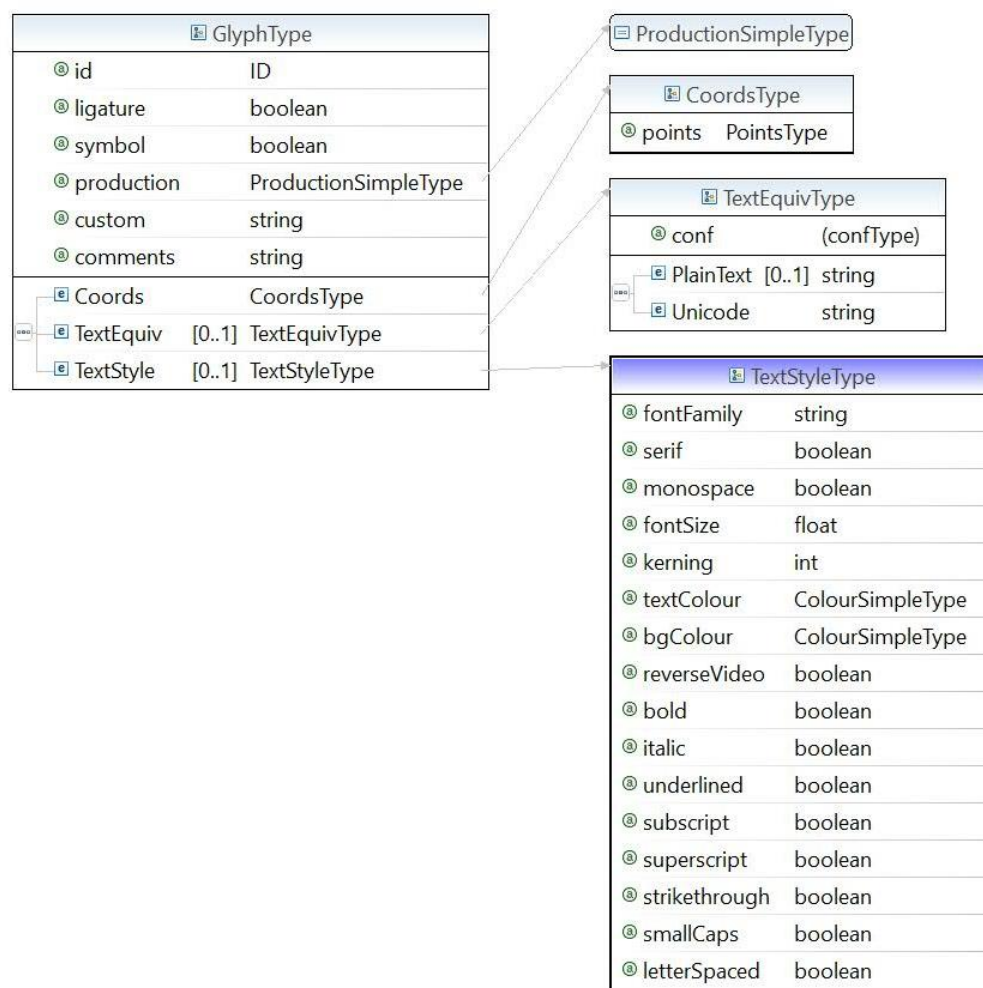


Figure 37: TextStyleType

Text style attributes are considered to be inherited by child elements. Values that are specifically set on child level will override inherited ones.

Overall, the introduction of a dedicated TextStyle element is a major change which requires existing documents to be converted and tools to be updated.

5.3.5 Custom and Comments Attributes

Despite all efforts to support as many application scenarios as possible it is very likely that there will be situations where the existing features are not sufficient. In anticipation of such cases, this release introduces the two new attributes custom and comments (Figure 38) which may be used to record additional information which is not strictly prescribed by the pagecontent format.

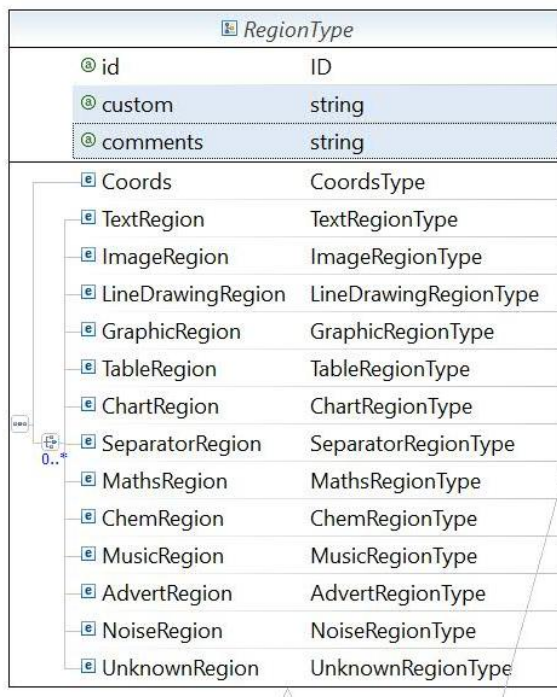


Figure 38: Custom and comments attributes in RegionType

The custom attribute is intended for machine use (such as a software tool defining a custom type or adding its own metadata). The comments attribute is aimed at human users for keeping notes and general information enclosed in the document.

Both attributes are optional and hence do not necessitate any changes when converting existing documents. It is recommended that tools support comments through some form of user interface but depending on the application scenario may not need the custom attribute if all required features are covered by the schema.

5.3.6 RelationsType

The addition of RelationsType allows relations between regions to be further modelled. Typical examples of such relations are a dop-capital and its paragraph or an image and its caption. Similar to ReadingOrder and Layers this is implemented through a logical structure that uses region references.

A Page element may now have a Relations child element as defined by RelationsType (Figure 39).

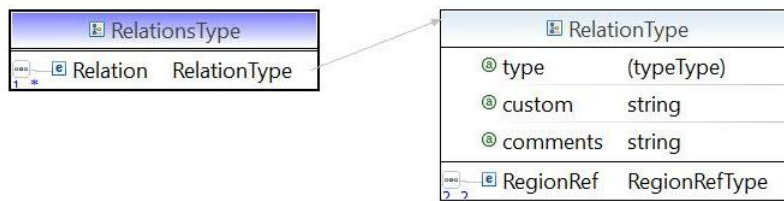


Figure 39: RelationsType

Individual relations are stored as Relation elements containing exactly two RegionRef elements which point to the regions in question. The type of a Relation can either be “link” (related but separate) or “join” (forming a unit). If further information about a relation is to be recorded then this can be achieved via the custom and comments attributes.

With Relations being an optional element, this format change has no direct impact on existing documents but requires tool support to be implemented.

5.3.7 Baseline

This new feature was introduced in response to requests from the handwritten text recognition community. The baseline is an important reference object when it comes to recognising cursive writing which may not always follow a straight line. It represents the (visible or invisible) line that regular letters sit on with descenders going below. In the pagecontent format it is now possible to add a Baseline element as child of TextLine (Figure 40).

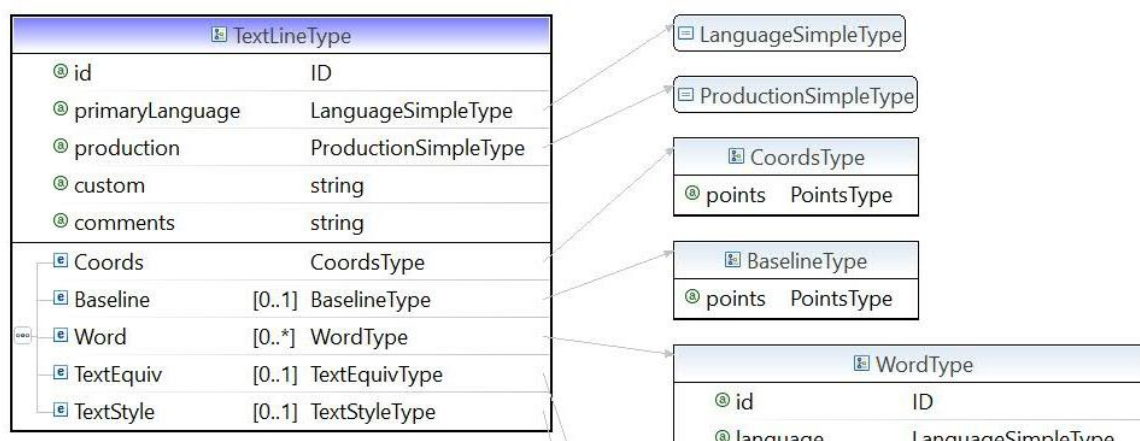


Figure 40: Baseline as defined in TextLineType

The general point format is the same as for region outlines (CoordsType) but the interpretation here (BaselineType) is that of a polyline i.e. the last and first point are not considered connected.

As a new optional element this has no direct impact on existing documents but tool support needs to be implemented.

5.3.8 Further Changes

This major release also saw various smaller changes and additions as listed below.

- The PlainText child element of TextEquiv is now optional. With the Unicode element remaining in place and required, this is to gradually phase out any transcription workarounds as were partly used in the past.
- Text recognition confidence has been introduced as a new “conf” attribute of TextEquiv. This allows OCR software to store values derived from internal metrics (where available) related to recognition results. The value range is floating point numbers from 0 to 1 (corresponding to 0 to 100%).
- Language can now be specified on text line and word level, in addition to region level. To accommodate this TextLine elements have a new attribute “primaryLanguage” and Word elements have a new attribute “language”. Values are inherited but if set at child level override values of parent elements.

- The elements <Layer>, <UnorderedGroup> and <OrderedGroup> may now have a “caption” attribute. This can be used to further describe these elements and allows for better usability in tools for editing reading order and layer structures.
- The Page element has a new optional attribute “type” allowing values of
 - front-cover
 - back-cover
 - title
 - table-of-contents
 - index
 - content
 - blank
 - other
- TextRegion, TextLine, Word and Glyph elements may now have a “production” attribute. Values are inherited but if set at child level override values of parent elements. The following values are allowed:
 - printed
 - typewritten
 - handwritten-cursive
 - handwritten-printschrift
 - medieval-manuscript
 - other
- The type attribute of TextRegion elements now has two more possible values:
 - endnote
 - other
- The type attribute of GraphicRegion elements now has three more possible values:
 - decoration
 - barcode
 - frame
- Colour attributes used in the schema now have an additional allowable value:
 - other

- Colour depth attributes used in the schema now have an additional allowable value:
 - other
- The Page element has a new optional child element `AlternativeImage` with attributes “filename” and “comments”. This can be used to link a secondary reference image (such as a black and white image in addition to a full colour image).
- The schema constraints for “primaryLanguage” and “secondaryLanguage” attributes were updated to now be in line with the language names as per ISO 639-1 [59].

5.4 Fifth Release

Previous updates had covered a significant number of improvements and additions leading to a maturity level that allowed the fourth release to stay in use for a considerable time span. During this time development of tools at the PRImA Research Group as well as collection of feedback and input from the wider community continued, leading to the fifth pagecontent format specification which was released in 2016 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2016-07-15>. The majority of changes are related to additional attributes and allowable values in order to cater for a wider range of application scenarios as well as greater level of detail. The only structural change in this release aims at greater flexibility for the representation of text in corresponding elements.

5.4.1 OCR/Text Variants

This feature was introduced to support OCR engines that are capable of producing text output variants and/or general use cases where digitised documents are of such quality that manual text entry may require the ability to record different possibilities (ideally in combination with a confidence value or explicit ranking).

From a schema point of view this is now implemented by allowing the `TextEquiv` element (as child of `TextRegion`, `TextLine`, `Word` or `Glyph`) to occur multiple times with each sibling representing a variant (Figure 41).

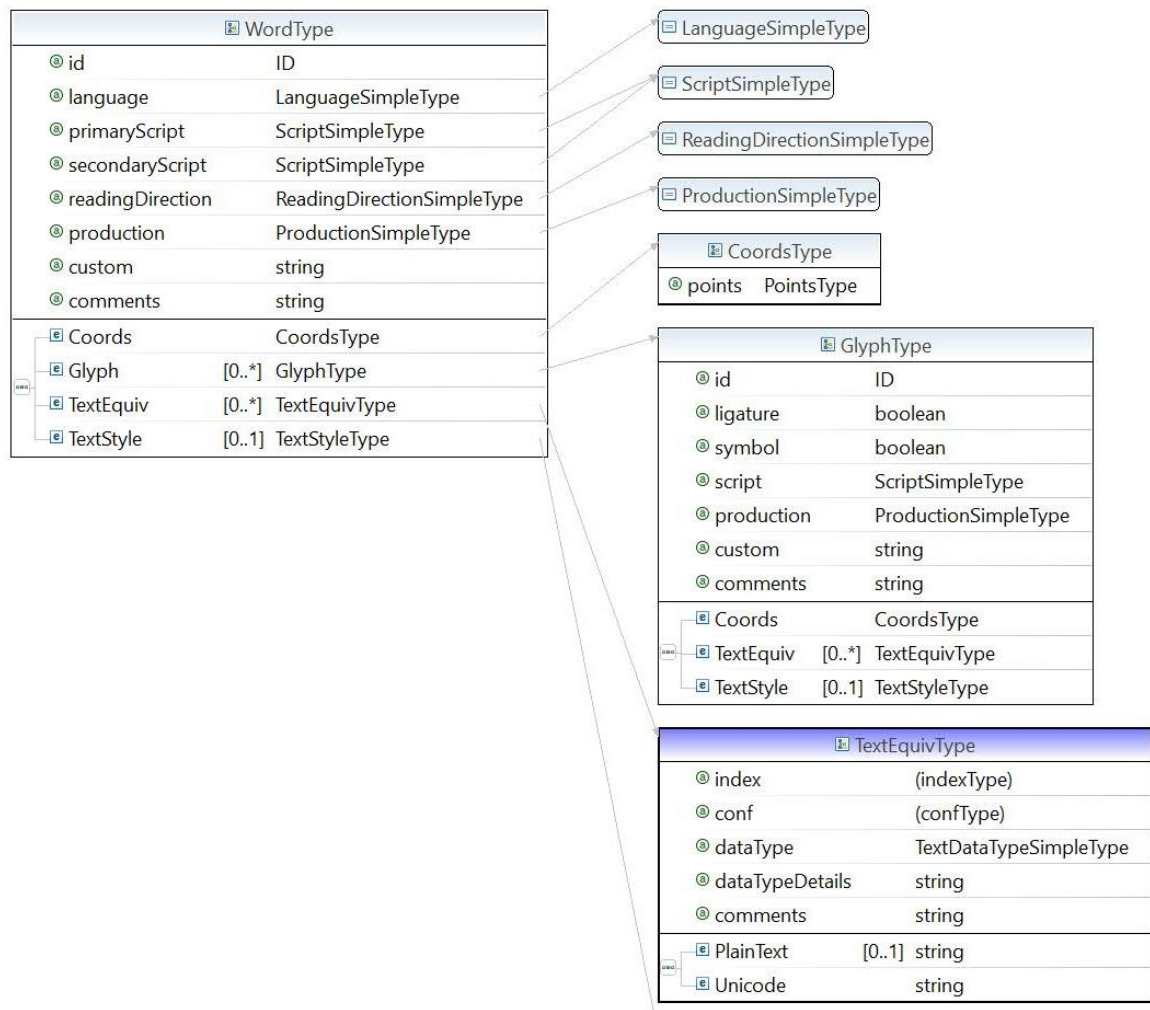


Figure 41: TextEquivType (as used on word level)

To express an explicit ranking, TextEquivType has now a new attribute “index”. The lowest index value indicates the default element i.e. the one to be use as main text content if variants are present. This may generally correspond with confidence values (if available) but this is not a requirement and index as an explicit ranking will always take precedence over confidence. This allows to deal with cases where an OCR engine may inappropriately return a high confidence for a wrong variant and (manual) intervention is required to select the best variant while keeping an unaltered record of all original confidence values.

The use of TextEquiv as specified in the previous version is still covered by the new release and therefore does not require special attention when converting documents to the new format. Tools will require a functional update to support the new feature of variants.

5.4.2 Attributes

As indicated before, the majority of changes in this release are related to attributes and their possible values.

- There is now a new optional attribute “textLineOrder” for Page and TextRegion elements. It represents the order of text lines within parent elements. This is in addition to the already existing “readingDirection” which describes the order below text line level (i.e. words and characters). Although common, textLineOrder and readingDirection do not have to be the same in one document. Possible values are:
 - top-to-bottom
 - bottom-to-top
 - left-to-right
 - right-to-left
- It is now possible to define page level defaults for “primaryLanguage”, “secondaryLanguage”, “primaryScript”, “secondaryScript”, “readingDirection” and “textLineOrder” via corresponding attributes of the Page element. Before, this information could only be recorded on region level. That meant even for documents where all regions share the exact same attribute values this had to be recorded for every region separately. Accordingly, this change will allow for more compact document instances by reducing the amount of redundant information.
- “primaryScript” and “secondaryScript” are now available as attributes for TextLine and Word elements. This is to handle cases where an individual text line or word differs from other siblings inside the same parent element in terms of the used script.
- On glyph level this is implemented through a single “script” attribute as there is no ambiguity to be expected. The “script” attribute shares the same value constraints as “primaryScript” and “secondaryScript” as defined in ScriptSimpleType.
- Regarding the constraints of ScriptSimpleType, these have been updated to comply with the latest version of ISO 15924 [60]. This requires value conversion from “Chinese-Traditional” to “Han (Traditional variant)” and “Chinese-Simplified” to “Han (Simplified variant)”.

- A further addition is the new optional attribute “datatype” for TextEquiv elements. It allows to describe the nature of stored content more precisely where appropriate. For instance, a sequence of Unicode characters could actually represent a number or a date which could be converted into a corresponding datatype in a computer program. Possible values follow the standard XML schema definition data types plus “other” as fallback solution:
 - xsd:decimal
 - xsd:float
 - xsd:integer
 - xsd:Boolean
 - xsd:date
 - xsd:time
 - xsd:dateTime
 - xsd:string
 - other
- As an extension to the point above, there is a new optional attribute “dataTypeInfo” for TextEquiv elements which allows further explanation and/or refinement of “datatype”. This can for instance be a precise regular expression describing allowed patterns or some other form of constraints and production rules such as an enumeration of possible values.
- TextStyle elements can now have an optional attribute “xHeight”. It represents the distance from the baseline to the mean line (top) of lower-case letters in a specific typeface. Values have to be of type integer with an implied unit of pixels.
- Finally, there is a new optional attribute “externalRef” for the Metadata element. It can store a generic string to express an external reference of any kind. This can for instance be useful if bi-directional links between pagecontent instances and a corresponding super-structure are to be maintained.

5.5 Sixth Release

The sixth pagecontent format specification was released in 2017 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2017-07-15>. Major new features are graphemes as further level in the element hierarchy, extended support for

user defined attributes and the concept of roles related to regions that are nested inside a parent region. As in previous versions, there are updates to attributes and their allowed values with the goal of covering more use scenarios and representing content more accurately.

5.5.1 GraphemesType

Graphemes are introduced as yet another level in the hierarchy of text related elements and are located below glyphs. As such, graphemes can represent sub-components of glyphs. They can be defined as the smallest graphical units that can be assigned Unicode code points. A use case for graphemes is presented in [61].

In the schema this is implemented via an optional Graphemes element as child of Glyph. The Graphemes element functions as a container for one or more individual Grapheme or NonPrintingChar or GraphemeGroup elements as defined in GraphemesType (Figure 42).

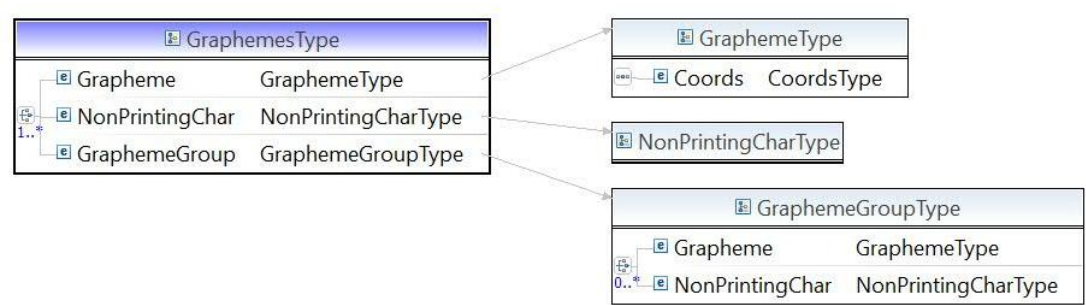


Figure 42: GraphemesType

All three possible child elements extend **GraphemeBaseType** (Figure 43) with common attributes and the general ability to store text via **TextEquiv**.

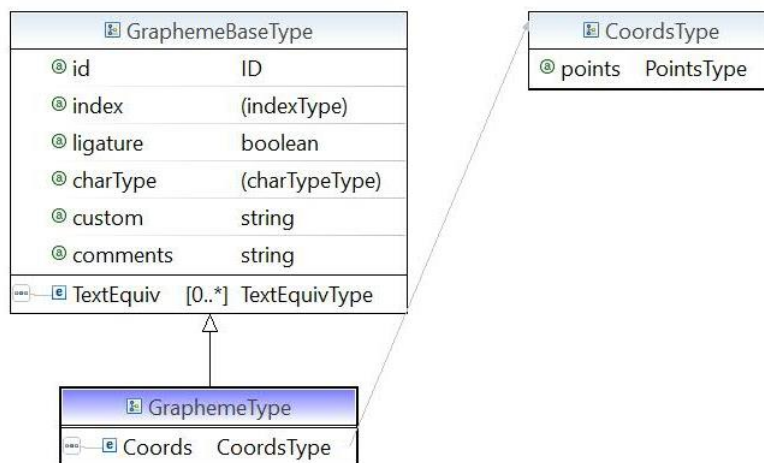


Figure 43: GraphemeType

An individual grapheme is also represented by a region outline as stored in a **Coords** element.

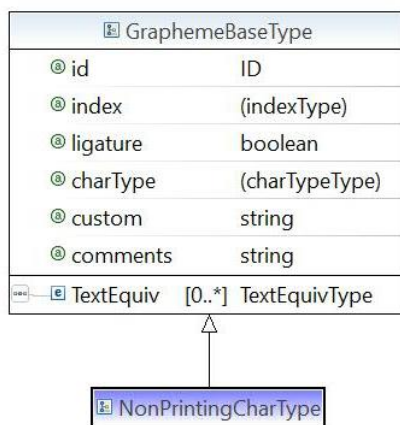


Figure 44: NonPrintingCharType

Non-printing characters as defined by **NonPrintingCharType** (Figure 44) relate to glyph components without visual representation (hence no region outline coordinates) but with a Unicode code point (such as modifiers/control characters).

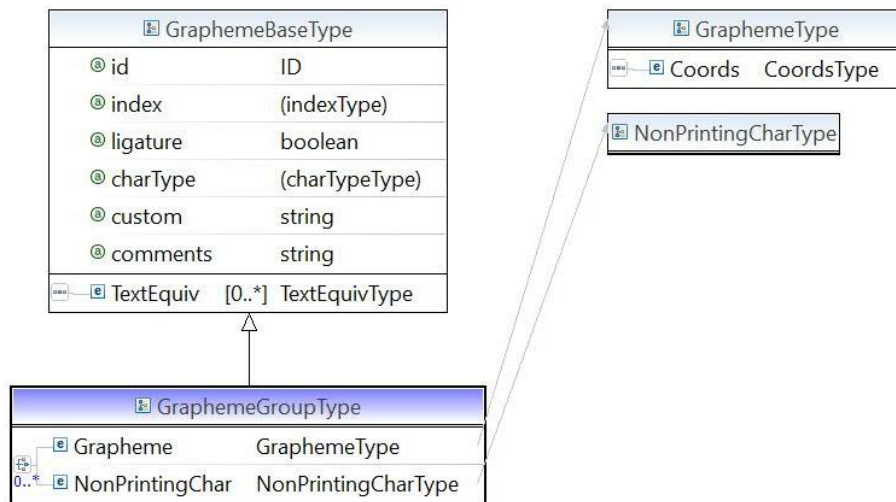


Figure 45: GraphemeGroupType

Grapheme and NonPrintingChar elements can also be grouped via a GraphemeGroup element, defined by GraphemeGroupType (Figure 45), as direct child of the Graphemes container (no further nesting possible though).

The addition of graphemes to the specification does not have an immediate impact on existing documents but will require specific tool support.

5.5.2 UserDefinedType

Previous versions have already seen the introduction of basic attributes that can be used to store application specific (custom) and/or additional user provided information (comments). This concept is now extended to allow for even more customisation that may be required in specific use scenarios.

There is now a new optional UserDefined element that may be used as child of Page, Metadata, all top level regions, TextLine, Word, Glyph and reading order groups. The UserDefined element may have one or more UserAttribute child elements as given in UserDefinedType (Figure 46).

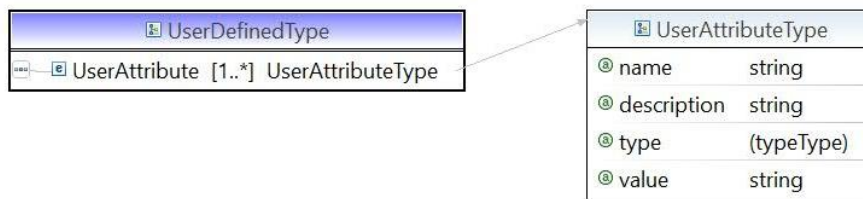


Figure 46: UserDefinedType

The attributes “name”, “description”, “type” and “value” can then be used for storing application specific information in a more structured way (potentially allowing values to be properly parsed if type is used) than was possible before with simple “custom” or “comments” strings.

The fact that all user defined elements are optional means that existing documents are not directly affected by this change. As with any new feature, tool support will need to be implemented.

5.5.3 RolesType

Roles are intended as a general mechanism for expressing the function of nested elements inside their parent region. At this point, roles are only used in the context of tables with the goal of more precisely expressing their inner structure. The abstract type RegionType has now a new optional element Roles as defined in RolesType (Figure 47).

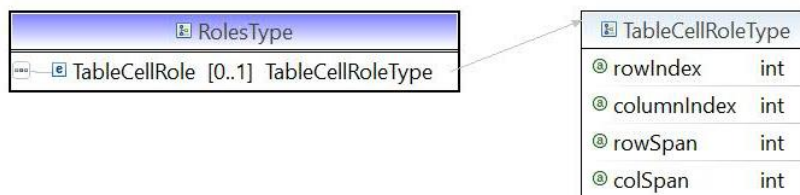


Figure 47: RolesType

TableCellRole is currently the only possible child element for the Roles element of a region. It allows to declare this region a cell inside its parent region, as specified via its “rowIndex”, “columnIndex”, “rowSpan” and “colSpan” attributes. Through this general approach it is possible to assign any type of region (text, image, graphic) as cell content.

The addition of roles to the specification does not have an immediate impact on existing documents but will require specific tool support.

5.5.4 Further Changes

As in previous releases, a number of attributes and possible values were added to cover more use cases.

- Reading order groups can now have an optional “type” attribute with one of the following values:
 - div
 - paragraph
 - list
 - list-item
 - figure
 - article
 - other
- In addition, reading order groups can now also have the optional attributes “custom”, “comments” and “regionRef”. The intention behind region references at group level is to establish an explicit link where reading order structure and region nesting are directly related. As such, “regionRef” of a group element may link to a parent region of nested regions. Consequently, only nested child regions should then be allowed as references inside the reading order group.
- Region elements and reading order groups may now have an optional attribute “continuation”. This can be used to indicate on a general level that the content of the current element is a continuation of a previous element (can be internal such as across columns or external such as across pages and thus pagecontent instances).
- The “type” attribute of TextRegion elements can now have “list-label” as a further possible value. This covers the very specific case of typically small regions representing list item labels (e.g. numbers or bullet characters).
- It is now possible to provide RGB codes beside colour names in TextStyle elements using the optional attributes “textColourRgb” and “bgColourRgb”.

5.6 Seventh Release

The seventh pagecontent format specification was released in 2018 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2018-07-15>. It introduces a number of advanced features as well as general improvements for more precise content representation.

5.6.1 Additional Region Types

This release adds support for two more region types. The first is intended for capturing regions that represent maps. This is implemented via a MapRegion element as defined in MapRegionType (Figure 48).

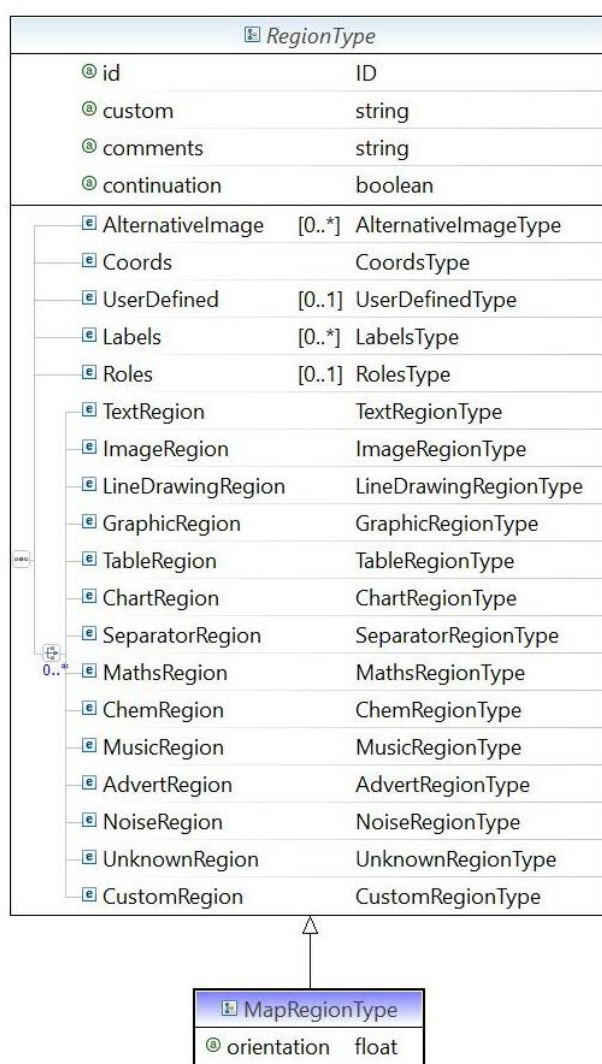


Figure 48: MapRegionType

At this point it is not envisaged to provide a deeper structure related to the inner content of map elements, only to record the existence and location of such content objects. The majority of properties are inherited from RegionType with only “orientation” as an additional attribute.

The second new region type is intended as another way to make the format more customisable. It is now possible to use CustomRegion elements as defined in CustomRegionType (Figure 49).

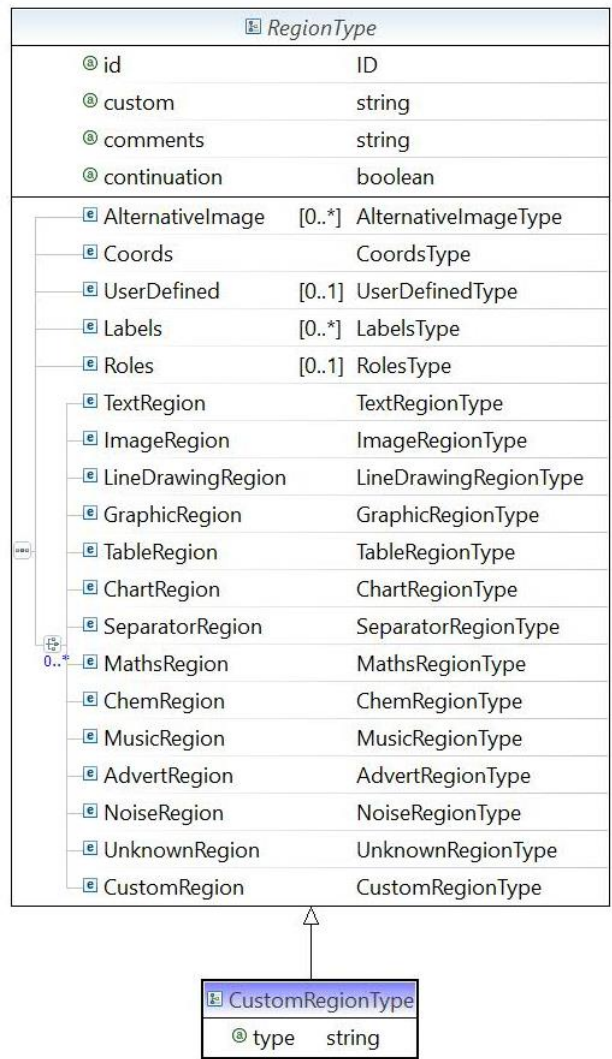


Figure 49: CustomRegionType

There is one specific “type” attribute which can be used to describe the nature of the custom region. Before, it was only possible to use UnknownRegion as a work-around but with CustomRegion (in combination with user defined attributes) there is now a better

and more explicit mechanism in place for cases where third party developers wish to implement their own application scenarios which may not be fully covered by the existing range of elements and attributes.

The two additional region types do not have an immediate impact on existing documents but specific tool support will need to be implemented.

5.6.2 LabelsType

Labels are an addition to the pagecontent schema that allows referencing of resources describing the semantics of document data and software methods. Labels can be attached to the majority of content elements (such as Page, all regions, reading order groups, Relation and MetadataItem).

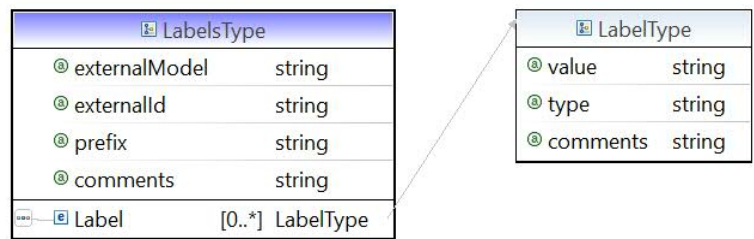


Figure 50: LabelsType

A Labels element can be used to group labels from the same model. The attributes support referencing of an external ontology or schema in “externalModel”, an identifier (such as used in RDF for subject/object triples) in externalId, a “prefix” for all labels and a generic comment. Individual Label elements are then stored as direct children of Labels and have attributes for “value”, “type” and “comments” (Figure 50).

Semantic labels allow the meaning of data elements and related methods to be further formalised and explained. This can for instance be utilised in workflow systems to determine compatibility and general configuration options (such as region type A can be processed/recognised using method X). An application of the semantic labelling feature is presented in [62].

Semantic labels are an addition that does not directly affect existing documents but tool support needs to be implemented.

5.6.3 MetadataItemType

Pagecontent files can now have additional information stored in optional MetadataItem elements as direct children of Metadata. MetadataItemType (Figure 51) defines four attributes for these elements.

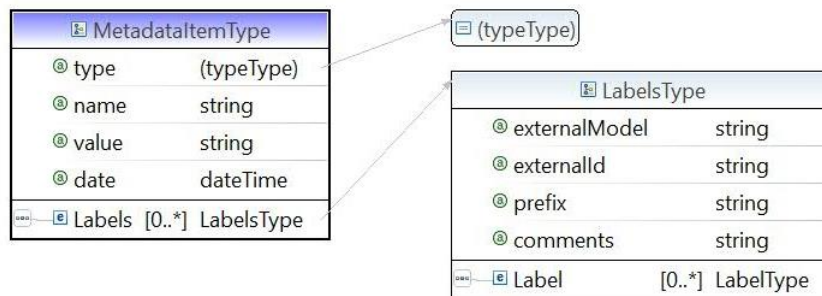


Figure 51: MetadataItemType

The optional “type” attribute can currently have one of the following values:

- author
- imageProperties
- processingStep
- other

The attributes “name” and “value” can then be used to store the actual information (with “value” being the only required attribute). If relevant, it is also possible to store a time stamp in “date”.

Metadata items are an addition that does not directly affect existing documents but will require specific tool support.

5.6.4 GridType

Table regions have already seen updates in previous versions that were targeted at their inner structure and actual cell content. This release adds the ability to store an explicit table grid which is a feature that can be utilised in template-based table recognition (as for instance described in [63]). There is now an optional Grid element as child of TableRegion which is further defined in GridType (Figure 52).

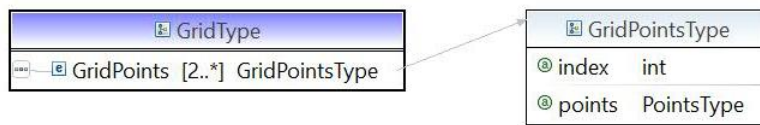


Figure 52: GridType

GridPoints elements are used to describe individual rows in the table. The first row is defined by two GridPoints elements and each subsequent row through one additional GridPoints element. GridPointsType uses the standard PointsType and an additional “index” attribute to record the explicit order (rather than just relying on the order of elements in the XML document which is not necessarily guaranteed).

The new table grid feature is an addition that does not directly affect existing documents but will require specific tool support.

5.6.5 RelationType Update

Relations were introduced in the fourth release and allowed two arbitrary RegionRef elements to be listed in a specific Relation element. The structure of such relations is now clearer in that the two references are explicitly named and directly accessible.

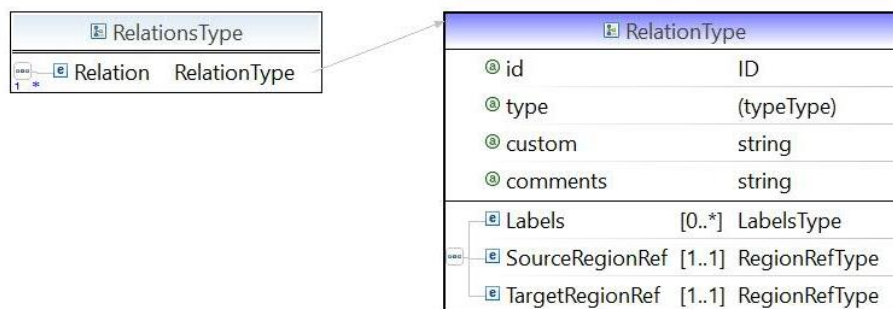


Figure 53: RelationType

The updated RelationType (Figure 53) defines now the two child elements “SourceRegionRef” and “TargetRegionRef” for storing region references. There is now also an additional “id” attribute required so that individual Relation elements can be addressed without ambiguity.

This is a structural change which requires existing documents to be actively converted. Tools need to be updated as well to use the new element names and to ensure valid identifiers are generated.

5.6.6 Further Changes

The following list outlines smaller additions to the format specification that were introduced in the seventh release.

- "AlternativeImage" is now available as optional child element for all regions, TextLine, Word, and Glyph. It allows linking of an external image file specifically for the concerned parent element using its "filename" attribute. This can be useful where individual parts of the overall document image may require separate pre-processing and/or enhancement.
- The ability to store confidence values (related to internal success measures of analysis/recognition methods) has been extended. Initially only introduced for text content, the "conf" attribute is now also available for Page, Coords, BaseLine, ReadingOrder and AlternativeImage. No assumptions are made on how a method would calculate its own confidence values apart from requiring them to be between 0 (lowest) and 1 (highest).
- It is now possible to indicate header cells in tables. This is achieved via a new Boolean "header" attribute in TableCellRoleType.
- TextLine elements can now have an optional attribute "index" to express an explicit order within their parent text region. This is in addition to the already existing textLineOrder attribute on Page or TextRegion level which only indicates a general direction.
- The page element has three new optional attributes to record the resolution of the reference image: "imageXResolution", "imageYResolution" and "imageResolutionUnit" with possible values of "PPI" (pixels per inch), "PPCM" (pixels per centimetre) and "other".

5.7 Eighth Release

The eighth version of the pagecontent format specification was released in 2019 under the namespace of <http://schema.primaresearch.org/PAGE/gts/pagecontent/2019-07-15>.

It constitutes so far the last release and addresses only three points.

- Similar to page level defaults for language and script attributes as introduced in the fifth release, it is now possible to define text style information at the highest level. This is achieved by allowing a TextStyle element directly underneath the Page element. The interpretation follows the common approach that elements on lower levels inherit attributes from above unless overridden.
- TextStyle elements may now have an optional attribute “underlineStyle” with the following values:
 - singleLine
 - doubleLine
 - other
- The Page element may now have an “orientation” attribute of type float to indicate a skew angle. It represents the angle by which the whole page has to be rotated in clockwise direction in order to obtain a correctly orientated result. This can be particularly relevant if coordinates inside a pagecontent instance are in relation to a skewed image.

Existing documents are not directly affected by these additions but tool support needs to be implemented.

The pagecontent format specification has reached a very mature state with its eighth release and there are currently no major feature updates planned. This may of course change in the future depending on advances and general developments in the field.

5.8 Summary

This chapter has covered the rather extensive version history and evolution of the pagecontent format. Additions and changes were ranging from minor improvements and corrections of problems to entirely new features, region types and abilities (e.g. nesting of all region types, recording of variants). More semantic elements were introduced, such as relations between regions and semantic label that allow the meaning of data elements to

be further described. With graphemes, an even finer structural level below glyphs was added. Certain extensions, such as the baseline element, opened the format up to new application areas like handwritten documents. Moreover, mechanisms for custom elements were added in order to accommodate very specific application scenarios.

The achieved level of maturity of the format specification indicates that the first objective of this work has been successfully achieved. This is further considered in Chapter 7 Discussion.

6 Encoding Unstandardised Characters

The previous chapters have addressed the problem of encoding digitised documents on a higher level i.e. with regard to content regions on a page, their relations and attributes. For storing actual text information, it was pointed to the relevant standards such as ASCII for simple representations and Unicode as more comprehensive solution. An open problem that remains is the handling of characters that are not part of a standard. Despite Unicode covering a vast number of scripts, languages and special characters, there are still cases where no corresponding code point can be found. Scenarios where this issue is often encountered are historical documents containing unique and/or not commonly used characters and symbols as well as documents in languages and scripts that are endangered or not in active use anymore.

The following presents a novel approach to deal with this problem by ascertaining a document specific alphabet which will reveal the existence of any unstandardised characters. In particular, it will be shown how the central issue of finding the set of unique characters can be achieved in a fully automated fashion through adaptive glyph clustering.

6.1 Adaptive Glyph Clustering

The problem of identifying unstandardised characters can be translated into filtering all instances of characters in a document into the set of unique ones i.e. the document specific alphabet and then to match this set against the existing standard (Unicode) to expose missing entries. A set of unique elements in a population can be computed using an appropriate clustering algorithm. The representations of elements to work with in this context are glyph images as found in the input document image.

The main limiting factor to implement a system enabling automatic processing of documents that contain previously unseen/unstandardised characters and symbols is the lack of an *adaptive* clustering method. Clustering algorithms typically require the setting of certain parameters in order to work. Depending on the actual algorithm this can be a threshold value applied to the similarity measure calculated for pairs of input elements. Ascertaining an appropriate threshold with regard to specific clustering goals is non-trivial

and is commonly done by a human operator and depends on the input material. This problem will be studied in the following, leading to a novel solution.

In the context of this work, adaptive glyph clustering is defined as a clustering approach that can automatically determine and apply an appropriate threshold for differentiating glyphs belonging to different clusters - directly in response to a given input (i.e. without user intervention). It will thus have the ability to adapt to different inputs by automatically selecting the best possible individual setting for its primary parameter in order to achieve an optimised result.

Clustering goals will arise from the application requirements and may for instance be to achieve a very small number of clusters or a very high accuracy within each cluster. Depending on the combination and nature of input content (i.e. glyphs), feature selection and representation, similarity measure, and clustering algorithm it may not always be possible to achieve all goals at the same time and/or to the extent desirable. The application scenario considered in this work depends on clusters that are as pure as possible. This means that the objective is to determine a safe threshold i.e. one that leads to clusters that are free from misclassifications while trying to obtain a number of clusters as close as possible to the actual number of classes in the input data as a subordinate objective. If such a safe threshold can be computed automatically from the input data then this will enable an *adaptive* method – one that can automatically set the appropriate threshold and therefore does not require human intervention – and hence fill the gap that was identified before. The research describing a path towards such an adaptive method for glyph clustering will be discussed in the following sections.

6.2 Formalising the Problem

In order to arrive at a more precise description of the problem the requirements and constraints need to be revisited. These can then be related to algorithmic choices to guide the process of devising a novel method.

The overall goal is to implement a clustering method that groups instances of glyph images into their respective classes in a *fully automated* fashion. To allow fully automated processing there needs to be a mechanism that calculates appropriate parameters for the

underlying clustering algorithm rather than expecting it as user input. The question of what constitutes appropriate parameters will be formalised further in Section 6.6.

Considering the nature of the input data two constraints related to the choice of clustering algorithm can be derived:

1. The number of true classes is unknown. This follows directly from the goal to develop a system that can process documents containing previously unseen characters and symbols. If they are indeed discovered for the first time, then there is no effective way to know in advance how many there might be.
2. The number of individual instances to be processed can potentially be very high. Glyph images will typically originate from sources that contain substantial amounts of text, such as books, manuscripts, and newspapers. Assuming that a book page contains 350 words and that there are on average 5 characters (glyphs) per word then this would result in 1,750 characters per page. For a book of 500 pages this would lead to a total of 875,000 characters. If a system should now be able to batch-process whole collections of books it becomes apparent that computational complexity will play a role.

The choice and adaptation of a clustering algorithm will be further discussed in section 6.5

Related to the clustering algorithm it is important to specify how the input data (bitmap images of glyphs) will be represented and to what extent abstraction needs to take place. More detail on this will be given in section 6.3. Also, there needs to be a way to calculate the similarity of two glyph instances in the chosen representation format for the clustering algorithm to work with. An adequate similarity measure will be discussed in section 6.4.

6.3 Glyph Representation

There are two main approaches related to representing images for classification and clustering algorithms. The first approach is to work directly with bitmaps i.e. pixel matrices as they are produced by digital imaging devices. The second approach is to derive high-level descriptions of the image content in a pre-processing step and then to work with vector representations containing specific sets of “feature = value” pairs. A deeper

discussion and formal descriptions of the two representation forms will be given in the following.

6.3.1 Bitmap Images

The starting point for the considered problem is the input data In in the form of a set of glyph images G .

$$In = \{ G(i) \mid i = 1, \dots, n \}$$

The cardinality of In i.e. the number of individual glyph images obtained from a preceding segmentation step is n .

Each individual glyph image $G(i)$ is now itself a set of pixels P .

$$G(i) = \{ P(x, y) \mid x = 0, \dots, xd_i - 1 \text{ and } y = 0, \dots, yd_i - 1 \}$$

The individual image dimensions are given by xd_i for width and yd_i for height. This representation allows for differing image dimensions per glyph which will typically be the case for raw segmentation results. If a normalisation step were to be applied – transforming all glyph images to the same fixed image width and height – then the index i in xd_i and yd_i could be eliminated. For this particular application, however, normalisation is not appropriate as resizing (scaling) images can introduce distortions and lead to information loss which may have a negative effect down the line (such as lower-case letters appearing identical to their upper-case counterparts).

All following calculations and processing steps will be based on either greyscale or – as special case thereof – black-and-white images. Colour information can be discarded at this point as the main focus is on geometrical features and the shape of glyphs. Accordingly, the following ranges can be defined for pixels P_{bw} in black-and-white images and P_g in greyscale images:

$$P_{bw}(x, y) \in \{0, 1\}, (0 = \text{white}, 1 = \text{black})$$

$$P_g(x, y) \in \{0, \dots, 255\}, (0 = \text{black}, 255 = \text{white})$$

This is following the interpretation that in a black-and-white image the interesting content (also commonly referred to as the foreground) is black – in analogy to black ink on a white

sheet of paper. For greyscale images, on the other hand, it is customary to consider how pixels should be stored in bitmap files to represent the required illumination on a display, leading to 0 for black and 255 for white as the highest brightness level.

6.3.2 Feature Vectors

The representation of glyphs as bitmap images has the advantage that it does not require any complex information-related⁶ pre-processing. However, bitmap images do not provide a particularly condensed or precise description of the actual content. With regard to the task of clustering (i.e. finding highly similar instances of glyphs) bitmaps typically contain a high proportion of irrelevant information (e.g. background) while important characteristics that may help distinguish different glyphs (such as number of components or stroke intersections) are not directly accessible. This problem can be addressed by extracting (calculating) meaningful features from the original images and then in subsequent steps to work with surrogate representations in the form of feature vectors. Table 1 illustrates the idea of representing glyphs using specific image features:

Table 1: Example features for distinguishing different character classes

Character class	X	Y	x	z
Number of junction points	1	1	1	0
Number of end points	4	3	4	2
Height in pixels	100	100	80	80

In this example, lower-case x, upper-case X as well as Y display one junction point of intersecting strokes at the centre. This poses a relevant feature as it can be used to differentiate the three from a lower-case z which does not have such a junction point. Further, the capital X and Y can be differentiated from the lower-case x by considering their height. Finally, X and Y can be distinguished by the differing number of stroke end points. The corresponding feature vectors of X (1,4,100), Y (1,3,100), x (1,4,80) and z (0,2,80) would be sufficient to differentiate each element from each other as demonstrated.

⁶ As opposed to image-related pre-processing steps such as contrast enhancement, binarisation, despeckle etc. which may be appropriate depending on the actual image quality.

Some features (such as image height) may be very simple to obtain while others might require considerable image processing potentially leading to results which could be inaccurate. In general, feature extraction requires a mapping φ from glyph images G to feature vectors F .

$$\varphi: G \rightarrow F$$

For a (feature) vector space of dimension m this translates into a set of individual functions (feature extraction methods) for every component F_j :

$$\varphi = \{G \rightarrow F_j \mid j = 1, \dots, m\}$$

and

$$F_j = \varphi_j(G)$$

The example in Table 1, for instance, is based on a 3-dimensional feature space ($m = 3$), requiring three feature extraction methods in order to map every input image $G(i)$ to a corresponding feature vector $F(i)$ composed of the three components $F_1(i)$, $F_2(i)$, and $F_3(i)$. Feature number three, in this case, would be very simple to obtain (effectively only requiring a single value-lookup from the input image metadata or alternatively counting the number of rows in the pixel matrix):

$$F_3(i) = \varphi_3(G(i)) = yd_i$$

The other two features here are much harder to determine and would require a number of image processing and analysis steps (such as edge detection, skeletonisation, and structural analysis of line segments).

Depending on the nature of the considered features further mapping/normalisation can be required in order to control the influence a single feature will have when individual glyph instances are compared. For instance, a feature extraction method φ_a could have a range of (false|true). To be used by a numerical similarity measure this would have to be mapped to real numbers first of all e.g. (0|1). Another feature extraction method φ_b , on the other hand, might produce values ranging from 0 to 100. If the employed similarity measure was now based on a Euclidian distance then this would result in feature F_b

dominating F_a due to potentially greater values of F_b . Normalisation in this context would involve mapping all individual features from a range ($min, \dots, max$) to ($0, \dots, 1$) to ensure equal relative importance.

Another important factor to consider is the expressive/discriminatory power of features for computing the similarity of glyphs in the scope of clustering. There is no upper limit on the number of features (and combinations thereof – cf. width to height ratio) that can be used to represent glyphs but not every single one will be useful to decide on the similarity of two instances. Feature selection is then the process of choosing the right number and types of features (i.e. cardinality of φ and concrete functions/methods φ_j) to solve the problem at hand. The term “right” can be difficult to define in this context and typically leads to an optimisation problem, aiming for the best possible selection of features i.e. achieving the highest accuracy under constraints such as limited time to search for a solution as well as availability of reliable algorithms for extracting meaningful features. With feature selection and optimisation being a major research problem in its own right it is intentionally not pursued further in this work apart from the goal of obtaining “a” viable solution to prove the point that adaptive glyph clustering is possible. Future work could explore this topic in more detail to arrive at an implementation with optimised performance.

For the work presented here, the following image features were considered based on findings reported in prior work such as [64]:

- Width
- Height
- Width to height ratio
- Number of black pixels (foreground)
- Number of white pixels (background)
- Black-to-white ratio
- Sub-resolution pixel values (regular 3×3 partition of glyph images - numbers of black pixels and black pixel densities in each of the nine rectangles)
- Glyph circumference (number of contour pixels)
- Compactness (circumference to area ratio)

- Centre of gravity
- Number of connected components (e.g. special character % -> 3)
- Number of holes (e.g. number 8 -> 2)
- Projection profiles (0° - horizontal, 45° - top left to lower right, 90° - vertical, and 135° - top right to lower left)

As indicated above, the concrete choice of features can have a significant effect on the achievable overall clustering accuracy. However, related to the particular research question studied here it is not of great significance. As set out in 6.2 the goal is to develop a fully automated clustering method with the ability to determine any parameters automatically from the input data, i.e. without user input. In this sense, the feature representation going into the clustering algorithm will be part of the input and hence the mechanism responsible for calculating parameters will have to work with what is available, effectively looking for the best possible solution given this particular input data. A non-perfect selection of features will not limit the ability to develop such an adaptive clustering method, as long as it is meaningful enough to allow for some form of correct clustering effect (types and significance of errors will be further discussed in 6.6).

6.4 Measuring the Similarity of Glyphs

At the core of any clustering algorithm will be a step that decides whether two instances from the input are considered to be of the same class or from different classes in order to assign them to the right cluster. Initially this may involve calculating a degree of similarity (i.e. a value from a continuous range) which is then mapped to a yes/no answer by applying a threshold.

For glyph images which may vary in size and feature vectors that can potentially have any number and range of features it will be beneficial to use an open scale. This, however, will require to inverse the measure from similarity to dissimilarity as there is always going to be a defined point at one end of the range – for two identical glyphs (dissimilarity of 0). The dissimilarity measure can then be open-ended, leading to higher values for image pairs that differ more as a result of a greater number of pixels and/or greater feature ranges.

In analogy to objects in space the term distance is going to be adopted from here. A distance function δ to be used in a clustering algorithm can then be defined as follows:

$$\delta: G \times G \rightarrow \mathbb{R}$$

According to the discussion above it will have to satisfy the following two requirements:

$$1. \delta(G(i), G(i)) = 0$$

and

$$2. \delta(G(a), G(b)) < \delta(G(a), G(c)) \text{ if } b \text{ more similar to } a \text{ than } c \text{ similar to } a$$

First of all, this means that the distance of two identical glyph images has to be zero. Secondly, it means that an order relation is required on δ so as to ensure that glyph image pairs with more pronounced differences lead to higher distance values.

Concrete distance measures fulfilling these criteria are going to be discussed in the following. First, a feature based distance measure will be defined, followed by one relying on template matching. It will then be shown how the two can be worked into a combined distance measure that can be used in a glyph clustering algorithm.

6.4.1 Feature Based

Operating on the feature vector representation of glyphs allows for a very direct way of calculating how similar two instances are. Using the Euclidean distance in an m -dimensional space it is possible to define a feature based distance function δ_f for two feature vectors F_a and F_b :

$$\delta_f(F_a, F_b) = \sqrt{\sum_{j=1}^m (F_{b_j} - F_{a_j})^2}$$

With

$$F_{a_j} = \varphi_j(G(a)) \text{ and } F_{b_j} = \varphi_j(G(b)).$$

For $F_a = F_b$ the term $(F_{b_j} - F_{a_j})^2$ will become 0 resulting in $\delta_f(F_a, F_b) = 0$ satisfying requirement 1. from section 6.4. For $F_a \neq F_b$ the term $(F_{b_j} - F_{a_j})^2$ will produce a positive

number leading to higher values for pairs of more distant points (i.e. with a greater absolute difference) in the feature space as expected according to requirement 2. from section 6.4.

There are alternative approaches such as Cosine Similarity or Dot Product Similarity for comparing vectors in a feature space. These are, however, less appropriate for the application at hand than the chosen Euclidean distance. Cosine Similarity is based on the angle between two vectors, meaning it focuses on the direction rather than the magnitude. For instance, if glyphs are described by a feature vector composed of (number of connected components, number of holes), then two rather different characters of “p” and “b” would result in the two different feature vectors of (1,1) and (2,2). The Cosine Similarity, however, would attribute them perfect identity due to the identical direction of their feature vectors. The Dot Product Similarity, on the other hand, suffers from its intrinsic zero value ambiguity where very different vector pairs such as $\{(1,0), (0,1)\}$ and $\{(1,0), (0,1000)\}$ will each lead to the exact same similarity result. Moreover, the identical pair of $\{(0,0), (0,0)\}$ will yield 0 similarity i.e. perfect dissimilarity.

6.4.2 Template Matching Based

Another strategy to compare two instances of glyph images is based directly on their bitmap representation. An advantage of this approach is that it does not depend on any intermediate steps such as feature extraction and selection. On the downside, it might be more susceptible to noise in images and can be computationally expensive due to the two-dimensional nature of bitmap images if every single pixel has to be visited.

When comparing two images, one can be considered the input or search image and the other the template. The task is now to find the best match (i.e. overlap of pixels) of the template inside the search image. If search image and template are of the same size⁷ then this requires only the comparison of all pixel values $P_s(x, y)$ in the search image and $P_t(x, y)$ in the template (for $x = 0, \dots, xd_t - 1$ and $y = 0, \dots, yd_t - 1$ with $xd_t = xd_s$ being the template/search image width and $yd_t = yd_s$ the template/search image height). With regard to computational complexity this means two loops that depend on

⁷ This should not be forced through size normalisation of images with differing dimensions as it would introduce distortions with potentially negative effects on the matching process.

the input data. If the two images differ in their dimensions, then it is also required to find the best possible location of the template within the search image (leading to the lowest possible distance value). For this to work in an actual algorithm it may be necessary to fill one image with more background pixels and/or to swap the two images with regard to what is used as search image and template so as to ensure that the search image has the greater dimensions allowing the template to be compared in all possible positions. This adds two more loops to try all possible locations of the template within the search image with regard to the x and y dimension. As discussed before, for images of almost identical size this will only have a minor impact but for images that differ significantly this can lead to substantially longer processing times.

A template matching based distance measure δ_t can now be defined as follows:

$$\delta_t = \min_{\substack{0 \leq l \leq yd_s - yd_t \\ 0 \leq k \leq xd_s - xd_t}} \left(\sum_{d=0}^{yd_t-1} \sum_{c=0}^{xd_t-1} |P_s(c+k, d+l) - P_t(c, d)| \right)$$

In the case of identical images, their width and height will be equal (meaning that no loops are required). Therefore, k and l will only take on the value of 0 and the term $|P_s(c+k, d+l) - P_t(c, d)|$ becomes $|P_s(c, d) - P_t(c, d)|$. Now with $P_s = P_t$ this will always produce 0 and therefore the sums will be 0. This in turn will also be the minimum over all possible positions - satisfying requirement 1. from section 6.4. Applied to differing images this term will produce values equal to or greater than 1 for every deviating pixel⁸ leading to greater sums for image pairs with more differences which follows requirement 2. from section 6.4.

Summarising, the described template matching based distance measure represents an error function with regard to the number of mismatching pixels in black-and-white images or the overall illumination difference in the case of greyscale images for the best possible fit of template and search image. Throughout this work, greyscale images have been used.

6.4.3 A Combined Distance Measure

With the fundamental clustering task being the comparison of glyph images i.e. pixel patterns it can be concluded that the template matching based distance measure will

⁸ 1 for differing pixels in black-and-white images, 1..255 for differing pixels in greyscale images.

produce reliable results due to its guaranteed ability to discern identical and different patterns. Its usefulness, however, declines in the presence of noise (i.e. pixels not actually belonging to a glyph) as it will produce greater distances than the true glyph shapes might warrant. Greater distances can lead to more fragmentation i.e. a greater number of clusters than actual classes but fortunately, will not result in misclassification (in the sense of glyphs of different true classes ending up in the same cluster).

The feature based distance measure, on the other hand, involves various forms of abstraction (related to the choice and calculation of features) and is therefore not guaranteed to be able to sufficiently differentiate all possible glyph patterns. Nonetheless, it has the advantage that it is less susceptible to noise due to the aforementioned abstraction.

To exploit the strengths of both approaches it is proposed to work with a combined distance measure δ_c which is strongly biased towards template matching given it is the safer one. The feature based measure can still contribute in order to add some extent of resilience to noise but should be clearly subordinate.

$$\delta_c = w_t \delta_t + w_f \delta_f s$$

To this end, a-priori weightings of w_t of 80% (template matching) and w_f of 20% (feature based) were chosen. This was the initial configuration (following the above reasoning) used in the experiments for exploring possible solutions. The problem of finding an optimal weight configuration is revisited in Subsection 6.8.3 where better values will be determined.

As the value of δ_t depends on the actual number of pixels involved whereas δ_f comes from vectors with normalised features (0..1), it is necessary to introduce a further scaling factor s which can be used to bring the feature based distance in the same region as the matching error for the relevant range of thresholds (below lossy compression – see 6.6.1). Generally, a suitable scaling factor can be chosen depending on the resolution in ppi (pixels per inch) of the input images. In the following experiments a scaling factor of 500 is used for images of 300ppi.

6.5 Clustering Algorithm

For the design of a suitable clustering algorithm it is necessary to consider the constraints formulated in 6.2. Constraint number 1 states that the true number of clusters is unknown. Consequently, the algorithm will have to work in an unsupervised fashion i.e. it cannot expect the number of cluster to be provided by an operator as for instance k-means would. Constraint 2 points out that the number of instances to process can potentially be very high. This will favour an iterative approach which would help keep the requirements on memory and processing power under control (as opposed to recursive approaches for instance).

A method that can be used for clustering and meets both requirements can be found in the domain of signal processing: Adaptive Vector Quantisation. Vector Quantisation was originally designed to compress information sources such as images or audio streams by mapping continuous signal values to a discrete range. With some modifications it can be made adaptive to the input and thereby suit the application scenario considered here. As such it is an iterative method that returns the number of clusters as a result of the process. The description below follows [65] and [64].

Adaptive Vector Quantisation is based on Vector Quantisation which in turn is based on Scalar Quantisation. The idea of scalar quantization is to restrict the range of a variable to discrete rather than continuous values. Vector quantisation uses then a codebook for allowed values and replaces original data with references to entries in the codebook. Adaptive vector quantisation extends this concept by allowing the codebook to be calculated dynamically depending on the input data. In the context of this work the codebook will contain prototypes of glyphs (i.e. a single glyph to represent each class). With regard to the overall application this has the useful side-effect that the set of prototypes can be considered the underlying alphabet of the input document and therefore be used for encoding as well as preservation of the graphical glyph appearance (i.e. font) of the original.

In the mathematical sense, clustering means to determine a specific partition of the set of elements which are given in the input data. A clustering function χ can then be defined as:

$$\chi: In \rightarrow Cl$$

with Cl as a cluster configuration $\{C_1, \dots, C_r\}$ (i.e. set of r individual clusters C each containing glyph images G as elements) derived from the input In for which all of the following criteria hold:

$$(1) \quad \emptyset \notin Cl$$

$$(2) \quad \bigcup_{C \in Cl} C = In$$

$$(3) \quad \forall C_p, C_q \in Cl : C_p \neq C_q \Rightarrow C_p \cap C_q = \emptyset$$

Or in other words, there cannot be empty clusters, the set of individual clusters covers all of the input (i.e. every input glyph is grouped into a cluster), and every element can only be contained in one cluster (clusters are disjoint).

The actual Adaptive Vector Quantisation algorithm (integrated into the Nearest Neighbour clustering approach) is outlined in the flowchart given in Figure 54.

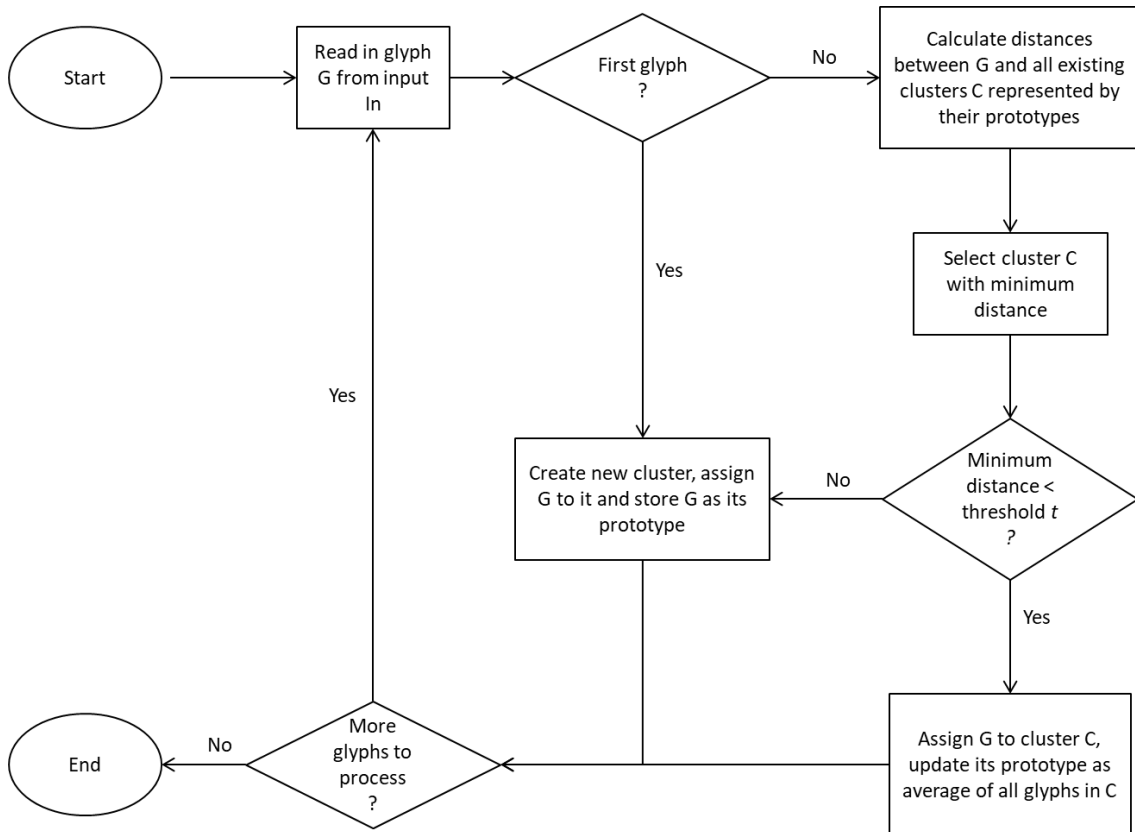


Figure 54: Adaptive Vector Quantisation algorithm for glyph clustering

The first glyph that is read in will always create a new cluster. For following glyphs, it has to be decided if they are similar enough to any of the existing clusters (as represented by their prototype) in order to get included in it or if they are so different that they should constitute the start of a new cluster. If a glyph is added to an existing cluster then its prototype is updated. This involves recalculating the prototype image matrix as well as the feature vector based on the weighted average of old prototype (potentially representing a great number of glyphs already) and the newly added glyph. The process runs until all input glyphs have been assigned to a cluster. The algorithm is adaptive since it reformulates the codebook (here the set of prototypes) as glyphs are added to clusters.

As can be seen from Figure 54, a concrete clustering algorithm will depend on a threshold t to be provided from outside i.e. forms part of the input in actual fact. Accordingly, the Adaptive Vector Quantisation clustering function can be written as follows:

$$\chi_{AVQ}(In, t) = Cl$$

The problem is now that while In is given, it is not known what value should be chosen for t to obtain a valid clustering result. The following sections provide a systematic analysis and propose a novel solution to this problem.

6.6 Determining an Appropriate Threshold

Following on from the discussion so far, the problem of developing a fully automated clustering algorithm can be reformulated as the search for a method that automatically determines an appropriate distance measure threshold to be used by the adaptive vector quantisation implementation discussed in 6.5.

The term “appropriate” which has been used previously is a rather vague description of the requirements on the threshold that such a method should return. Linking back to the bigger picture and potential application scenarios leads to more specific constraints. The goal of this work is to deal with documents containing potentially unknown characters and symbols as manifested in their glyph images. The ultimate goal of digitising historical documents is to obtain an encoded representation that captures the actual meaning of all textual content and allows full text search. For previously unknown characters this will at some point require a stage of manual labelling (e.g. historians and linguists deciding on

the actual meaning of symbols). This process can be made much more efficient by providing means to label a relatively small number of clusters rather than every single glyph instance in the original document. However, in order for this approach to work it depends on all clusters to be free of actual misclassifications (i.e. no a's in a cluster representing b's) as this would otherwise lead to partially mislabelled glyphs (a's wrongly being tagged as b's). If the clustering is too lax, on the other hand, this would only result in a less efficient process as more clusters, potentially containing glyphs belonging to the same class, would have to be labelled (e.g. two separate clusters of b's) but it would not introduce actual errors in the final encoding of the document.

Based on these deliberations it is possible to formalise the requirements on an appropriate threshold:

1. It has to be safe so as to avoid hard misclassifications in the result (i.e. avoid glyphs of different true classes to be assigned to the same cluster). Accordingly, the following condition should hold:

$$\forall G(a), G(b) \in In : G(a) \in C_p \wedge G(b) \in C_p \Rightarrow G(a) \in C_q \wedge G(b) \in C_q$$

With $C_p \in Cl_{res}$ a cluster in the result and $C_q \in Cl_{gt}$ a cluster in the ground truth (i.e. the ideal/correct result).

A clustering result that is free of misclassifications in this sense can be called a valid clustering Cl_{val} .

2. While being safe, it should also aim at producing the smallest possible number of clusters r_{res} in the result (converging to the true number of classes) for a given input.

$$r_{res} = \min_{Cl_{val}} r$$

The appropriate threshold can then be defined as t_a that produces a clustering result satisfying the two conditions above.

The following sections will give a detailed description of the investigations and development towards a method that can calculate such an appropriate threshold automatically, based only on the input data. First, it is established from a dataset with

known class labels (ground truth), how increasing thresholds affect the error rate of the produced results and which threshold leads to the optimal result. Then, a surrogate metric for the error rate (which is unknown in the absence of ground truth) is established so that it can be used for estimating (calculating) an appropriate threshold when new input data is to be processed.

6.6.1 Supervised – From Known Examples

In the following, the dataset German Fraktur (o) with fully labelled data (ground truth) is used to reconstruct what would be the appropriate threshold for this particular input. To this end, salient patterns in the corresponding graphs are explored in order to arrive at a potential solution. This follows a supervised approach whereby the system is given the information about the true classes of all glyphs from outside.

The starting point is the trivial case that satisfies the two threshold requirements from above: Every glyph is contained in its own cluster ($\forall C \in Cl \Rightarrow |C| = 1$). Such a solution will always be free of hard misclassifications. Strictly speaking, this does not require any processing as the input will be equal to the output. It would also be the result from a distance measure threshold of 0 for which the distance test in the clustering algorithm would always fail. Moving on, growing threshold values have to be tested in order to establish if a better solution can be found with regard to requirement 2 (smallest possible number of clusters) while requirement 1 (no hard misclassifications) still holds. Typically, increasing the distance measure threshold will lead to fewer clusters. This effect will be referred to as compression in relation to the trivial solution of one cluster per input glyph. The compression rate *comp* can then be defined as:

$$comp(In, t) = \frac{r_{gt}}{r_{res}}$$

Where r_{gt} is the true number of clusters in the ground truth. Accordingly, the lowest possible compression rate for a particular input is given by $r_{res} = n$ where the number of clusters in the result equals the number of glyphs in the input (without violating requirement 1 from above).

The results shown in Figure 55 were obtained by iteratively running the clustering stage (as per Subsection 6.5) for growing threshold values using the labelled test dataset German Fraktur (o) as described before.

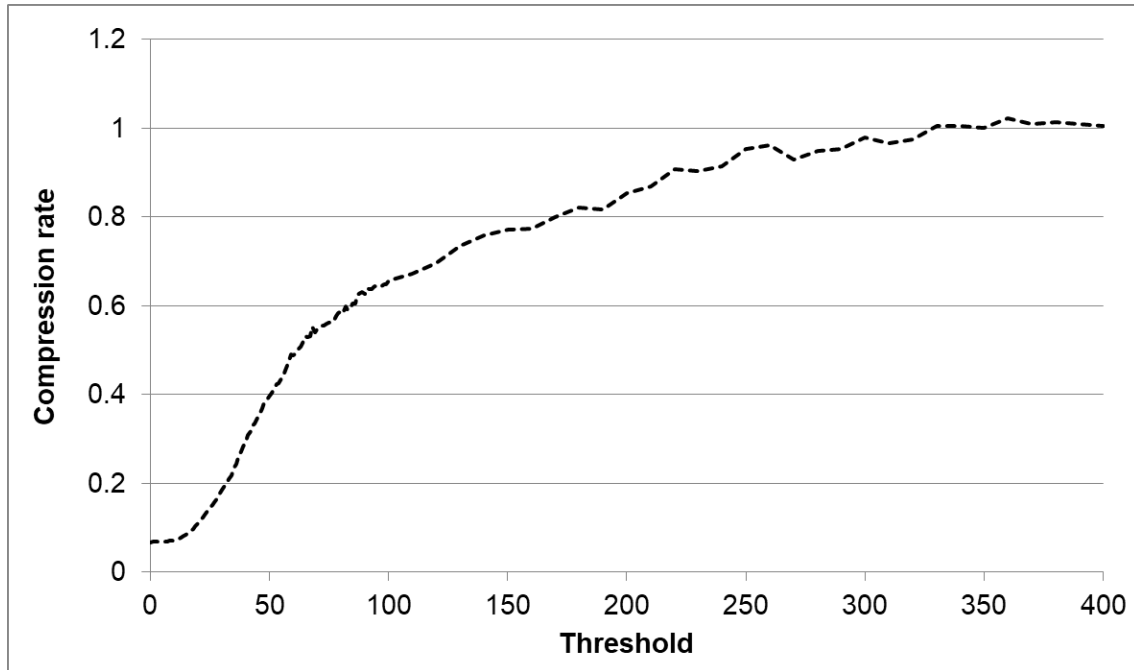


Figure 55: Compression rate over increasing distance measure threshold

The first observation from the graph in Figure 55 is that higher threshold values lead to higher compression with a growth rate resembling a logarithmic function that is slightly shifted up⁹.

With regard to requirement 2, the best threshold would achieve a compression of close or equal to 1. This would suggest a value in the region of 300 for this particular input data. There is, however, also requirement 1 which has a higher priority and therefore may overrule this initial choice. In general, it is worth noting that compression rates greater than 1 represent “lossy” compression. As the name suggests, this has the effect that information is lost by allowing too much deviation within clusters and therefore incurring misclassifications. The extreme would be to group all glyphs in one single cluster which would also be the maximum achievable (lossy) compression. Taking this into account,

⁹ Small drops e.g. at around a threshold of 200 and 275 can be explained by how the adaptive vector quantisation grows clusters. As this happens in an iterative and adaptive way it is natural that the prototypes representing clusters will develop differently for different thresholds. This can potentially lead to small deviations from the general trend.

threshold values beyond 350 can be disregarded in this experiment as they are guaranteed to violate requirement 1.

At this point it is helpful to introduce accuracy acc and error rate err in order to allow for a graphical representation of how growing thresholds affect the quality of clustering results:

$$acc(In, t) = \frac{|A|}{n}$$

With A the set of valid glyphs in the clustering result (i.e. those not misclassified per requirement 1 from above), n the number of glyphs in the input data In and t the used clustering threshold.

The error rate can then be written as:

$$err(In, t) = 1 - acc(In, t)$$

Accordingly, a clustering result with no misclassifications will yield an error rate of 0 while deteriorating clustering results will converge towards an error rate of 1.

Figure 56 shows now how the error rate behaves over the same range of distance measure thresholds as in Figure 55.

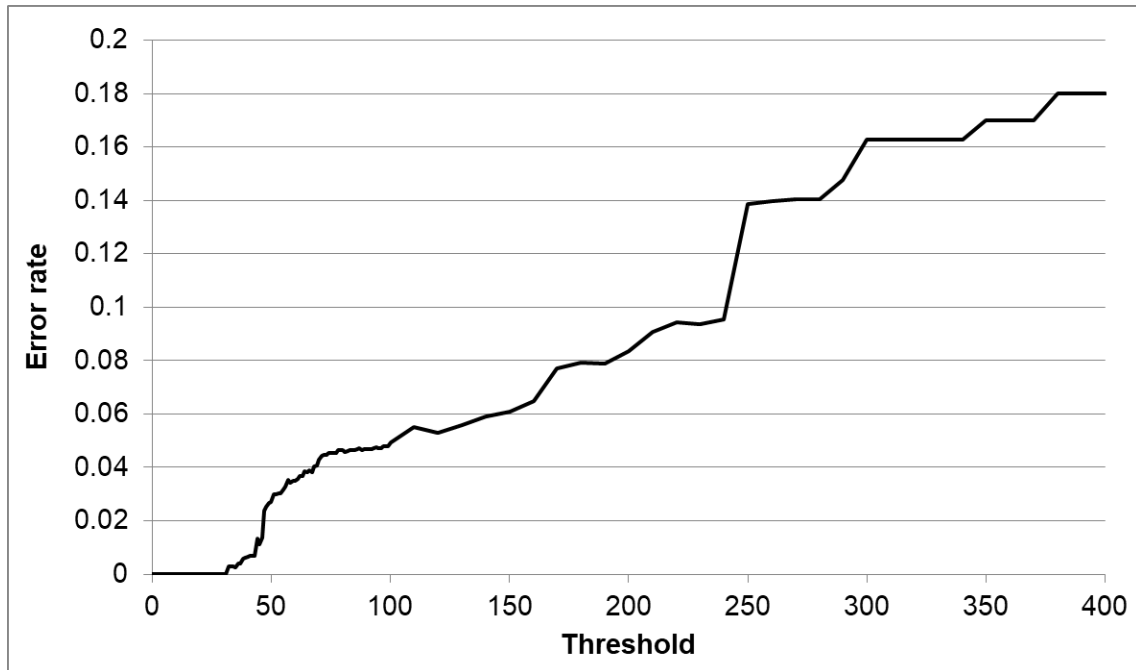


Figure 56: Error rate over increasing distance measure threshold

As the true class labels for all input glyphs are known in this supervised scenario, it is possible to calculate the actual error rates err . The error rate graph also shows a mostly continuous growth but closer to a linear function and shifted to the right. This right shift represents the region in which the combination of glyph representation, distance measure and clustering algorithm can achieve a result containing only pure clusters i.e. completely free of misclassifications as per requirement 1 from above.

Locating the threshold of 300, which would have been a good candidate with regard to requirement 2, now comes up as a rather poor choice, prompting an error rate of slightly over 16% which would clearly violate requirement 1. The only possible range of threshold values for the given input data lies between 0 and 35.

Combining the findings from these investigations leads to the following criterion on how the appropriate threshold t_a can be determined for a specific input In :

$$t_a = \max(t) \mid t \geq 0 \text{ and } err(t, In) = 0$$

To summarise, the appropriate threshold is identified as the greatest (rightmost) value for which the error rate is still 0 (i.e. before it starts to rise). With regard to the above results, this translates into a threshold of 35 which should be chosen to process this particular input data.

6.6.2 Unsupervised – Using a Surrogate Metric for Error Rate

In the previous section it was shown how an appropriate threshold can be determined from the graphs of compression and error rate. Unfortunately, these are only available if the true classes of all input glyphs are known. In order to be able to process any type of previously unseen documents (potentially containing unknown characters and symbols) it is therefore necessary to devise a surrogate metric that is independent of any background knowledge (i.e. can be calculated solely from the input data). This follows an unsupervised approach.

The search for such a surrogate metric can be further narrowed down to a function which only needs to replicate the error rate. This can be concluded from the threshold criterion

defined above which only depends on *err* and no longer on *comp*¹⁰. Also, the surrogate metric does not need to be identical to the error rate but should display the same overall trend and, most-importantly, location of characteristic points such as local maxima and inflection points in order to allow the application of the previously established threshold criterion.

A solution to this problem can be found in the way clusters behave as a result of applying less restrictive thresholds (i.e. greater values of *t*) allowing a greater distance between instances belonging to the same cluster. Even without knowledge of the actual classes of glyphs it is still possible to calculate the relative extent of clusters based on the distance function used during clustering. This leads to the concept of the intra cluster distance.

$$dist_{IC}(C) = \max_{G \in C} \delta_c(G, P)$$

It basically captures the dimension of an individual cluster *C* as given by its outermost glyph in relation to the prototype *P* constituting the origin.

In order to arrive at a metric representing the whole clustering result this has to be integrated into a combined value based on all clusters. The arithmetic mean is one of the most common ways to characterise a whole set of data through a single value. The average intra cluster distance $dist_{AIC}$ for a clustering *Cl* can be defined as:

$$dist_{AIC}(Cl) = \frac{1}{|Cl|} \sum_{C \in Cl} dist_{IC}(C)$$

While the average intra cluster distance could be shown to follow the same overall trend as the error function it turned out to be not very conclusive for the task of selecting a threshold as it tends to smooth out graph features and does not replicate characteristic points well enough on which the selection process is based. As a way forward the maximum intra cluster distance $dist_{MIC}$ proved more robust.

$$dist_{MIC}(Cl) = \max_{C \in Cl} dist_{IC}(C)$$

¹⁰ Requirement 2 (which relates to compression) was translated into the greatest (rightmost) threshold in the error rate graph.

It reflects the greatest difference of the distance measure between any glyph and its cluster prototype among all clusters. Or in other words, it looks at how different the most dissimilar glyph of any cluster is, across all clusters. Intuitively, this would be the first candidate to lead to a misclassification as the process cycles through less and less restrictive thresholds.

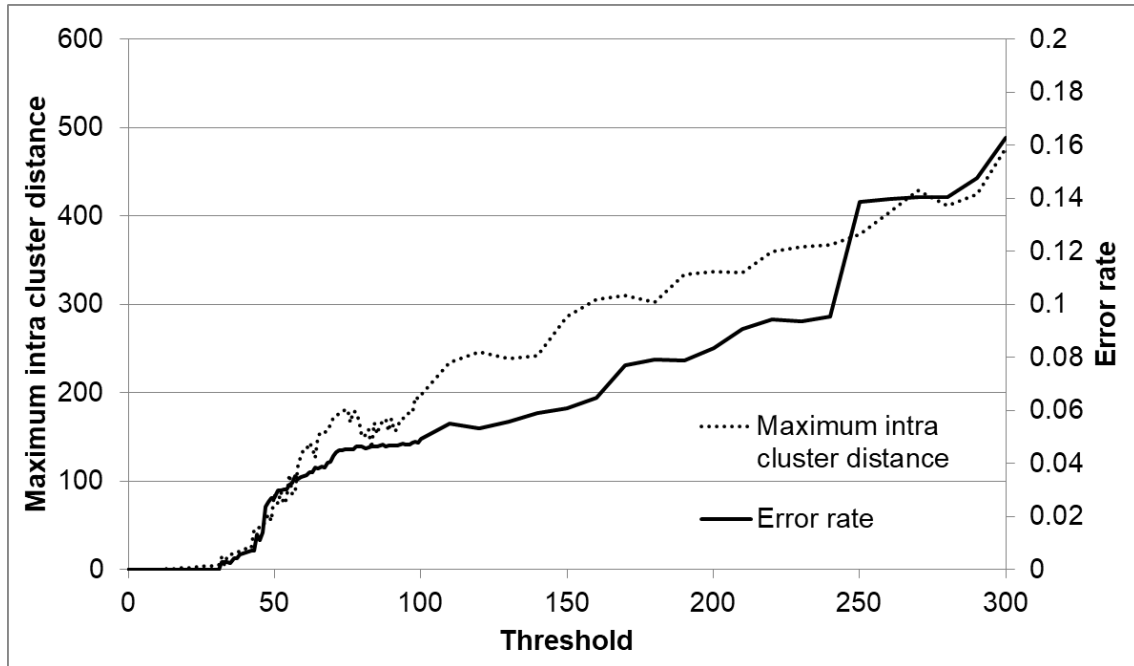


Figure 57: Maximum intra cluster distance and error rate over increasing threshold

Figure 57 shows an overlay of the maximum intra cluster distance and the error rate for clustering results with increasing threshold on the dataset German Fraktur (o) from before. A comparison reveals very similar curves – not only with regard to the general trend but also, and most importantly, related to the point where they start rising from zero. As discussed before, this is the point where the appropriate threshold can be located. The fact that the maximum intra cluster distance shows the same behaviour as the error rate regarding this critical point confirms that it can be used as a surrogate metric for application in an unsupervised setting on this particular input data. The hypothesis at this point is that the presented approach would also work for other types of input data indicating that a generic solution has been found.

6.6.3 Generalisation

In order to support the hypothesis from above it is required to show that the surrogate metric will generally display the same behaviour as the error rate for different types of input data (the discussion in 6.6.2 was based on one specific dataset).

To this end a collection of different datasets representing various types of documents was used. In addition to the real-world glyph images (i.e. scanned from actual historical books) used in the experiments so far, synthetic examples were leveraged as well in order to cover a broader spectrum and to manage the effort that is required when manually creating ground truth. Synthetic ground truth can be generated efficiently by creating born digital documents first (i.e. in a text editor) and then rendering glyph images using different fonts. Assigning the true class labels is in this context a trivial task only requiring a character lookup as the text is already available in an encoded format.

The question whether the maximum intra cluster distance is an acceptable surrogate for the actual error rate can be answered by testing how similar the two functions are for a whole range of document types such as the original and additional datasets. To measure the similarity of these two functions it is possible to calculate the Pearson Correlation Coefficient (PCC). The PCC is defined as the covariance of the pair of variables (here as given through the error rate and maximum intra cluster distance) divided by the product of their standard deviations. For samples, this can be obtained by calculating estimates of covariance and standard deviations from the available data points. With the symbols used in this work this leads to the following formula to calculate the correlation coefficient $corr$ over all tested threshold values up to t_{max} :

$$corr = \frac{t_{max} \sum_{h=1}^{t_{max}} err(h) dist_{MIC}(h) - \sum_{h=1}^{t_{max}} err(h) \sum_{h=1}^{t_{max}} dist(h)}{\sqrt{t_{max} \sum_{h=1}^{t_{max}} err(h)^2 - (\sum_{h=1}^{t_{max}} err(h))^2} \sqrt{t_{max} \sum_{h=1}^{t_{max}} dist(h)^2 - (\sum_{h=1}^{t_{max}} dist(h))^2}}$$

It can be noted that $dist_{MIC}$ depends on t via its corresponding clustering result Cl .

Correlation values can range from -1 to 1 with

- 1 : total positive linear correlation

- -1 : total negative linear correlation and
- 0 : no linear correlation.

A strong positive linear correlation means that if the error rate goes up then the corresponding maximum intra cluster distance will go up as well (proportionally).

Figure 58 shows the correlation coefficients for the intra cluster distance and error rate, obtained from the proposed clustering method on the German Fraktur (o) dataset (as used before) and the additional datasets from synthetic images (s).

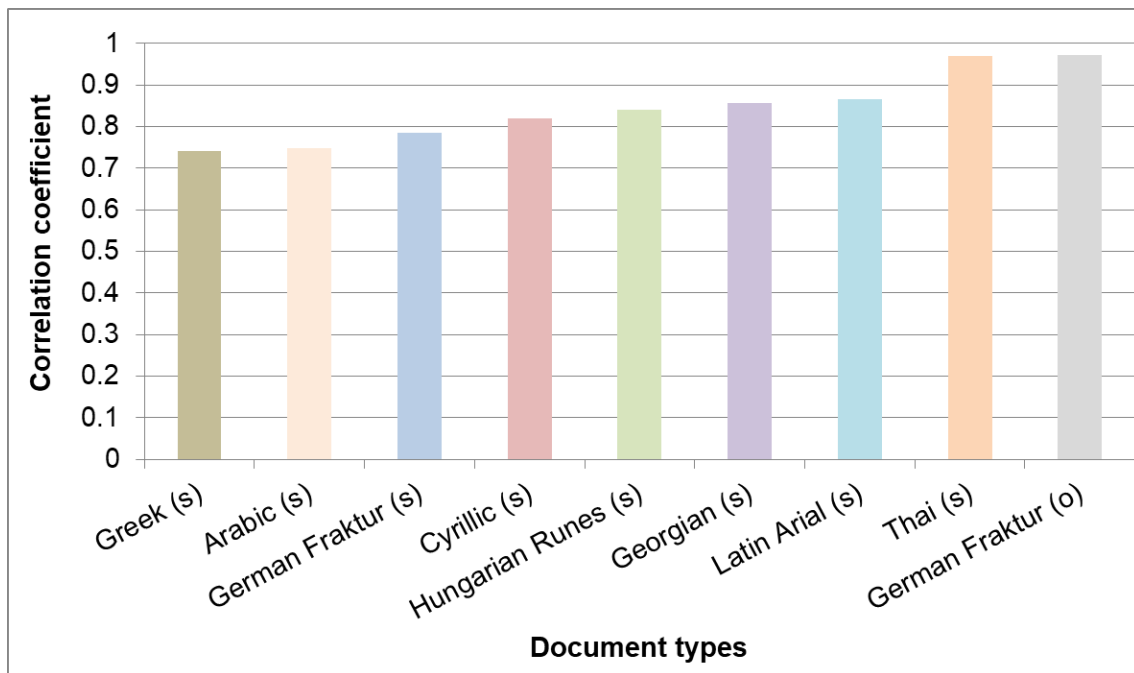


Figure 58: Correlation between maximum intra cluster distance and error rate

Overall, it can be stated that maximum intra cluster distance and error rate are indeed highly correlated with values close to 1. Two datasets (including the one from scanned originals) reach almost total linear correlation with a coefficient of 0.97. The two datasets with the lowest values display still a very high correlation with coefficients of 0.74 and 0.75.

As all datasets show the same general trend it can be concluded that it was not a coincidence in the initial experiments but that there is a systematic relation between the two functions and that the maximum intra cluster distance can hence be used as a surrogate measure in lieu of the actual error rate.

6.6.4 Calculating the Threshold

Having established that the maximum intra cluster distance can be used to predict the error rate, it is now a matter of repeatedly running the clustering algorithm with increasing threshold values to observe the relevant points in the curve. This adds computational complexity but can be considered of constant order as the range of thresholds to be tested is small. The stop criterion is when the initial steep rise starts to slow down (see Figure 59 at a Threshold of 80).

The process of ascertaining an appropriate threshold then depends on two key points in the graph:

1. The point where the maximum intra cluster distance starts to rise above 0 (left dotted line in graph below) i.e. glyphs with variation are contained in clusters.
2. The intersection of the linear approximation (regression) of the steep curve section with the x-axis (dotted line on the right). The start of the steep section indicates the turning point where increasingly more errors are made. This part of the curve can be isolated by its significantly higher and near constant gradient.

For these calculations it is beneficial to smoothen the curve i.e. to apply a moving average (interval of 10 in graph below) to reduce the effect of small deviations.

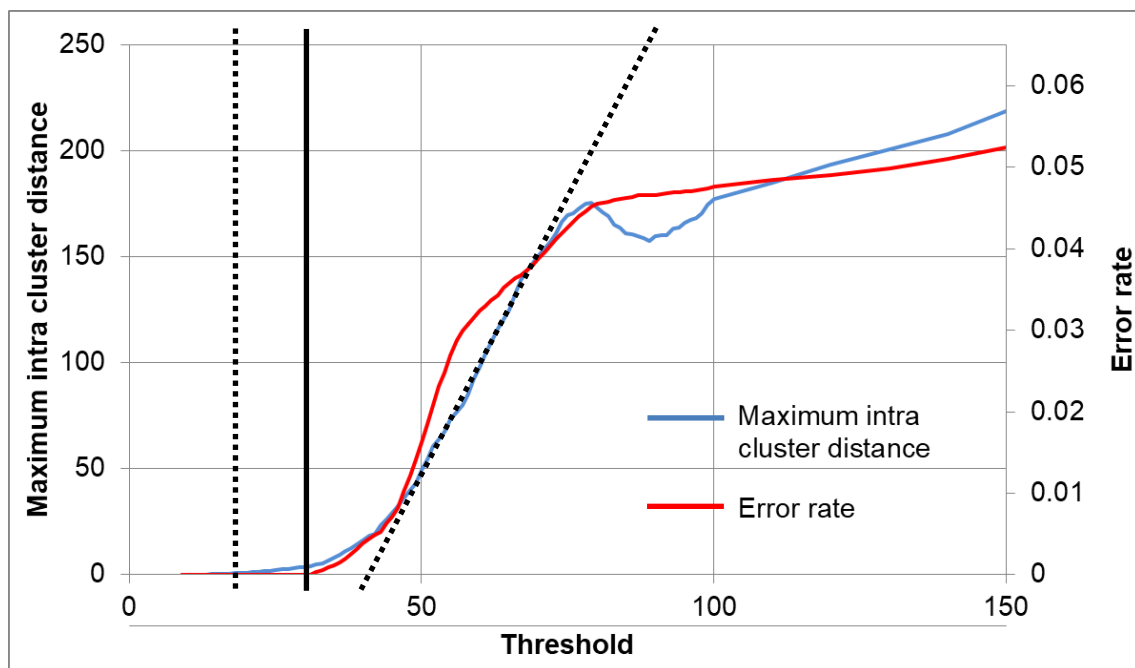


Figure 59: Close-up of the interval relevant to the threshold selection problem.

The two points described above can now be used as lower and upper bound for the threshold. The actual choice of an appropriate threshold should then be in the middle of this range so as to benefit from the compression effect as much as possible while avoiding actual misclassification which will gradually start with increasing threshold values.

Accordingly, the output of the described approach on the used German Fraktur (o) dataset is a value of 30 as calculated appropriate threshold. The overall output from the clustering method is then the clustering result that was obtained with this threshold of 30 as the identified best solution.

The actual graph and resulting threshold will always be specific to the particular input data as well as the concrete feature representations and distance measures. The findings presented up to now show the general feasibility of the approach but do not include any form of parameter tuning yet. This will be followed up in Section 6.8 Evaluation.

6.7 Assigning Code Points

The last step towards encoding unstandardised characters is to deal with the problem of how to assign code points.

The output from running the adaptive glyph clustering algorithm is a set of separate clusters, each represented by one prototype (the codebook entry from clustering).

The actual meaning of glyphs in these clusters (i.e. what character they represent) is not known and will somehow need to be determined. As indicated earlier, for completely accurate results this will require manual labelling i.e. a human user assigning a valid Unicode code point to each cluster which is then automatically conferred to each member. The compression achieved in the clustering result indicates the reduction in effort related to a completely manual labelling task whereby each single glyph would have to be looked at individually.

Depending on the nature of the content, however, it is possible that no entry exists in the Unicode standard, marking the discovery of an unstandardised character. The next step towards resolving such cases should be to consult non-official recommendations for character encoding. Such recommendations often feed into the general Unicode standard with a chance of being adopted into the official code charts in future updates. An example

with particular relevance to this work is the Medieval Unicode Font Initiative (MUFI) [66]. MUFI is a workgroup of scholars that aims at achieving standardisation for the encoding and display of special characters as typically found in historical documents. While providing clarification on the use of characters that are part of the Unicode standard, they are also proposing additions and make recommendations as to what temporary code points should be used.

This leads to the most generic approach that can be taken – the application of Private Use Area code points as defined in Unicode. Such code points are basically reserved for individual applications and no further meaning is prescribed by the standard. Developers are free to use Private Use Area code points in their applications and can be confident not to interfere with other fully defined characters. This, however, can very easily lead to different applications using the same reserved codes for different purposes. In this context, further recommendations (such as MUFI) can come in and provide guidance on Private Use Area codes in a specific field.

In the case that an unstandardised character as clustering output cannot be found in unofficial recommendations either it is always possible to assign Private Use Area codes completely independently. In fact, this can be done as first step after clustering to provide every cluster with an initial label. Manual labelling according to Unicode and/or lookup in further recommendations can then follow as a refinement process.

As Private Use Area code points behave in exactly the same way as fully specified ones, it is possible to store documents in such an intermediate state while still allowing the same applications as for other “fully” encoded documents. The main difference is then that the document is encoded using a document specific alphabet. Any user interface (e.g. for searching or editing) would have to provide means for interacting with this custom set of characters (for instance in the form of a dynamically generated virtual keyboard). Also, rendering of such documents is very straightforward if a corresponding document specific font is derived from the cluster prototypes. This general concept has been explored in [67], [47] and [68].

6.8 Evaluation

With an end-to-end approach in place, it is now possible to empirically verify the observations made so far and to characterise the overall achievable results based on the available datasets.

6.8.1 Datasets

All datasets consisted of glyph images that were cropped from documents scanned in greyscale at a resolution of 300 dpi. Table 2 summarises the nature of the different datasets.

Table 2: Datasets used for evaluation

Dataset	Glyphs	Classes	Composition	Source
Greek (s)	4900	49	Upper/lower case	synthetic
Arabic (s)	2900	29	No differentiation	synthetic
German Fraktur (s)	5200	52	Upper/lower case	synthetic
Cyrillic (s)	6600	66	Upper/lower case	synthetic
Hungarian Runes (s)	3000	30	No differentiation	synthetic
Georgian (s)	3300	33	No differentiation	synthetic
Latin Arial (s)	5200	52	Upper/lower case	synthetic
Thai (s)	4400	44	Consonants	synthetic
German Fraktur (o)	5500	55	Upper/lower case, ligatures	originals

Each dataset contained 100 glyph images per class. Class labels (ground truth information) as required for evaluation were acquired automatically from the digital source for the synthetically generated datasets and in the case of German Fraktur (o) assigned manually.

6.8.2 Overall Results – Initial Configuration

Figure 60 shows the results for error and compression rate as obtained from the proposed adaptive glyph clustering method (using the features and distance measures as discussed in the previous sections) with the initial weights of 80% for w_t (template matching) and 20% for w_f (feature based).

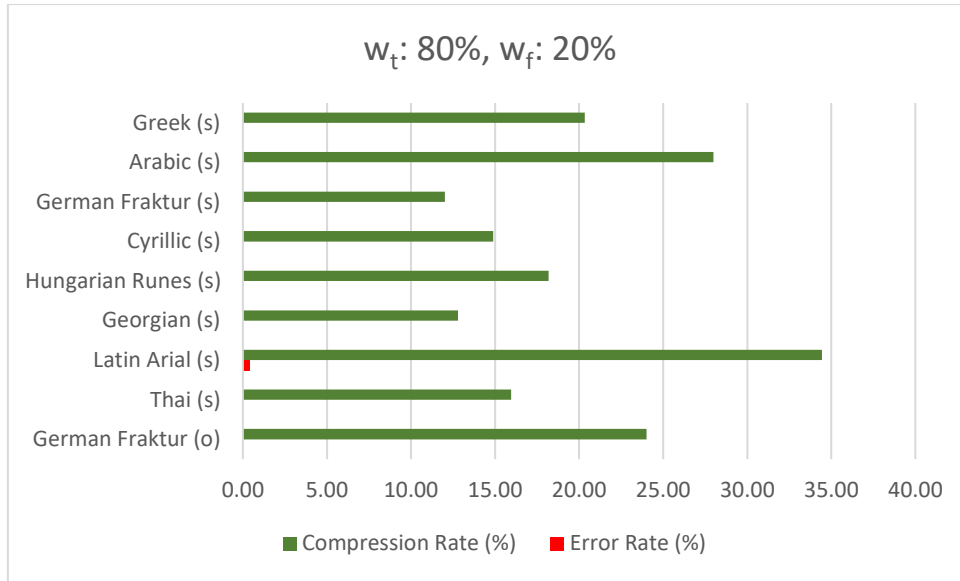


Figure 60: Compression and error rate for w_t : 80% and w_f : 20% (initial configuration)

In addition to the German Fraktur (o) dataset which was already studied during the research and development stages it can now be seen that the intended compression (clustering) effect is indeed achievable across a wide range of language/script combinations. It can also be seen that the different datasets lent themselves slightly differently to the proposed approach. Latin Arial (s) reached the highest compression rate but at the cost of a small number of mistakes with an error rate of 0.38% (see Table 3 for detailed results). Among the error free results, Arabic (s) performed best and German Fraktur (s) achieved the lowest compression rate. These results indicate that the proposed method is working for the majority of datasets but, in its current configuration, may still violate requirement 1 from Section 6.6 related to avoiding misclassifications as manifested through an error rate greater than 0.

A result achieving higher compression but at the cost of errors, as observed for Latin Arial (s), points in the direction of insufficient input from the template matching distance measure which was attributed with producing safer results in the sense of better avoiding errors (Subsection 6.4.3). The effect of shifting the balance between the weights assigned to the two distance measures (template matching and feature based) is therefore further investigated in the following.

6.8.3 Revisiting the Combined Distance Measure Weights

To systematically study the impact of different weighting combinations, both directions were analysed - with increased resolution in the most promising region. A short notation of (w_t/w_f) will be used for the different configurations in the following. Starting from (80/20), the next step was to run the proposed method with alternative weights of (70/30) and (90/10). As the direction of (90/10) displayed better results, this was followed further to (100/0) denoting a configuration where solely template matching is considered. Observing worse results at (100/0), a further step of (95/5) was added leading to the complete results as shown in Table 3.

Table 3: Compression and error rate for different combinations of weights

w_t/w_f (%)	100/0		95/5		90/10		80/20		70/30	
Dataset	Error %	Comp. %	Error %	Comp. %	Error %	Comp. %	Error %	Comp. %	Error %	Comp. %
Greek (s)	0.54	59.76	0.00	48.51	0.00	16.61	0.00	20.33	0.00	20.33
Arabic (s)	3.57	87.50	0.71	59.57	0.00	22.58	0.00	28.00	0.00	15.73
German Fraktur (s)	0.00	28.11	0.00	14.90	0.00	18.71	0.00	12.01	0.26	11.90
Cyrillic (s)	2.32	74.16	0.00	66.00	0.00	40.24	0.00	14.90	0.30	18.70
Hungarian Runes (s)	6.67	90.91	0.00	66.67	0.00	26.79	0.00	18.18	0.00	13.39
Georgian (s)	0.00	22.92	0.00	19.08	0.00	27.73	0.00	12.79	0.00	11.54
Latin Arial (s)	4.74	94.55	0.00	27.51	0.00	32.10	0.38	34.44	0.00	16.99
Thai (s)	4.39	68.33	0.00	34.75	0.00	19.07	0.00	15.95	0.16	11.17
German Fraktur (o)	0.00	77.46	0.00	20.33	0.00	30.39	0.00	24.02	0.00	20.00
Average	2.47	67.08	0.08	39.70	0.00	26.02	0.04	20.07	0.08	15.53

Results where an error rate greater 0 occurred are highlighted in the table, allowing better identification of configurations which were in violation of the criterion relating to misclassification. While not achieving maximum compression, it can be concluded that (90/10) is the optimal configuration as it avoids errors entirely, across all datasets.

This should be seen as the recommended configuration of the proposed method for common use if no prior knowledge on the input data is available. The breadth of script/language combinations in the datasets used for evaluation gives a good level of confidence that this poses a universally applicable solution.

6.8.4 Overall Results – Optimal Configuration

Following on from the considerations on the best setting of weights, the overall results of the (90/10) configuration as optimal solution are now shown in Figure 61.

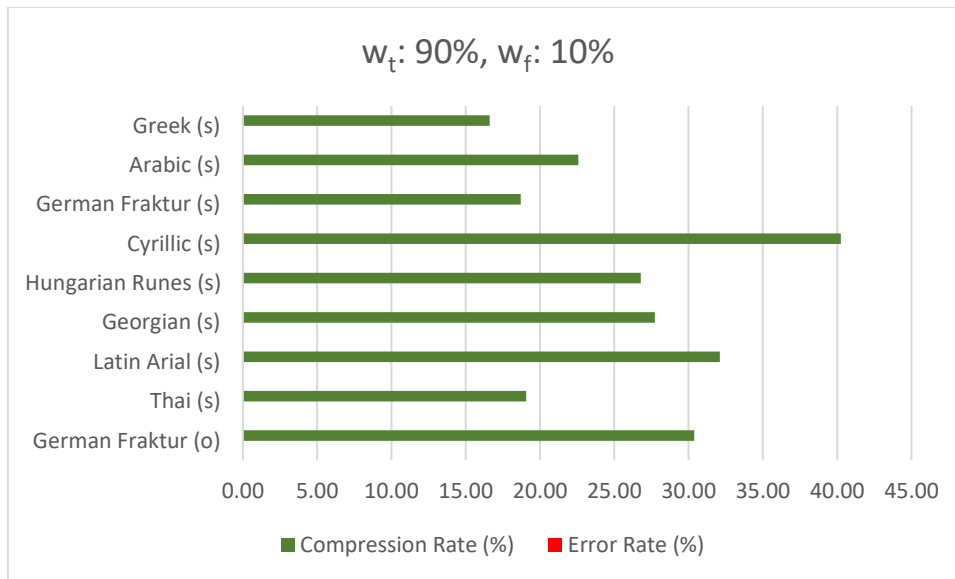


Figure 61: Compression and error rate for w_t : 90% and w_f : 10% (optimal configuration)

Besides leading to a completely error free result it can also be noted that the achievable compression goes up on average, compared to (80/20). As before, there is variation between the different datasets. Cyrillic (s), for instance, fares particularly well at (90/10) while Greek (s) loses around 4 percentage points compared to (80/20).

The assessment of (90/10) as optimal configuration was based on all 9 datasets used in the presented work. This reflects the aim of developing a universally applicable method, i.e. one that does not make any assumptions and works reliably across a diverse range of input data. At the level of individual datasets, however, this can mean a reduction of achievable compression. For instance, referring to Table 3. it can be seen that Greek (s) would perform best at (95/5) but this configuration was dismissed as it would produce invalid results when applied to Arabic (s).

If background information about the nature of the input data were available in real world application scenarios, it could potentially be utilised to arrive at a more targeted configuration for the material at hand. Any suspected similarity to the datasets considered in this work could be used as a starting point for further application specific weighting optimisation. Any such attempts would need to be empirically verified through separate evaluation on this particular type of material (requiring the creation of ground truth information).

6.8.5 Discussion of Results and Benefits

Variation in the individual compression rates of different datasets aside, it has been shown that the proposed approach can achieve a significant clustering effect across the board. For a universally applicable method that does not come at any extra cost other than computational power (as opposed to manual work) while at the same time promising to avoid the introduction of errors, this has the potential to yield significant benefits in terms of effort reduction in real world applications.

Related to the task of labelling glyph instances in order to arrive at a properly encoded representation, the following two major comparisons can be made:

- Proposed approach vs no clustering
- Proposed (adaptive) approach vs fixed threshold (non-adaptive) clustering

In the first case, if no clustering is applied, the number of manual labelling steps would be equal to the total number of glyphs in the input. This constitutes the worst-case scenario. If, on the other hand, the proposed approach is applied, the number of manual labelling steps will be equal to the (typically much lower) number of clusters in the output. This number follows automatically from the output of the method or, if evaluation results are available, it can also be derived from the number of true classes in the input (see Table 2) and the ascertained compression rates (see Table 3) via the relation of $\text{compression} = \text{number of true classes} / \text{number of clusters in output}$ (as per Subsection 6.6.1). The information on required manual labelling steps as well as effort reduction is presented in Table 4.

Table 4: Manual labelling effort – proposed approach vs no clustering

Dataset	Glyphs	Classes	Steps no clustering	Steps proposed method	Effort reduction (steps)	Effort reduction (%)	Compression (90/10) (%)
Greek (s)	4900	49	4900	295	4605	93.98	16.61
Arabic (s)	2900	29	2900	128	2772	95.57	22.58
German Fraktur (s)	5200	52	5200	278	4922	94.66	18.71
Cyrillic (s)	6600	66	6600	164	6436	97.51	40.24
Hungarian Runes (s)	3000	30	3000	112	2888	96.27	26.79
Georgian (s)	3300	33	3300	119	3181	96.39	27.73
Latin Arial (s)	5200	52	5200	162	5038	96.88	32.10
Thai (s)	4400	44	4400	231	4169	94.76	19.07
German Fraktur (o)	5500	55	5500	181	5319	96.71	30.39

The combined information shows very impressively that already small compression effects will result in a very significant effort reduction related to the task of glyph labelling.

Applying the proposed method on the 9 datasets resulted in a labelling effort reduction of close to 96% compared to no clustering and labelling each glyph separately.

This is due to the fact that each cluster will typically contain numerous glyphs which can all be assigned their (identical) label in one single manual step. Equally, compression rate improvements in the higher ranges (moving towards the maximum of 100% for error free results) will only marginally reduce the required labelling effort.

The second case considers how much the required labelling effort is improved by moving from a fixed threshold to an adaptive approach that calculates and applies input-specific thresholds. In order to analyse this aspect, the actual thresholds t_a (as calculated by the proposed method described in Subsection 6.6.4) have to be considered. The value to be selected as fixed threshold will have to be the lowest (most restrictive) across all datasets as error free results are required. This is necessary to replicate a universally applicable fixed threshold approach that is independent of knowledge about the input data in the same way as the proposed approach.

Table 5: Manual labelling effort – proposed (adaptive) vs fixed threshold approach

Dataset	t_a	Compression (90/10) adaptive (%)	$t_{\text{fixed}} = \min(t_a)$	Compression (90/10) fixed (%)	Steps fixed threshold	Steps proposed method	Effort reduction (steps)	Effort reduction (%)
Greek (s)	65	16.61	65	16.61	295	295	0	0.00
Arabic (s)	101	22.58	65	13.86	209	128	81	38.62
German Fraktur (s)	101	18.71	65	11.16	466	278	188	40.35
Cyrillic (s)	141	40.24	65	14.77	447	164	283	63.30
Hungarian Runes (s)	140	26.79	65	13.95	215	112	103	47.93
Georgian (s)	133	27.73	65	11.91	277	119	158	57.05
Latin Arial (s)	107	32.10	65	17.26	301	162	139	46.23
Thai (s)	101	19.07	65	13.58	324	231	93	28.79
German Fraktur (o)	133	30.39	65	18.09	304	181	123	40.47

Table 5 shows that the lowest individual threshold occurred for Greek (s). Using this as the safe fixed threshold at which all 9 datasets can be processed without errors means significantly lower compression rates for the other datasets. Overall, the improvement

due to using the proposed (adaptive) threshold compared to a fixed (safe) threshold comes down to an effort reduction of around 41%.

The two presented comparisons show clearly, how the proposed approach can have a very positive impact on real world applications by significantly reducing the amount of required manual work.

The overall positive evaluation results show that the second major objective of this work (development of an approach to automatically identify and encode unstandardised characters) has been successfully achieved.

6.9 Summary

This chapter presented the research and development stages towards a novel adaptive glyph clustering approach to enable detection and encoding of unstandardised characters originating from scanned documents in an efficient manner. Key points were the development of an iterative clustering algorithm, operating on a combined distance measure that incorporates template matching and feature analysis, and the adaptive clustering threshold calculation based on a surrogate metric emulating the error rate which is unknown in an unsupervised setup. A comprehensive evaluation showed the validity of the proposed approach and its potential for reducing manual work in real world applications.

7 Discussion

The pagecontent format specification presented in this work has seen major uptake since its inception and made a significant contribution to the fields of Document Image Analysis, Digital Libraries and Archives as well as Digital Humanities. The following discussion provides details on success factors and evidence to support this claim.

The pagecontent format as presented in this work was initially developed as a sub-specification of a wider framework (PAGE) which was also developed by the author of this thesis. This wider framework has not seen much uptake – in stark contrast to the pagecontent format. As a result, the acronym PAGE (sometimes also PAGE-XML) has become the de-facto name for the pagecontent format. This was not a decision by the author but rather what naturally happened within the community. Any mention of PAGE in the following is meant in the way in which it is used by the community – as a memorable name for the pagecontent format which was developed within the scope of this work.

The adaptive glyph clustering approach as a possible solution to the problem of encoding unstandardised characters, must realistically be seen as a smaller contribution of the presented work. Nonetheless, there is potential for it to be utilised in practical applications as will be discussed at the end of this chapter.

Next is a discussion of factors that led to the wide success of PAGE. This is followed by evidence of the positive impact the format has had as well as real world applications of the proposed adaptive glyph clustering approach.

7.1 Success Factors of PAGE

The main factors that have led to PAGE becoming one of the most important formats for representing digitised documents can be broken down into

- Technical validity
- Format features
- Official tool support
- Datasets

- Competitions

It must be noted that in particular the points on official tool support, datasets and competitions are not only related to efforts by the author but the PRIMA Research Group as a whole.

7.2 Technical Validity

On a most basic level it can be said that the format would not have seen any major success if it had not been technically sound. Great care was taken to ensure the schema implementation would be fit for purpose. On the one hand, every schema release was systematically validated against the XML Schema Definition (XSD) to rule out any syntax errors and formal problems. On the other hand, every feature included in PAGE has been extensively tested in a reference tool implementation. This hands-on approach has been crucial for identifying and correcting numerous problems before releases. Also, it has provided valuable feedback leading to improvements in later versions.

Another important aspect that has been taken very seriously from the start was documentation. Every PAGE release has its own documentation, embedded inside the schema definition using the appropriate annotation/documentation structures as defined in the XSD standard. This allows not only for up-to-date documentation manuals to be generated but also enables Integrated Development Environments (IDEs) and other tools to provide context specific help.

Lastly, every new version is accompanied by specific release notes. Release notes provide a complete list of changes in relation to the previous version and give advice on how to migrate existing documents and tools.

Further confirmation for the technical validity of PAGE can be found in the following sections on its actual use.

7.3 Format Features

This point is related to the competitive advantages of PAGE (i.e. the gap it fills) compared to other formats. Important features PAGE offered from the start, setting it apart from its main contenders (such as ALTO) were:

- More detailed hierarchy of text elements – down to glyph level
- Ability to store text on all hierarchy levels
- Precise region outlines
- Greater level of detail for region types and attributes
- Advanced region order representation
- Concept of layers

Later features that fostered uptake of the format and application in further domains were related to:

- Support for handwritten text
- Customisation mechanisms
- General nesting of regions
- Graphemes
- Semantic labels

In particular the ability to adapt the format via customisation (within well defined constraints) opened up new application scenarios that may otherwise have been considered too marginal for inclusion of dedicated elements and attributes into the specification.

Also, general format improvements have played a role, such as more efficient storage of coordinate points representing shapes and efforts to simplify the structure (and readability) of the schema.

7.4 Official Tool Support

The most prominent software to support PAGE is Aletheia [69] which can also be seen as reference implementation (see Figure 62 for a screenshot). Aletheia (from Greek – truth) is an advanced document analysis system supporting manual annotation of digitised documents as well as fully automatic OCR. The features in Aletheia directly correspond with elements of PAGE. Aletheia is available in different versions from a free to use community edition to a commercial product with additional support and production features [70]. Development of a web-based solution is currently underway [71].

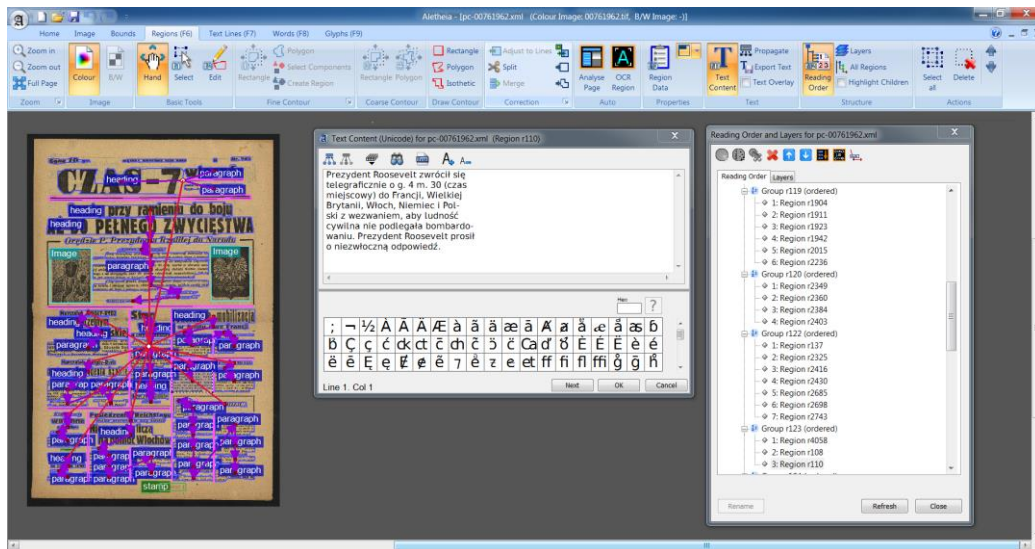


Figure 62: Editing a Newspaper Page in Aletheia [11]

The second most important software developed at PRImA Research is the Performance Evaluation tool [72]. It provides a complete system for performance evaluation related to OCR methods (including layout and text) and is based on PAGE as central format. A screenshot of the tool being used for evaluating and inspecting layout analysis results is shown in Figure 63.

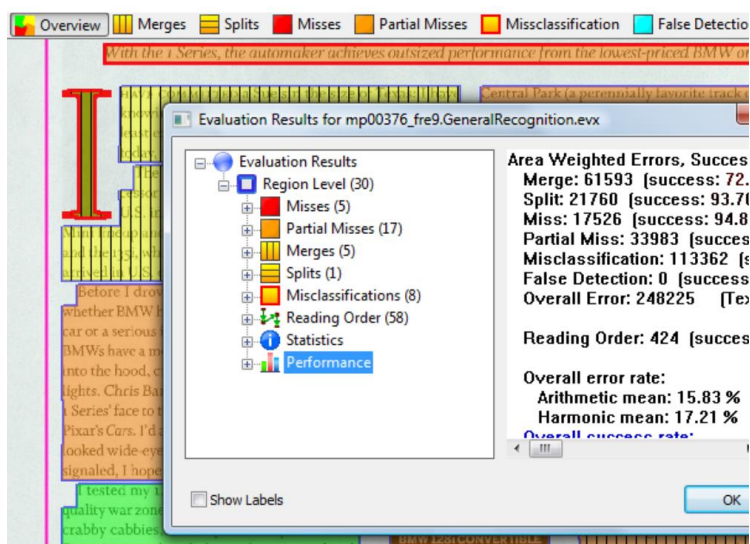


Figure 63: PRImA Performance Evaluation Tool [73]

Another major component are the PAGE Libraries which provide developers with resources to include in their own software projects [74]. This allows third party developers to implement PAGE input and output as well as internal object representations without

much effort. The PAGE Libraries support Java and C++ as programming languages and also include usage examples.

Further software includes auxiliary tools for handling various tasks as they may occur in production workflows, evaluation experiments and during data migration:

- The PAGE Converter and Validator [75] can be used to migrate existing documents from older versions to the latest format specification (the whole history of releases is supported). In addition, it can validate PAGE files against a set of rules as may be given in ground-truthing guidelines for a specific application scenario.
- The PAGE Metadata Scanner [76] is a command line tool that can analyse individual PAGE files to produce statistics on the usage of elements, availability of text, used languages etc. as per the set of supported attributes.
- Tesseract OCR to PAGE [77] is a command line tool that packages the popular open-source OCR engine Tesseract to produce PAGE as direct output. This allows for straightforward integration into production and evaluation workflow systems.
- The PAGE Extractor/Exporter [78] is a command line tool that allows specific information to be extracted from PAGE files. Main applications are the generation of image snippets representing individual layout elements (e.g. isolated table region or glyph images) as well as text serialisations (respecting reading order information if available). Again, this tool can be very easily integrated into workflow systems and/or used standalone to feed into development and evaluation tasks.
- The PAGE Viewer [79] is a Graphical User Interface (GUI) tool that can be used as standalone application for displaying page layout and text content of PAGE files overlaid on their corresponding image. This tool has proven particularly popular as it also supports other formats such as ALTO, FineReader XML and hOCR for which support is less readily available.

The substantial list of official PRImA tools shows the comprehensive support that is available for PAGE. The constant stream of website requests and software downloads can be seen as evidence for the success of PAGE.

7.5 Datasets

Datasets of document images with corresponding ground truth are major assets in the context of research and development activities. Without relevant datasets of critical mass it is virtually impossible to conduct effective research. One factor leading to the success of PAGE is the availability of large-scale datasets with content from major stakeholders in the libraries and archives sector.

The creation of the IMPACT Dataset of Historical Document Images [80] marked a milestone in terms of availability of images (over 600,000) and ground truth (approximately 45,000) from major European libraries. All ground truth was created in PAGE format and contains not only full text but also detailed layout information (region outlines etc).

Another major contribution was the ENP Image and Ground Truth Dataset of Historical Newspapers [81]. It comprises over 500 images of very densely populated newspaper pages together with comprehensive ground truth in PAGE format. A particular challenge for this dataset was the sheer amount of text information and the complex layouts as well as reading order structures that are typically found in newspapers.

PRImA's own Layout Analysis Dataset [82] representing contemporary documents is another significant resource in the field of document image analysis. Detailed layout ground truth including precise region outlines is provided in PAGE format. This dataset has been used in numerous competitions as outlined in the following section.

Various other datasets (with all ground truth in PAGE format) are available via the PRImA website [1]. Beyond, there are numerous datasets using PAGE, created by researchers that have no affiliation with PRImA. This will be further discussed in the Impact section below.

7.6 Competitions

Competitions evaluating advances in the state-of-the-art are important research community activities. The PRImA Research Group has a long-standing tradition of hosting such events [83].

Starting with the ICDAR2009 Page Segmentation Competition [29], events hosted by PRImA have relied on PAGE. More competitions were held, following the biennial schedule

of ICDAR (International Conference on Document Analysis and Recognition) and associated workshops:

- Historical Document Layout Analysis Competition 2011 [30]
- HNLA Historical Newspaper Layout Analysis 2013 [31]
- HBR Historical Book Recognition 2013 [84]
- REID Recognition of Early Indian printed Documents 2017 [85], 2019 [86]
- RASM Recognition of Historical Arabic Scientific Manuscripts 2018 [87]
- RDCL Recognition of Documents with Complex Layouts 2015 [32], 2017 [88], 2019 [89]

Utilising the general infrastructure and resources surrounding PAGE in competitions has led to further proliferation of the format. The provision of datasets in, and tools working with the format have not only stimulated uptake by other researchers but also contributed to interoperability.

7.7 Impact

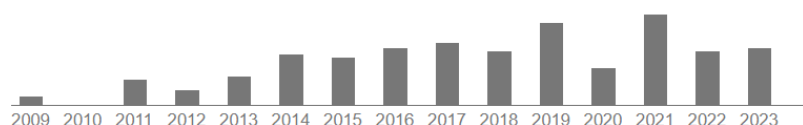
This section looks at the lasting (positive) effects that the presented work has had. Impact in that sense is largely characterised by the interest and practical uptake by external entities.

7.7.1 Citations

The first pointer to gauge the impact of the PAGE Format Framework can be taken from the number of citations the original publication [56] has received. At the time of writing, Google Scholar reports a total of 194 citations (Figure 64).

The page (page analysis and ground-truth elements) format framework

Authors	Stefan Platschacher, Apostolos Antonacopoulos
Publication date	2010/8/23
Conference	2010 20th International Conference on Pattern Recognition
Pages	257-260
Publisher	IEEE
Description	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.
Total citations	Cited by 194



Scholar articles [The page \(page analysis and ground-truth elements\) format framework](#)
S Platschacher, A Antonacopoulos - 2010 20th International Conference on Pattern ..., 2010
[Cited by 194](#) [Related articles](#) [All 17 versions](#)

Figure 64: Citations of the Original PAGE Format Framework Paper [90]

It is hard to confirm if this number is completely accurate, but it has been observed that numerous papers refer to PAGE without citing the original publication (citing related resources such as datasets and competitions instead). Overall, this indicates significant interest in the research community and, judging from the way the format has been adopted in other projects, datasets and tools, actual impact.

Taking a closer look at individual publications referencing PAGE one can find verdicts such as the following:

“In terms of annotation format, we note that most datasets use PAGE XML, three datasets use the COCO format, and only one dataset uses the VOC format.” (A survey of historical document image datasets [91])

“Many datasets use a specific XML schema for ground-truth in several domains. In some cases some specific standard formats derived from XML have been defined and used to represent ground-truth as it is the case of ALTO (Analyzed Layout and Text Object, at <http://www.loc.gov/standards/alto/>) or PAGE [53] for page analysis, ..” (Datasets and Annotations for Document Analysis and Recognition. In: Handbook of Document Image Processing and Recognition [92])

“The actual full text obtained by OCR is then encoded in the PAGE-XML format, which has been established as the richest format in the pattern recognition community by means of its use in scientific competitions and various research projects.” (OCR-D: An end-to-end open source OCR framework for historical printed documents [93])

“Every page image in the dataset contains annotation in the widely used XML-based PAGE format..” (ChronSeg: Novel Dataset for Segmentation of Handwritten Historical Chronicles. [94])

“The images are groundtruthed using PAGE XMLs¹ [sic] which is commonly used in document analysis.” (cBAD: ICDAR2017 Competition on Baseline Detection [95])

“We again use the PAGE XML Schema [9] which is well-established in the document analysis community and best serves our needs.” (cBAD: ICDAR2019 Competition on Baseline Detection [96])

“The well-known PAGE XML³ scheme is used for storing text region and baseline information.” (READ-BAD: A New Dataset and Evaluation Scheme for Baseline Detection in Archival Documents [97])

These quotes portray a general view that PAGE is well established and widely adopted in the field.

7.7.2 External Projects, Datasets and Tools

The following provides a list of further projects, datasets and tools that use PAGE as storage and/or exchange format, without the author having any direct affiliation:

- READ Project: ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset [98], Transkribus. A Platform for Automated Text Recognition and Searching of Historical Documents [99]
- tranScriptorium Project: Ground-Truth Production in the tranScriptorium Project [100], Handwritten Text Recognition on tranScriptorium Datasets: Bentham R0 [101], Handwritten Text Recognition on tranScriptorium Datasets: Bentham R1 [102]
- HisDoc Project: DIVA-HisDB dataset used in ICDAR2017 Competition on Layout Analysis for Challenging Medieval Manuscripts [103]
- The EScriptorium VRE for Manuscript Cultures [104]
- LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis [105]
- HDPa: historical document processing and analysis framework [106]
- Laypa: A Novel Framework for Applying Segmentation Networks to Historical Documents [107]
- OCR4all—An Open-Source Tool Providing a (Semi-)Automatic OCR Workflow for Historical Printings [108]
- VML-HTR dataset: Handwritten Text Recognition dataset, VML-AHTE dataset: Arabic Handwritten Text Line Extraction, VML-MOC: Multiply oriented and curved handwritten text line dataset [109]
- Handwritten Annotation Detection Dataset (AnnotationDB) [110]

The above does by no means constitute a complete list. Its main purpose is to give further evidence for the uptake of PAGE.

7.7.3 Influence on Other Formats

Another key outcome of the development of the PAGE Format Framework is the influence it has had on other formats. The PAGE specification addressed gaps in the existing landscape of formats and thus set the standard for a number of advanced features.

The most noteworthy adaptation concerned ALTO as the other major format related to the representation of digitised documents. ALTO is maintained by the Library of Congress, in particular by the Network Development and MARC Standards Office. One of the earlier

features that was first introduced in PAGE and then found its way into ALTO is the hierarchical level of glyphs (characters). Following the general success and growing importance of PAGE, efforts were undertaken to increase compatibility between the two formats. This led to the author being invited to join the editorial board of ALTO with the goal of working towards further standardisation in the field. Version 4.3 of ALTO [111] then saw the introduction of explicit reading order support as a direct adaptation of the PAGE specification (including the concept of ordered and unordered groups).

This convergence of key features does not render one or the other format redundant. On the contrary, it improves interoperability and data exchange between different systems and thereby allows to focus on the particular strengths for a specific application scenario.

7.7.4 REF Impact Case Study

The final piece of evidence for the success of the PAGE Format Framework is its inclusion in a Research Excellence Framework (REF) 2021 impact case study submitted by the University of Salford to Unit of assessment: 11 - Computer Science and Informatics. The case study is titled “Enabling digital transformation through effective digitisation” and it prominently places PAGE as a key contribution with regard to Underpinning research (section 2), References to research (section 3) and Details of impact (section 4) [112].

The case study received the best possible rating of 4 stars. According to the UK Research and Innovation (UKRI) REF quality level descriptors [113] this is classed as “Quality that is world-leading in terms of originality, significance and rigour.”

7.8 Applications of Adaptive Glyph Clustering

Even though the proposed approach of adaptive glyph clustering has not had the same impact as the PAGE Format Framework it still enables various practical applications and/or can make them more efficient.

As discussed in Section 6.7, if digitised documents with unstandardised characters are to be encoded in the most accurate way possible then there needs to be a stage where any such instances are identified and subsequently assigned a code point. This is normally a manual task as it is not reasonable to expect the existence of an OCR engine that supports unstandardised (unknown) characters.

Once such a manual code point assignment has been undertaken, however, its output will be an extremely valuable resource in the form of labelled examples which can be used for training of future OCR engines to support the newly discovered characters. Such a glyph labelling application can therefore be used for training data generation, or equally to produce accurately encoded document representations in the sense of semi-automatic OCR.

Another possible application, not requiring any manual intervention, is the faithful reproduction of document images in an encoded form that is not fully recognised i.e. relies on Private Use Area code points.

7.8.1 Training Data Generation / Semi-automatic OCR

Starting point of adaptive glyph clustering is a set of isolated glyph images as input. The problem of obtaining such instances via appropriate segmentation methods has been explicitly excluded from this work. For a production-grade system this would require more research and development e.g. following approaches such as presented in [114] and [115]. For proof of concept, a very basic segmentation system based on connected component analysis as presented in [116] has been used (Figure 65). It still depends on manual intervention, especially to combine multi-component glyphs which the limited approach cannot handle.



Figure 65: Basic Glyph Segmentation Tool [116]

The second system component then allows to execute the adaptive glyph clustering. It shows the list of all glyph images as per input in the left-most column, the set of calculated cluster prototypes in the middle and for reference all cluster members of the selected prototype on the right (Figure 66). A text input field allows a code point (or generic string if desired) to be entered and assigned to the prototype. In extension that assigns the code to all cluster members.

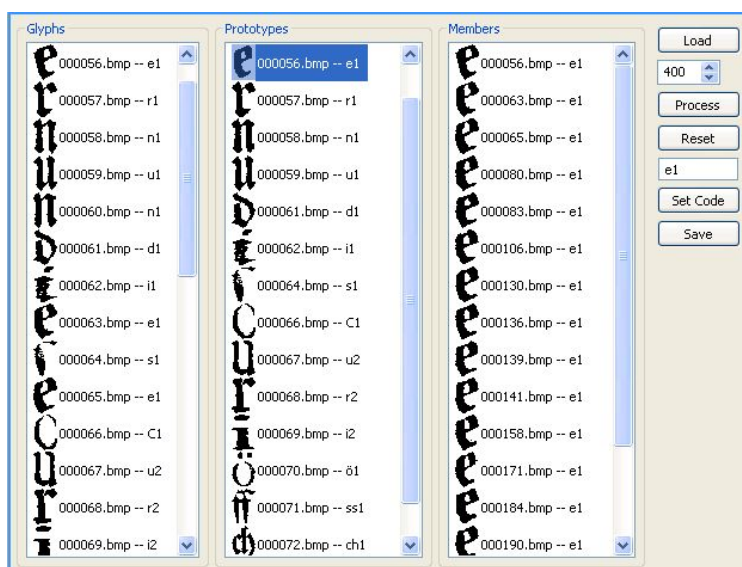


Figure 66: Glyph Labelling Interface [116]

This can then feed into an overall structure in PAGE format (layout information such as region outlines etc. again to be obtained from appropriate external page segmentation methods).

The main benefit of the adaptive glyph clustering step lies in the reduction of the labelling effort. Instead of assigning a code point individually to every single glyph as shown in the left column, only the smaller number of cluster prototypes needs to be labelled.

7.8.2 Faithful Reproduction of Document Images

The presented adaptive glyph clustering approach can be used to derive a document specific alphabet and font which in turn can be utilised in scenarios where content is only to be displayed. Such a document representation can be considered Unicode encoded but the actual meaning of characters is not known (with Private Use Area code points being used).

Figure 67 shows an example following earlier work related to electronic publishing of digitised documents [116], [68].

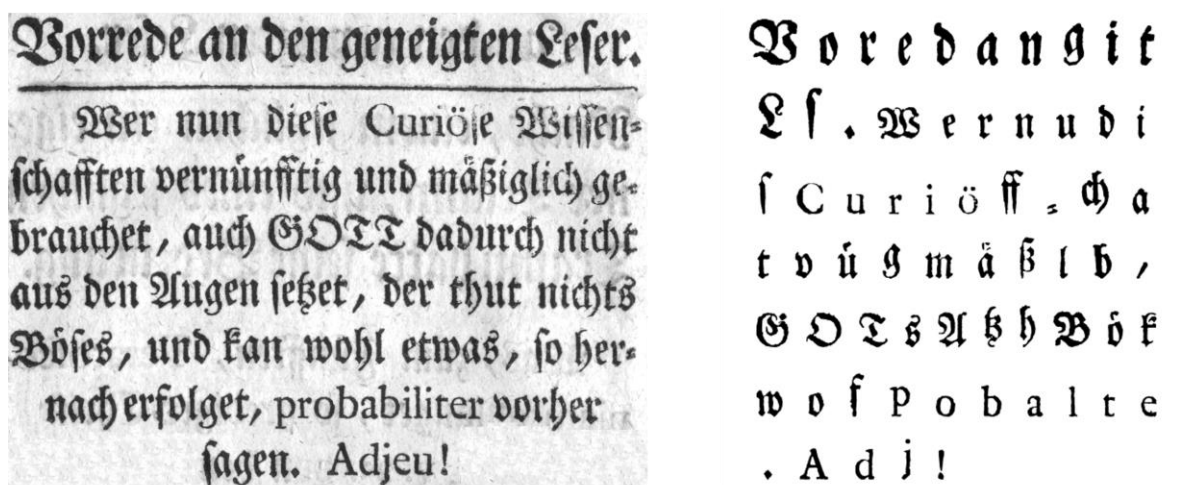


Figure 67: Original Document Image (left) and Prototypes of the Document Specific Alphabet (right) [116]

The prototype images can then be vectorised in order to create a document specific font e.g. using Scalable Vector Graphics (SVG). The general approach as well as the achievable visual quality (Figure 68) related to the applied level of abstraction have been studied in [68].

h	h	h	h	h	h	h	h	h	h	h	h	h
k	k	k	k	k	k	k	k	k	k	k	k	k
d	d	d	d	d	d	d	d	d	d	d	d	d
0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	1,1	1,2

vectorization parameter ϵ

Figure 68: Visual Quality of Vectorised Glyphs for Increasing Abstraction Level [68]

The document specific alphabet and font can then be used to generate an encoded document that behaves in very much the same way as normal text even though the actual meaning of characters is unknown. A rendering of such a document is shown in Figure 69.

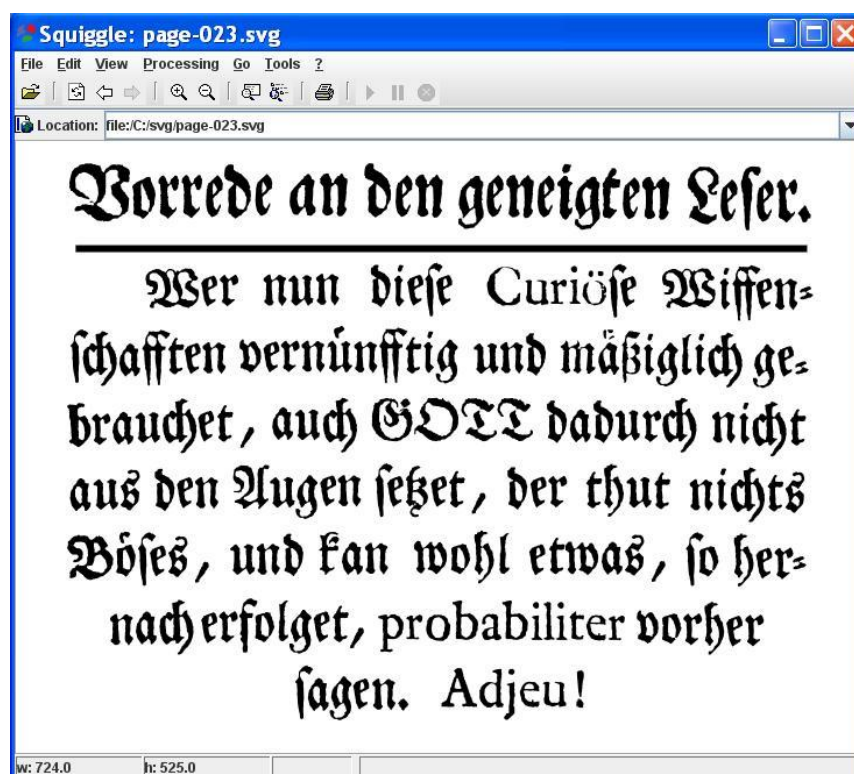


Figure 69: Rendered Output (Based on a Document Specific Alphabet and Font)

As a major advantage, rendered documents can then be scaled without the typical quality loss of bitmap images and/or “reflowed” i.e. presented in different layouts via output transformations such as for bigger and smaller screens. A further advantage is the dramatic reduction in required storage space due to symbolic compression from clustering

as well as further compression through raster to vector conversion (vectorisation) of glyph images used in the font.

This application is even robust against mistakes made during clustering (which are in other scenarios to be avoided at all costs) as wrong groupings will still involve similar glyphs which humans tend to interpret correctly from the given context. If a “O” is mistakenly displayed as “0” then this will go largely undetected to the human eye. In fact, various types of documents have this as an inherent issue. It is for instance common in typewritten material that the same type is used to produce characters with different meaning. Also, actual character codes for identical glyph images may entirely depend on context such as the dual use of letters in roman numbers.

As possible extension to a simple system for displaying document specific alphabet and font encoded material, text search can be implemented via a custom user interface supporting the underlying Private Use Area code points (e.g. in the form of a virtual keyboard).

Finally, the here omitted glyph labelling step can still be undertaken at a later time, leading to fully encoded documents.

7.9 Summary

This chapter provided a systematic appraisal of the work done and the outcomes that were achieved. It confirmed not only the validity of the results but also that the presented work has had a major impact on the research and library landscape, in particular related to the PAGE format. The usefulness of the proposed adaptive glyph clustering approach has been demonstrated through its applicability to real world use scenarios.

8 Conclusions and Future Work

This final chapter wraps up the achievements of the presented work. It summarises how a substantial contribution has been made to the field of document analysis and how new knowledge has been generated. A brief discussion of future work concludes the thesis.

8.1 Conclusions

This work presents original contributions to the general problem of encoding digitised documents. On a higher level it describes a format specification that allows capturing of detailed information in the context of document image analysis and OCR. On a more specific level it addresses challenges associated with encoding of unstandardised characters.

In particular, the PAGE (pagecontent) format specification has seen extensive uptake and is today widely considered the de-facto standard for representing ground truth data related to layout analysis and text recognition. This is the result of long-term efforts to continuously improve the format specification but also grounded in the work at PRImA Research to provide a whole eco system with comprehensive tool support as well as datasets with critical mass. The success factors of PAGE have been discussed and evidence of its significant real-world impact has been presented. Especially its prominent placement in a four-star REF 2021 case study as well as the adoption of specific features by the Library of Congress maintained ALTO format can be highlighted to this end.

Adaptive glyph clustering as described in this work is a novel solution to the specific problem of encoding previously unknown and/or unstandardised characters. First and foremost, this is a theoretical work showing the general feasibility of the approach. Nonetheless, it has been possible to demonstrate how the application of adaptive glyph clustering can significantly reduce the manual effort normally required in application scenarios such as glyph labelling.

8.2 Future Work

While the PAGE (pagecontent) format has reached a very mature state it is envisaged that future work will see maintenance updates (such as keeping up with ISO standards for language names) as well new features. As in the past, this will be driven by emerging use cases. Possible updates at this stage are new region types (e.g. for forms), additional attributes and attribute values as well as further information related to text lines (such as representations for mean line, ascent line and descent line).

With regard to the proposed adaptive glyph clustering approach, there is potential for further optimisation in order to achieve compression rates closer to 100%. The problems of feature selection for the feature based distance measure and the tuning of weights for the combined distance measure could be studied further. In particular for the choice of weights it would be a very appealing research goal to find a method that is adaptive as well (i.e. not requiring user input or pre-set values). Further exploration and utilisation of background knowledge relating to the nature of the input data could provide leads in this direction. Finally, integration of the proposed method in readily available software tools (such as Aletheia) could lead to more positive impact of the work presented here.

References

- [1] "PRImA Datasets," [Online]. Available: <http://www.primaresearch.org/datasets>.
- [2] F. E. Froehlich and A. Kent, The Froehlich/Kent Encyclopedia of Telecommunications: Volume 3 - Codes for the Prevention of Errors to Communications Frequency Standards: v. 3, CRC Press, 1991.
- [3] H. F. Schantz, The history of OCR, optical character recognition, Recognition Technologies Users Association, 1982.
- [4] H. Bunke and P. S. P. Wang, Eds., Handbook of Character Recognition and Document Image Analysis, World Scientific, 1997.
- [5] H. S. Baird, H. Bunke and K. Yamamoto, Eds., Structured Document Image Analysis, Springer, 1992.
- [6] L. O'Gorman and R. Kasturi, Document Image Analysis, IEEE Computer Society Executive Briefings, 1997.
- [7] R. Kasturi, L. O'Gorman and V. Govindaraju, "Document image analysis: A primer," *Sadhana*, vol. 27, pp. 3-22, 2002.
- [8] G. Nagy, "Twenty years of Document Image Analysis in PAMI," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 38-62, 2000.
- [9] H. S. Baird, "Digital Libraries and Document Image Analysis," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR'03)*, Edinburgh, Scotland, UK, 2003.
- [10] S. Pletschacher, C. Clausner and A. Antonacopoulos, "Europeana Newspapers OCR Workflow Evaluation," in *Proceedings of the 2015 Workshop on Historical Document Imaging and Processing (HIP2015)*, Nancy, France, 2015.

- [11] "Europeana Newspapers | A Gateway to European Newspapers Online," 2015.
[Online]. Available: <http://www.europeana-newspapers.eu/>.
- [12] Adobe Systems Incorporated, "The Four Flavors of Adobe PDF for Paper-based Documents," Adobe Acrobat Capture - White Paper, 2000.
- [13] T. M. Breuel, "The Future of Document Imaging in the Era of Electronic Documents," in *Proceedings of the International Workshop on Document Analysis*, Kolkata, India, 2005.
- [14] T. M. Breuel, W. C. Janssen, K. Popat and H. S. Baird, "Paper to PDA," in *Proceedings of the 16th International Conference on Pattern Recognition*, 2002.
- [15] S. Marinai, A. Anzivino and M. Spampani, "Towards a Faithful Visualization of Historical Books on E-book Readers," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing (HIP'11)*, Beijing, China, 2011.
- [16] A. K. Jain and B. Yu, "Document Representation and Its Application to Page Decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 294-308, 1998.
- [17] S. Mao, A. Rosenfeld and T. Kanungo, "Document Structure Analysis Algorithms: A Literature Survey," in *Document Recognition and Retrieval X*, Santa Clara, CA, USA, 2003.
- [18] L. Likforman-Sulem, A. Zahour and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal of Document Analysis and Recognition (IJDAR)*, pp. 123-138, 2007.
- [19] M. Makridis, N. Nikolaou and B. Gatos, "An Efficient Word Segmentation Technique for Historical and Degraded Machine-Printed Documents," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Curitiba, Paraná, Brazil, 2007.

- [20] R. G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 690-706, 1996.
- [21] G. Nagy, S. Seth and M. Viswanathan, "A Prototype Document Image Analysis System for Technical Journals," *Computer*, vol. 25, pp. 10-22, 1992.
- [22] H. S. Baird, S. E. Jones and S. J. Fortune, "Image Segmentation by Shape-Directed covers," in *Proceedings of the 10th International Conference on Pattern Recognition*, Atlantic City, NJ, USA, 1990.
- [23] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1162-1173, 1993.
- [24] K. Kise, A. Sato and M. Iwata, "Segmentation of Page Images Using the Area Voronoi Diagram," *Computer Vision and Image Understanding*, vol. 70, p. 370-382, 1998.
- [25] F. M. Wahl, K. Y. Wong and R. G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents," *Computer Graphics and Image Processing*, vol. 20, p. 375-390, 1982.
- [26] T. Pavlidis and J. Zhou, "Page Segmentation and Classification," *Graphical Models and Image Processing*, vol. 54, p. 484-496, 1992.
- [27] A. Antonacopoulos, "Page Segmentation Using the Description of the Background," *Computer Vision and Image Understanding*, pp. 350-369, 1998.
- [28] C. Strouthopoulos and N. Papamarkos, "Text identification for document image analysis using a neural network," *Image and Vision Computing*, vol. 16, p. 879-896, 1998.
- [29] A. Antonacopoulos, D. Bridson, C. Papadopoulos and S. Pletschacher, "ICDAR2009 Page Segmentation Competition," in *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR2009)*, Barcelona, Spain,

2009.

- [30] A. Antonacopoulos, C. Clausner, C. Papadopoulos and S. Pletschacher, "Historical Document Layout Analysis Competition," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011)*, Beijing, China, 2011.
- [31] A. Antonacopoulos, C. Clausner, C. Papadopoulos and S. Pletschacher, "ICDAR2013 Competition on Historical Newspaper Layout Analysis – HNLA2013," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR2013)*, Washington DC, USA, 2013.
- [32] A. Antonacopoulos, C. Clausner, C. Papadopoulos and S. Pletschacher, "ICDAR2015 Competition on Recognition of Documents with Complex Layouts – RDCL2015," in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR2015)*, Nancy, France, 2015.
- [33] "ALTO - Analyzed Layout and Text Object," 2015. [Online]. Available: <http://www.loc.gov/standards/alto/>.
- [34] T. M. Breuel, "The hOCR Microformat for OCR Workflow and Results," in *Proceedings Ninth International Conference on Document Analysis and Recognition (ICDAR2007)*, Curitiba, Paraná, Brazil, 2007.
- [35] "Text Encoding Initiative," [Online]. Available: <https://tei-c.org/>.
- [36] A. Constantin, S. Peroni, S. Pettifer, D. Shotton and F. Vitali, "The document components ontology (DoCO)," *Semantic Web*, vol. 7, no. 2, pp. 167-181, 2016.
- [37] "DJVU at Fileformat.com," [Online]. Available: <https://docs.fileformat.com/image/djvu/>.
- [38] "DocLayNet - COCO," [Online]. Available: <https://github.com/DS4SD/DocLayNet#data-format-details>.

- [39] W3C, "Scalable Vector Graphics (SVG)," 2011. [Online]. Available: <http://www.w3.org/Graphics/SVG/>.
- [40] R. N. Ascher and G. Nagy, "A Means for Achieving a High Degree of Compaction on Scan-Digitized Printed Text," *IEEE Transactions on Computers*, Vols. C-23, pp. 1174 - 1179, 1974.
- [41] A. K. Jain, M. N. Murty and P. J. Flynn, "Data Clustering: A Review," *Computing Surveys (CSUR)*, vol. 31, pp. 264-323, 1999.
- [42] O. Maimon and L. Rokach, Eds., *Data Mining and Knowledge Discovery Handbook*, Springer, 2010.
- [43] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, p. 651–666, 2010.
- [44] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, pp. 4-29, 1984.
- [45] J. E. Fowler, "A Survey of Adaptive Vector Quantization - Part I: A Unifying Structure," IPS Lab. Tech. Rep. TR-97-01, 1997.
- [46] K. Tombre, C. Ah-Soon, P. Dosch, G. Masini and S. Tabbone, "Stable and Robust Vectorization: How to Make the Right Choices," in *Graphics Recognition - Recent Advances*, A. K. Chhabra and D. Dori, Eds., Berlin Heidelberg, Springer, 2000, pp. 3-18.
- [47] S. Pletschacher, "Representation of Digitized Documents Using Document Specific Alphabets and Fonts," in *Proceedings of the 5th IS&T Archiving Conference*, Bern, Switzerland, 2008.
- [48] V. Kluzner, A. Tzadok and M. Stanich, "Variational-Methods-Based Glyph Vectorization," in *TAGA - Technical Association of the Graphic Arts*, San Francisco, CA, USA, 2008.
- [49] H. S. M. Al-Khaffaf, F. Shafait, M. P. Cutter and T. M. Breuel, "On the Performance of

Decapod's Digital Font Reconstruction," in *Proceedings 21st International Conference on Pattern Recognition (ICPR 2012)*, Tsukuba, Japan, 2012.

- [50] "AutoTrace," 2004. [Online]. Available: <http://autotrace.sourceforge.net/>.
- [51] "Potrace," 2015. [Online]. Available: <http://potrace.sourceforge.net/>.
- [52] A. Antonacopoulos, D. Karatzas and D. Bridson, "Ground Truth for Layout Analysis Performance Evaluation," in *Proceedings of the 7th International Association for Pattern Recognition (IAPR) Workshop on Document Analysis Systems (DAS2006)*, Nelson, New Zealand, 2006.
- [53] H. Balk and L. Ploeger, "IMPACT: working together to address the challenges involving mass digitization of historical printed text.," *OCLC Systems & Services: International digital library perspectives*, vol. 25, no. 4, pp. 233-248, 2009.
- [54] "SUCCEED," 2015. [Online]. Available: <https://www.primaresearch.org/projects/SUCCEED>.
- [55] W3C, "Extensible Markup Language (XML)," 2008. [Online]. Available: <http://www.w3.org/XML/>.
- [56] S. Pletschacher and A. Antonacopoulos, "The PAGE (Page Analysis and Ground-Truth Elements) Format Framework," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR2010)*, Istanbul, Turkey, 2010.
- [57] "PRImA Tools," [Online]. Available: <http://www.primaresearch.org/tools>.
- [58] "Census 1961," [Online]. Available: <http://primaresearch.org/projects/Census1961>.
- [59] Library of Congress, "Codes for the Representation of Names of Languages," 2010. [Online]. Available: https://www.loc.gov/standards/iso639-2/php/code_list.php.
- [60] Unicode Consortium, "Codes for the representation of names of scripts," [Online]. Available: <http://unicode.org/iso15924/iso15924-codes.html>. [Accessed 14 07

2016].

- [61] J. Biggs, "Comparison of Visual and Logical Character Segmentation in Tesseract OCR Language Data for Indic Writing Scripts.," in *Proceedings of the Australasian Language Technology Association Workshop 2015*, Parramatta, Australia, 2015.
- [62] C. Clausner and A. Antonacopoulos, "Ontology and Framework for Semantic Labelling of Document Data and Software Methods," in *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS2018)*, Vienna, Austria, 2018.
- [63] C. Clausner, J. Hayes, A. Antonacopoulos and S. Pletschacher, "Creating a Complete Workflow for Digitising Historical Census Documents: Considerations and Evaluation," in *Proceedings of the 2017 Workshop on Historical Document Imaging and Processing (HIP2017)*, Kyoto, Japan, 2017.
- [64] C. Clausner, Extraktion dokumentspezifischer Alphabete aus digitalisierten Vorlagen, Chemnitz: Chemnitz University of Technology, 2006.
- [65] D. Salomon, Data Compression, 3rd Edition ed., New York: Springer-Verlag, 2004.
- [66] "MUFI: The Medieval Unicode Font Initiative," [Online]. Available: <https://mufi.info/>.
- [67] S. Pletschacher, "OCR Alternatives for Electronic Publishing of Digitised Documents," in *From Author to Reader: Challenges for the Digital Content Chain, Proceedings of the 9th ICCG International Conference on Electronic Publishing*, Leuven-Heverlee, Belgium, 2005.
- [68] S. Pletschacher, M. Eckert and A. C. Huebler, "Vectorization of Glyphs and Their Representation in SVG for XML-Based Processing," in *Digital Spectrum: Integrating Technology and Culture, Proceedings of the 10th International Conference on Electronic Publishing*, Bansko, Bulgaria, 2006.

- [69] C. Clausner, S. Pletschacher and A. Antonacopoulos, "Aletheia - An Advanced Document Layout and Text Ground-Truthing System for Production Environments," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011)*, Beijing, China, 2011.
- [70] A. -. D. A. System. [Online]. Available: <http://primaresearch.org/tools/Aletheia>.
- [71] "WebAletheia," [Online]. Available: <http://primaresearch.org/tools/WebAletheia>.
- [72] "PRImA Performance Evaluation," [Online]. Available: <http://primaresearch.org/tools/PerformanceEvaluation>.
- [73] C. Clausner, S. Pletschacher and A. Antonacopoulos, "Scenario Driven In-Depth Performance Evaluation of Document Layout Analysis Methods," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011)*, Beijing, China, 2011.
- [74] "PAGE Libraries," [Online]. Available: <http://primaresearch.org/tools/PAGELibraries>.
- [75] "PAGE Converter and Validator," [Online]. Available: <http://primaresearch.org/tools/PAGEConverterValidator>.
- [76] "PAGE Metadata Scanner," [Online]. Available: <http://primaresearch.org/tools/PAGEMetadataScanner>.
- [77] "Tesseract OCR to PAGE," [Online]. Available: <http://primaresearch.org/tools/TesseractOCRToPAGE>.
- [78] "PAGE Extractor/Exporter," [Online]. Available: <http://primaresearch.org/tools/ExtractorExporter>.
- [79] "PAGE Viewer," [Online]. Available: <http://primaresearch.org/tools/PAGEViewer>.
- [80] C. Papadopoulos, S. Pletschacher, C. Clausner and A. Antonacopoulos, "The IMPACT Dataset of Historical Document Images," in *Proceedings of the 2013 Workshop on*

Historical Document Imaging and Processing (HIP2013), Washington DC, USA, 2013.

- [81] C. Clausner, C. Papadopoulos, S. Pletschacher and A. Antonacopoulos, "The ENP Image and Ground Truth Dataset of Historical Newspapers," in *Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR2015)*, Nancy, France, 2015.
- [82] A. Antonacopoulos, D. Bridson, C. Papadopoulos and S. Pletschacher, "A Realistic Dataset for Performance Evaluation of Document Layout Analysis," in *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR2009)*, Barcelona, Spain, 2009.
- [83] "PRImA Competitions," [Online]. Available: <http://primaresearch.org/competitions>.
- [84] A. Antonacopoulos, C. Clausner, C. Papadopoulos and S. Pletschacher, "ICDAR2013 Competition on Historical Book Recognition – HBR2013," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR2013)*, Washington DC, USA, 2013.
- [85] C. Clausner, A. Antonacopoulos, T. Derrick and S. Pletschacher, "ICDAR2017 Competition on Recognition of Early Indian Printed Documents – REID2017," in *Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR2017)*, Kyoto, Japan, 2017.
- [86] C. Clausner, A. Antonacopoulos, T. Derrick and S. Pletschacher, "ICDAR2019 Competition on Recognition of Early Indian Printed Documents – REID2019," in *Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR2019)*, Sydney, Australia, 2019.
- [87] C. Clausner, A. Antonacopoulos, N. McGregor and D. Wilson-Nunn, "ICFHR 2018 Competition on Recognition of Historical Arabic Scientific Manuscripts – RASM2018," in *Proceedings of the 17th International Workshop on Frontiers in Handwriting Recognition (ICFHR2018)*, Niagara Falls, USA, 2018.

- [88] C. Clausner, A. Antonacopoulos and S. Pletschacher, "ICDAR2017 Competition on Recognition of Documents with Complex Layouts – RDCL2017," in *Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR2017)*, Kyoto, Japan, 2017.
- [89] C. Clausner, A. Antonacopoulos and S. Pletschacher, "ICDAR2019 Competition on Recognition of Documents with Complex Layouts – RDCL2019," in *Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR2019)*, Sydney, Australia, 2019.
- [90] "Google Scholar," [Online]. Available: https://scholar.google.co.uk/citations?view_op=view_citation&hl=en&user=6zSOWIIAAAAJ&citation_for_view=6zSOWIIAAAAJ:d1gkVwhDpl0C.
- [91] K. Nikolaidou, M. Seuret, H. Mokayed and M. Liwicki, "A survey of historical document image datasets," *IJDAR*, vol. 25, p. 305–338, 2022.
- [92] E. Valveny, "Datasets and Annotations for Document Analysis and Recognition," in *Handbook of Document Image Processing and Recognition*, London, Springer, 2014.
- [93] C. Neudecker, K. Baierer, M. Federbusch, M. Boenig, K. Würzner, V. Hartmann and E. Herrmann, "OCR-D: An end-to-end open source OCR framework for historical printed documents," in *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage (DATECH2019)*, Brussels, Belgium, 2019.
- [94] J. Baloun, P. Král and L. Lenc, "ChronSeg: Novel Dataset for Segmentation of Handwritten Historical Chronicles," in *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, 2021.
- [95] M. Diem, F. Kleber, S. Fiel, T. Grüning and B. Gatos, "cBAD: ICDAR2017 Competition on Baseline Detection," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, Japan, 2017.
- [96] M. Diem, F. Kleber, R. Sablatnig and B. Gatos, "cBAD: ICDAR2019 Competition on

Baseline Detection,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, NSW, Australia, 2019.

- [97] T. Grüning, R. Labahn, M. Diem, F. Kleber and S. Fiel, “READ-BAD: A New Dataset and Evaluation Scheme for Baseline Detection in Archival Documents,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Vienna, Austria, 2018.
- [98] J. A. Sánchez, V. Romero, A. H. Toselli and E. Vidal, “ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset,” in *016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Shenzhen, China, 2016.
- [99] S. Colutto, P. Kahle, H. Guenter and G. Muehlberger, “Transkribus. A Platform for Automated Text Recognition and Searching of Historical Documents,” in *2019 15th International Conference on eScience (eScience)*, San Diego, CA, USA, 2019.
- [100] B. Gatos, G. Louloudis, T. Causer, K. Grint, V. Romero, J. A. Sánchez;, A. H. Toselli; and E. Vidal, “Ground-Truth Production in the Transcriptorium Project,” in *2014 11th IAPR International Workshop on Document Analysis Systems*, Tours, France, 2014.
- [101] J. A. Sánchez, V. Romero, A. H. Toselli and E. Vidal, “ICFHR2014 Competition on Handwritten Text Recognition on Transcriptorium Datasets (HTRtS),” in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Hersonissos, Greece, 2014.
- [102] J. A. Sánchez, A. H. Toselli, V. Romero and E. Vidal, “ICDAR 2015 competition HTRtS: Handwritten Text Recognition on the tranScriptorium dataset,” in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, 2015.
- [103] F. Simistira, M. Bouillon, M. Seuret, M. Würsch, M. Alberti, R. Ingold and M. Liwicki, “ICDAR2017 Competition on Layout Analysis for Challenging Medieval Manuscripts,” in *2017 14th IAPR International Conference on Document Analysis and Recognition*

(ICDAR), Kyoto, Japan, 2017.

- [104] P. Stokes, B. Kiessling, B. E. Stökl, R. Tissot and E. H. Gargem, “The EScriptorium VRE for Manuscript Cultures,” *Classics@ Journal, Ancient Manuscripts and Virtual Research Environments*, vol. 18, 2021.
- [105] Z. Shen, R. Zhang, M. Dell, B. Lee, J. Carlson and W. Li, “LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis,” in *Document Analysis and Recognition – ICDAR 2021*, 2021.
- [106] L. Lenc, J. Martínek, P. Král, A. Nicolao and V. Christlein, “HDPa: historical document processing and analysis framework,” *Evolving Systems*, vol. 12, pp. 177-190, 2021.
- [107] S. Klut, R. van Koert and R. Sluijter, “Laypa: A Novel Framework for Applying Segmentation Networks to Historical Documents,” in *Proceedings of the 7th International Workshop on Historical Document Imaging and Processing (HIP '23)*, San Jose, CA, USA, 2023.
- [108] C. Reul, D. Christ, A. Hartelt, N. Balbach, M. Wehner, U. Springmann, C. Wick, C. Grundig, A. Büttner and F. Puppe, “OCR4all—An Open-Source Tool Providing a (Semi-)Automatic OCR Workflow for Historical Printings,” *Applied Sciences*, vol. 9, no. 22, 2019.
- [109] B. Kurar Barakat, R. Cohen and J. El-Sana, “VML-MOC: Segmenting a Multiply Oriented and Curved Handwritten Text Line Dataset,” in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, Sydney, NSW, Australia, 2019.
- [110] A. Kölsch, A. Mishra, S. Varshneya, M. Z. Afzal and M. Liwicki, “Recognizing Challenging Handwritten Annotations with Fully Convolutional Networks,” in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, 2018.
- [111] “ALTO 4.3,” [Online]. Available: <https://www.loc.gov/standards/alto/v4/alto-4->

3.xsd.

- [112] “REF2021 Impact Case Study - Enabling digital transformation through effective digitisation,” [Online]. Available: <https://results2021.ref.ac.uk/impact/3eac07b1-7c69-4191-a622-85f64476c018?page=1>.
- [113] UKRI, “Guidance on REF 2021 results,” [Online]. Available: <https://ref.ac.uk/guidance-on-results/guidance-on-ref-2021-results/>.
- [114] G. Vamvakas, N. Stamatopoulos, B. Gatos and S. J. Perantonis, “Automatic unsupervised parameter selection for character segmentation,” in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS '10)*, Boston, USA, 2010.
- [115] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos and N. Papamarkos, “Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths,” *Image and Vision Computing*, vol. 28, no. 4, pp. 590-604, 2010.
- [116] S. Pletschacher, “A Self-Adaptive Method for Extraction of Document-Specific Alphabets,” in *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR2009)*, Barcelona, Spain, 2009.

Appendix

Pagecontent Format Specification – First Release

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2009-03-
16"
    elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2009-03-
16">

    <element name="PcGts" type="pc:PcGtsType">
        <annotation>
            <documentation>Page Content - Ground Truth and
Storage</documentation>
        </annotation></element>

    <complexType name="PcGtsType">
        <sequence>
            <element name="Metadata" type="pc:MetadataType"></element>
            <element name="Page" type="pc:PageType"></element>
        </sequence>
        <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
        <sequence>
            <element name="Creator" type="string"></element>
            <element name="Created" type="dateTime"></element>
            <element name="LastChange" type="dateTime"></element>
            <element name="Comments" type="string" minOccurs="0"
                maxOccurs="1"></element>
        </sequence>
    </complexType>

    <complexType name="PageType">
        <sequence>
            <element name="Border" type="pc:BorderType" minOccurs="0"
                maxOccurs="1">
            </element>
            <element name="PrintSpace" type="pc:PrintSpaceType"
                minOccurs="0" maxOccurs="1">
            </element>
            <element name="ReadingOrder" type="pc:ReadingOrderType"
                minOccurs="0" maxOccurs="1">
            </element>
        <annotation>
            <documentation>If the reading order element exists, all
regions have to be covered (i.e. all region ids must be mentioned exactly
once)!</documentation>
        </annotation>
    </element>
```

```

        <element name="Layers" type="pc:LayersType" minOccurs="0"
maxOccurs="1">
        <annotation>
            <documentation>If the layers element exists, all regions
have to be covered (i.e. all region ids must be mentioned exactly
once)!</documentation>
        </annotation></element>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="TextRegion"
type="pc:TextRegionType"></element>
            <element name="ImageRegion"
                type="pc:ImageRegionType">
            </element>
            <element name="LineDrawingRegion"
                type="pc:LineDrawingRegionType">
            </element>
            <element name="GraphicRegion"
                type="pc:GraphicRegionType">
            </element>
            <element name="TableRegion"
                type="pc:TableRegionType">
            </element>
            <element name="ChartRegion"
                type="pc:ChartRegionType">
            </element>
            <element name="SeparatorRegion"
                type="pc:SeparatorRegionType">
            </element>
            <element name="MathsRegion"
                type="pc:MathsRegionType">
            </element>
            <element name="NoiseRegion"
                type="pc:NoiseRegionType">
            </element>
            <element name="FrameRegion"
                type="pc:FrameRegionType">
            </element>
            <element name="UnknownRegion"
                type="pc:UnknownRegionType">
            </element>
        </choice>
    </sequence>
    <attribute name="imageFilename" type="string"
use="required"></attribute>
    <attribute name="imageWidth" type="int" use="required"></attribute>
    <attribute name="imageHeight" type="int"
use="required"></attribute>
</complexType>
<complexType name="TextRegionType">
    <annotation>
        <documentation>Pure text is represented as a text region. This
includes drop capitals, but practically ornate text may be considered as a
graphic.</documentation></annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"

```

```

        minOccurs="0" maxOccurs="1">
    </element>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="orientation" type="float" use="optional">
    <annotation>
        <documentation>Individual skew of the region in
degrees (Range: -89.999,90)</documentation>
    </annotation></attribute>
<attribute name="type" type="pc:TextTypeSimpleType" use="optional">
    <annotation>
        <documentation>The nature of the text in the region</documentation>
    </annotation>
</attribute>
<attribute name="textColour" type="pc:ColourSimpleType" use="optional">
    <annotation>
        <documentation>The text colour of the
region</documentation></annotation>
    </attribute>
<attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
    <annotation>
        <documentation>The background colour of the
region</documentation></annotation>
    </attribute>
<attribute name="reverseVideo" type="boolean" use="optional">
    <annotation>
        <documentation>Specifies whether the colour of the text appears
reversed against a background colour</documentation></annotation>
    </attribute>
<attribute name="fontSize" type="float" use="optional">
    <annotation>
        <documentation>The size of the characters in
points</documentation></annotation></attribute>
<attribute name="Leading" type="int" use="optional">
    <annotation>
        <documentation>The degree of space in points between
the lines of text</documentation>
    </annotation></attribute>
<attribute name="kerning" type="int" use="optional">
    <annotation>
        <documentation>The degree of space in points between
the characters in a string of text</documentation>
    </annotation></attribute>
<attribute name="readingDirection" type="pc:ReadingDirectionSimpleType"
use="optional">
    <annotation>
        <documentation>The direction in which text in a region should be
read (within lines)</documentation>
    </annotation>
</attribute>
<attribute name="readingOrientation" type="float" use="optional">
    <annotation>
        <documentation>The degrees by which you need to turn
your head in order to read the text when it is placed on the horizontal (Range:
-89.999,90)</documentation>
    </annotation></attribute>
<attribute name="indented" type="boolean" use="optional">
    <annotation>

```

```

        <documentation>Defines whether a region of text is indented or
not</documentation></annotation>
        </attribute>
        <attribute name="primaryLanguage" type="pc:LanguageSimpleType"
use="optional">
        <annotation>
        <documentation>The primary language used in the
region</documentation></annotation>
        </attribute>
        <attribute name="secondaryLanguage" type="pc:LanguageSimpleType"
use="optional">
        <annotation>
        <documentation>The secondary language used in the
region</documentation>
        </annotation>
        </attribute>
        <attribute name="primaryScript" type="pc:ScriptSimpleType"
use="optional">
        <annotation>
        <documentation>The primary script used in the
region</documentation>
        </annotation>
        </attribute>
        <attribute name="secondaryScript" type="pc:ScriptSimpleType"
use="optional">
        <annotation>
        <documentation>The secondary script used in the
region</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="CoordsType">
        <sequence>
            <element name="Point" type="pc:PointType" minOccurs="0"
maxOccurs="unbounded"></element>
        </sequence>
    </complexType>
    <complexType name="PointType">
        <attribute name="x" type="int" use="required"></attribute>
        <attribute name="y" type="int" use="required"></attribute>
    </complexType>
    <complexType name="TextLineType">
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Word" type="pc:WordType" minOccurs="0"
maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"
minOccurs="0" maxOccurs="1">
            </element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="WordType">
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Glyph" type="pc:GlyphType" minOccurs="0"
maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"

```

```

        minOccurs="0" maxOccurs="1">
    </element>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
</complexType>
<complexType name="GlyphType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="1">
            </element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="ligature" use="optional" type="boolean">
    </attribute>
    <attribute name="symbol" use="optional" type="boolean">
    </attribute>
</complexType>
<complexType name="TextEquivType">
    <sequence>
        <element name="PlainText" type="string">
            <annotation>
                <documentation>Text in a "simple" form (ASCII
or extended ASCII as mostly used for typing).
I.e. no use of special characters for ligatures (should be stored as two
separate characters)
etc.</documentation>
            </annotation></element>
        <element name="Unicode" type="string">
            <annotation>
                <documentation>Correct encoding of the
original, always using the corresponding Unicode code point.
I.e. ligatures have to be represented as one character
etc.</documentation>
            </annotation></element>
    </sequence>
</complexType>
<complexType name="ImageRegionType">
    <annotation>
        <documentation>An image is considered to be more intricate and
complex than a graphic. These can be photos or
drawings.</documentation></annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
        <annotation>
            <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
        </annotation></attribute>
    <attribute name="colourDepth" type="pc:ColourDepthSimpleType"
use="optional">
        <annotation>
            <documentation>The colour bit depth required for the
region</documentation>
        </annotation>
    </attribute>

```

```

    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
      <annotation>
        <documentation>The background colour of the region</documentation>
      </annotation>
    </attribute>
    <attribute name="embText" type="boolean" use="optional">
      <annotation>
        <documentation>Specifies whether the region also contains
text</documentation></annotation>
      </attribute>
    </complexType>
    <complexType name="LineDrawingRegionType">
      <annotation>
        <documentation>A line drawing is a single colour illustration
without solid areas.</documentation>
      </annotation>
      <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
      </sequence>
      <attribute name="id" type="ID" use="required"></attribute>
      <attribute name="orientation" type="float" use="optional">
        <annotation>
          <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
        </annotation></attribute>
      <attribute name="penColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
          <documentation>The pen (foreground) colour of the
region</documentation></annotation>
        </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
          <documentation>The background colour of the
region</documentation></annotation>
        </attribute>
      <attribute name="embText" type="boolean" use="optional">
        <annotation>
          <documentation>Specifies whether the region also contains
text</documentation>
        </annotation>
      </attribute>
    </complexType>
    <complexType name="GraphicRegionType">
      <annotation>
        <documentation>Regions containing simple graphics, such as a
company logo, should be marked as graphic regions.</documentation>
      </annotation>
      <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
      </sequence>
      <attribute name="id" type="ID" use="required"></attribute>
      <attribute name="orientation" type="float" use="optional">
        <annotation>
          <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90).</documentation>
        </annotation></attribute>
      <attribute name="type" use="optional" type="pc:GraphicsTypeSimpleType">

```



```

        <annotation>
            <documentation>The type of graphic in the
region</documentation></annotation>
        </attribute>
        <attribute name="numColours" type="int" use="optional">
            <annotation>
                <documentation>An approximation of the number of
colours used in the region</documentation>
            </annotation></attribute>
        <attribute name="embText" type="boolean" use="optional">
            <annotation>
                <documentation>Specifies whether the region also contains
text.</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="TableRegionType">
        <annotation>
            <documentation>Tabular data in any form is represented with a table
region. Rows and columns may or may not have separator lines; these lines are
not separator regions.</documentation></annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="orientation" type="float" use="optional">
            <annotation>
                <documentation>The orientation in degrees of the
baseline of the region (Range: -89.999,90).</documentation>
            </annotation></attribute>
        <attribute name="rows" type="int" use="optional">
            <annotation>
                <documentation>The number of rows present in the
table</documentation>
            </annotation></attribute>
        <attribute name="columns" type="int" use="optional">
            <annotation>
                <documentation>The number of columns present in the
table</documentation>
            </annotation></attribute>
        <attribute name="lineColour" type="pc:ColourSimpleType" use="optional">
            <annotation>
                <documentation>The colour of the lines used in the
region</documentation>
            </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
            <annotation>
                <documentation>The background colour of the region</documentation>
            </annotation>
        </attribute>
        <attribute name="lineSeparators" type="boolean" use="optional">
            <annotation>
                <documentation>Specifies the presence of line
separators</documentation>
            </annotation>
        </attribute>
        <attribute name="embText" type="boolean" use="optional">
            <annotation>

```

```

        <documentation>Specifies whether the region also contains
text</documentation>
        </annotation>
        </attribute>
    </complexType>
    <complexType name="ChartRegionType">
        <annotation>
            <documentation>Regions containing charts or graphs of any type,
should be marked as chart regions.</documentation></annotation>
            <sequence>
                <element name="Coords" type="pc:CoordsType"></element>
            </sequence>
            <attribute name="id" type="ID" use="required"></attribute>
            <attribute name="orientation" type="float" use="optional">
                <annotation>
                    <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
                </annotation></attribute>
            <attribute name="type" use="optional" type="pc:ChartTypeSimpleType">
                <annotation>
                    <documentation>The type of chart in the
region</documentation></annotation>
                </attribute>
            <attribute name="numColours" type="int" use="optional">
                <annotation>
                    <documentation>An approximation of the number of
colours used in the region</documentation></annotation></attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
                <annotation>
                    <documentation>The background colour of the region</documentation>
                </annotation>
                </attribute>
            <attribute name="embText" type="boolean" use="optional">
                <annotation>
                    <documentation>Specifies whether the region also contains
text</documentation></annotation>
                </attribute>
        </complexType>
    <complexType name="SeparatorRegionType">
        <annotation>
            <documentation>Separators are lines that lie between columns and
paragraphs and can be used to logically separate different articles from each
other.</documentation></annotation>
            <sequence>
                <element name="Coords" type="pc:CoordsType"></element>
            </sequence>
            <attribute name="id" type="ID" use="required"></attribute>
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The orientation in degrees of the
region (Range: -89.999,90)</documentation>
                </annotation></attribute>
            <attribute name="colour" type="pc:ColourSimpleType" use="optional">
                <annotation>
                    <documentation>The colour of the separator</documentation>
                </annotation>
                </attribute>

```

```

</complexType>
<complexType name="MathsRegionType">
  <annotation>
    <documentation>Regions containing equations and mathematical
symbols should be marked as maths regions.</documentation></annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
      <annotation>
        <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
      </annotation></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
      <annotation>
        <documentation>The background colour of the
region</documentation></annotation>
      </attribute>
    </complexType>
  <complexType name="NoiseRegionType">
    <annotation>
      <documentation>Noise regions are regions where no real data lies,
only false data created by artifacts on the document or scanner
noise.</documentation>
    </annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
  </complexType>
  <complexType name="UnknownRegionType">
    <annotation>
      <documentation>To be used if the region type cannot be
ascertained.</documentation>
    </annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
  </complexType>
  <complexType name="FrameRegionType">
    <annotation>
      <documentation>
        A region that surrounds other regions (e.g. a box with
        blue background containing text regions)
      </documentation>
    </annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
      <choice minOccurs="0" maxOccurs="unbounded">
        <element name="TextRegion" type="pc:TextRegionType">
          </element>
        <element name="ImageRegion"
          type="pc:ImageRegionType">
          </element>
        <element name="LineDrawingRegion"
          type="pc:LineDrawingRegionType">
          </element>
      </choice>
    </sequence>
  </complexType>

```

```

        </element>
        <element name="GraphicRegion"
            type="pc:GraphicRegionType">
        </element>
        <element name="TableRegion"
            type="pc:TableRegionType">
        </element>
        <element name="ChartRegion"
            type="pc:ChartRegionType">
        </element>
        <element name="SeparatorRegion"
            type="pc:SeparatorRegionType">
        </element>
        <element name="MathsRegion"
            type="pc:MathsRegionType">
        </element>
        <element name="NoiseRegion"
            type="pc:NoiseRegionType">
        </element>
        <element name="FrameRegion"
            type="pc:FrameRegionType">
        </element>
        <element name="UnknownRegion"
type="pc:UnknownRegionType"></element>
    </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
    </attribute>
    <attribute name="borderPresent" type="boolean"
use="optional"></attribute>
</complexType>

<complexType name="PrintSpaceType">
    <annotation>
        <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.
It does not contain pagenumber (if not part of running title), marginals,
signature mark, preview words.
</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>

<complexType name="ReadingOrderType">
    <annotation>
        <documentation>Definition of the reading order within the page.
Without further grouping, regions are supposed to be unordered on the page.
To express a reading order between elements they have to be included in a
ordered group. Groups may contain further groups.</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>

```

```

        <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
    </choice>
</complexType>

<complexType name="RegionRefIndexedType">
    <annotation>
        <documentation>Numbered region</documentation>
    </annotation>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
        </annotation>
    </attribute>
    <attribute name="regionRef" type="IDREF" use="required"></attribute>
</complexType>

<complexType name="OrderedGroupIndexedType">
    <annotation>
        <documentation>Indexed group containing ordered
elements</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"></element>
        <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"></element>
        <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"></element>
    </choice>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
        </annotation>
    </attribute>
</complexType>

<complexType name="UnorderedGroupIndexedType">
    <annotation>
        <documentation>Indexed group containing unordered
elements</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
        <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
    </choice>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>
                Position (order number) of this item within the
                current hierarchy level.
            </documentation>
        </annotation>
    </attribute>

```

```

        </attribute>
    </complexType>

    <complexType name="RegionRefType">
        <attribute name="regionRef" type="IDREF"
use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupType">
        <annotation>
            <documentation>Numbered group (contains ordered
elements)</documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"></element>
            <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"></element>
            <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"></element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>

    <complexType name="UnorderedGroupType">
        <annotation>
            <documentation>Numbered group (contains unordered
elements)</documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef" type="pc:RegionRefType"></element>
            <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
            <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>

    <complexType name="BorderType">
        <annotation>
            <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
    </complexType>
    <simpleType name="ColourSimpleType">
        <restriction base="string">
            <enumeration value="black"></enumeration>
            <enumeration value="blue"></enumeration>
            <enumeration value="brown"></enumeration>
            <enumeration value="cyan"></enumeration>
            <enumeration value="green"></enumeration>
            <enumeration value="grey"></enumeration>
            <enumeration value="indigo"></enumeration>
            <enumeration value="magenta"></enumeration>
            <enumeration value="orange"></enumeration>

```

```

        <enumeration value="pink"></enumeration>
        <enumeration value="red"></enumeration>
        <enumeration value="turquoise"></enumeration>
        <enumeration value="violet"></enumeration>
        <enumeration value="white"></enumeration>
        <enumeration value="yellow"></enumeration>
    </restriction>
</simpleType>
<simpleType name="ReadingDirectionSimpleType">
    <restriction base="string">
        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
        <enumeration value="top-to-bottom"></enumeration>
        <enumeration value="bottom-to-top"></enumeration>
    </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
    <restriction base="string">
        <enumeration value="paragraph"></enumeration>
        <enumeration value="heading"></enumeration>
        <enumeration value="caption"></enumeration>
        <enumeration value="header"></enumeration>
        <enumeration value="footer"></enumeration>
        <enumeration value="page-number"></enumeration>
        <enumeration value="drop-capital"></enumeration>
        <enumeration value="credit"></enumeration>
        <enumeration value="floating"></enumeration>
    </restriction>
</simpleType>
<simpleType name="LanguageSimpleType">
    <restriction base="string">
        <enumeration value="Afrikaans"></enumeration>
        <enumeration value="Albanian"></enumeration>
        <enumeration value="Amharic"></enumeration>
        <enumeration value="Arabic"></enumeration>
        <enumeration value="Basque"></enumeration>
        <enumeration value="Bengali"></enumeration>
        <enumeration value="Bulgarian"></enumeration>
        <enumeration value="Cambodian"></enumeration>
        <enumeration value="Cantonese"></enumeration>
        <enumeration value="Chinese"></enumeration>
        <enumeration value="Czech"></enumeration>
        <enumeration value="Danish"></enumeration>
        <enumeration value="Dutch"></enumeration>
        <enumeration value="English"></enumeration>
        <enumeration value="Estonian"></enumeration>
        <enumeration value="Finnish"></enumeration>
        <enumeration value="French"></enumeration>
        <enumeration value="German"></enumeration>
        <enumeration value="Greek"></enumeration>
        <enumeration value="Gujarati"></enumeration>
        <enumeration value="Hebrew"></enumeration>
        <enumeration value="Hindi"></enumeration>
        <enumeration value="Hungarian"></enumeration>
        <enumeration value="Icelandic"></enumeration>
        <enumeration value="Gaelic"></enumeration>
        <enumeration value="Italian"></enumeration>
        <enumeration value="Japanese"></enumeration>
        <enumeration value="Korean"></enumeration>
    </restriction>
</simpleType>

```



```

        <enumeration value="Latin"></enumeration>
    <enumeration value="Latvian"></enumeration>
    <enumeration value="Malay"></enumeration>
    <enumeration value="Norwegian"></enumeration>
    <enumeration value="Polish"></enumeration>
    <enumeration value="Portuguese"></enumeration>
    <enumeration value="Punjabi"></enumeration>
    <enumeration value="Russian"></enumeration>
    <enumeration value="Spanish"></enumeration>
    <enumeration value="Swedish"></enumeration>
    <enumeration value="Thai"></enumeration>
    <enumeration value="Turkish"></enumeration>
    <enumeration value="Urdu"></enumeration>
    <enumeration value="Welsh"></enumeration>
    <enumeration value="other"></enumeration>
</restriction>
</simpleType>

<simpleType name="ScriptSimpleType">
    <restriction base="string">
        <enumeration value="Arabic"></enumeration>
        <enumeration value="Bengali"></enumeration>
        <enumeration value="Chinese-simplified"></enumeration>
        <enumeration value="Chinese-traditional"></enumeration>
        <enumeration value="Cyrillic"></enumeration>
        <enumeration value="Devangari"></enumeration>
        <enumeration value="Ethiopic"></enumeration>
        <enumeration value="Greek"></enumeration>
        <enumeration value="Gujarati"></enumeration>
        <enumeration value="Gurmukhi"></enumeration>
        <enumeration value="Hebrew"></enumeration>
        <enumeration value="Latin"></enumeration>
        <enumeration value="Thai"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ColourDepthSimpleType">
    <restriction base="string">
        <enumeration value="bilevel"></enumeration>
        <enumeration value="greyscale"></enumeration>
        <enumeration value="colour"></enumeration>
    </restriction>
</simpleType>

<simpleType name="GraphicsTypeSimpleType">
    <restriction base="string">
        <enumeration value="Logo"></enumeration>
        <enumeration value="Letterhead"></enumeration>
        <enumeration value="handwritten-annotation"></enumeration>
        <enumeration value="stamp"></enumeration>
        <enumeration value="signature"></enumeration>
        <enumeration value="paper-grow"></enumeration>
        <enumeration value="punch-hole"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ChartTypeSimpleType">
    <restriction base="string">
        <enumeration value="bar"></enumeration>
        <enumeration value="line"></enumeration>
    </restriction>
</simpleType>

```



```

        <enumeration value="pie"></enumeration>
        <enumeration value="scatter"></enumeration>
        <enumeration value="surface"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<complexType name="LayersType">
    <annotation>
        <documentation>
            Can be used to express the z-index of overlapping
            regions. An element with a greater z-index is always in
            front of another element with lower z-index.
        </documentation>
    </annotation>
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="Layer" type="pc:LayerType"></element>
    </sequence>
</complexType>

<complexType name="LayerType">
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="zIndex" type="int" use="required"></attribute>
</complexType>

</schema>

```

Pagecontent Format Specification – Second Release

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-01-12"
  elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-01-12">

  <element name="PcGts" type="pc:PcGtsType">
    <annotation>
      <documentation>Page Content - Ground Truth and
Storage</documentation>
    </annotation></element>

    <complexType name="PcGtsType">
      <sequence>
        <element name="Metadata" type="pc:MetadataType"></element>
        <element name="Page" type="pc:PageType"></element>
      </sequence>
      <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
      <sequence>
        <element name="Creator" type="string"></element>
        <element name="Created" type="dateTime"></element>
        <element name="LastChange" type="dateTime"></element>
        <element name="Comments" type="string" minOccurs="0"
maxOccurs="1"></element>
      </sequence>
    </complexType>

    <complexType name="PageType">
      <sequence>
        <element name="Border" type="pc:BorderType" minOccurs="0"
maxOccurs="1">
</element>
        <element name="PrintSpace" type="pc:PrintSpaceType"
minOccurs="0" maxOccurs="1">
</element>
        <element name="ReadingOrder" type="pc:ReadingOrderType"
minOccurs="0" maxOccurs="1">
<annotation>
      <documentation>If the reading order element exists, all
regions have to be covered (i.e. all region ids must be mentioned exactly
once)!</documentation>
    </annotation>
</element>
        <element name="Layers" type="pc:LayersType" minOccurs="0"
maxOccurs="1">
<annotation>
      <documentation>If the layers element exists, all regions
have to be covered (i.e. all region ids must be mentioned exactly
once)!</documentation>
    </annotation>

```

```

        </annotation></element>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="TextRegion"
type="pc:TextRegionType"></element>
            <element name="ImageRegion"
                type="pc:ImageRegionType">
            </element>
            <element name="LineDrawingRegion"
                type="pc:LineDrawingRegionType">
            </element>
            <element name="GraphicRegion"
                type="pc:GraphicRegionType">
            </element>
            <element name="TableRegion"
                type="pc:TableRegionType">
            </element>
            <element name="ChartRegion"
                type="pc:ChartRegionType">
            </element>
            <element name="SeparatorRegion"
                type="pc:SeparatorRegionType">
            </element>
            <element name="MathsRegion"
                type="pc:MathsRegionType">
            </element>
            <element name="NoiseRegion"
                type="pc:NoiseRegionType">
            </element>
            <element name="FrameRegion"
                type="pc:FrameRegionType">
            </element>
            <element name="UnknownRegion"
                type="pc:UnknownRegionType">
            </element>
        </choice>
    </sequence>
    <attribute name="imageFilename" type="string"
use="required"></attribute>
    <attribute name="imageWidth" type="int" use="required"></attribute>
    <attribute name="imageHeight" type="int"
use="required"></attribute>
</complexType>
<complexType name="TextRegionType">
    <annotation>
        <documentation>Pure text is represented as a text region. This
includes drop capitals, but practically ornate text may be considered as a
graphic.</documentation></annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="1">
        </element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
    <annotation>

```

```

        <documentation>Individual skew of the region in
degrees (Range: -89.999,90)</documentation>
    </annotation></attribute>
    <attribute name="type" type="pc:TextTypeSimpleType" use="optional">
        <annotation>
            <documentation>The nature of the text in the region</documentation>
        </annotation>
    </attribute>
    <attribute name="textColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The text colour of the
region</documentation></annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The background colour of the
region</documentation></annotation>
    </attribute>
    <attribute name="reverseVideo" type="boolean" use="optional">
        <annotation>
            <documentation>Specifies whether the colour of the text appears
reversed against a background colour</documentation></annotation>
    </attribute>
    <attribute name="fontSize" type="float" use="optional">
        <annotation>
            <documentation>The size of the characters in
points</documentation></annotation></attribute>
    <attribute name="Leading" type="int" use="optional">
        <annotation>
            <documentation>The degree of space in points between
the lines of text</documentation>
        </annotation></attribute>
    <attribute name="kerning" type="int" use="optional">
        <annotation>
            <documentation>The degree of space in points between
the characters in a string of text</documentation>
        </annotation></attribute>
    <attribute name="readingDirection" type="pc:ReadingDirectionSimpleType"
use="optional">
        <annotation>
            <documentation>The direction in which text in a region should be
read (within lines)</documentation>
        </annotation>
    </attribute>
    <attribute name="readingOrientation" type="float" use="optional">
        <annotation>
            <documentation>The degrees by which you need to turn
your head in order to read the text when it is placed on the horizontal (Range:
-89.999,90)</documentation>
        </annotation></attribute>
    <attribute name="indented" type="boolean" use="optional">
        <annotation>
            <documentation>Defines whether a region of text is indented or
not</documentation></annotation>
    </attribute>
    <attribute name="primaryLanguage" type="pc:LanguageSimpleType"
use="optional">
        <annotation>

```

```

        <documentation>The primary language used in the
region</documentation></annotation>
        </attribute>
        <attribute name="secondaryLanguage" type="pc:LanguageSimpleType"
use="optional">
        <annotation>
        <documentation>The secondary language used in the
region</documentation>
        </annotation>
        </attribute>
        <attribute name="primaryScript" type="pc:ScriptSimpleType"
use="optional">
        <annotation>
        <documentation>The primary script used in the
region</documentation>
        </annotation>
        </attribute>
        <attribute name="secondaryScript" type="pc:ScriptSimpleType"
use="optional">
        <annotation>
        <documentation>The secondary script used in the
region</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="CoordsType">
        <sequence>
            <element name="Point" type="pc:PointType" minOccurs="0"
maxOccurs="unbounded"></element>
        </sequence>
    </complexType>
    <complexType name="PointType">
        <attribute name="x" type="int" use="required"></attribute>
        <attribute name="y" type="int" use="required"></attribute>
    </complexType>
    <complexType name="TextLineType">
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Word" type="pc:WordType" minOccurs="0"
maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"
minOccurs="0" maxOccurs="1">
            </element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="WordType">
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Glyph" type="pc:GlyphType" minOccurs="0"
maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"
minOccurs="0" maxOccurs="1">
            </element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="GlyphType">

```

```

    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
      <element name="TextEquiv" type="pc:TextEquivType"
        minOccurs="0" maxOccurs="1">
    </element>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="Ligature" use="optional" type="boolean">
</attribute>
  <attribute name="symbol" use="optional" type="boolean">
</attribute>
</complexType>
<complexType name="TextEquivType">
  <sequence>
    <element name="PlainText" type="string">
      <annotation>
        <documentation>Text in a "simple" form (ASCII
or extended ASCII as mostly used for typing).
I.e. no use of special characters for ligatures (should be stored as two
separate characters)
etc.</documentation>
      </annotation></element>
    <element name="Unicode" type="string">
      <annotation>
        <documentation>Correct encoding of the
original, always using the corresponding Unicode code point.
I.e. ligatures have to be represented as one character
etc.</documentation>
      </annotation></element>
    </sequence>
  </complexType>
  <complexType name="ImageRegionType">
    <annotation>
      <documentation>An image is considered to be more intricate and
complex than a graphic. These can be photos or
drawings.</documentation></annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
      <annotation>
        <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
      </annotation></attribute>
    <attribute name="colourDepth" type="pc:ColourDepthSimpleType"
use="optional">
      <annotation>
        <documentation>The colour bit depth required for the
region</documentation>
      </annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
      <annotation>
        <documentation>The background colour of the region</documentation>
      </annotation>
    </attribute>
    <attribute name="embText" type="boolean" use="optional">

```

```

        <annotation>
            <documentation>Specifies whether the region also contains
text</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="LineDrawingRegionType">
        <annotation>
            <documentation>A line drawing is a single colour illustration
without solid areas.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="orientation" type="float" use="optional">
            <annotation>
                <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
            </annotation></attribute>
        <attribute name="penColour" type="pc:ColourSimpleType" use="optional">
            <annotation>
                <documentation>The pen (foreground) colour of the
region</documentation></annotation>
            </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
            <annotation>
                <documentation>The background colour of the
region</documentation></annotation>
            </attribute>
        <attribute name="embText" type="boolean" use="optional">
            <annotation>
                <documentation>Specifies whether the region also contains
text</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="GraphicRegionType">
        <annotation>
            <documentation>Regions containing simple graphics, such as a
company logo, should be marked as graphic regions.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="orientation" type="float" use="optional">
            <annotation>
                <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90).</documentation>
            </annotation></attribute>
        <attribute name="type" use="optional" type="pc:GraphicsTypeSimpleType">
            <annotation>
                <documentation>The type of graphic in the
region</documentation></annotation>
            </attribute>
        <attribute name="numColours" type="int" use="optional">
            <annotation>

```

```

        <documentation>An approximation of the number of
colours used in the region</documentation>
    </annotation></attribute>
    <attribute name="embText" type="boolean" use="optional">
        <annotation>
            <documentation>Specifies whether the region also contains
text.</documentation>
        </annotation>
    </attribute>
</complexType>
<complexType name="TableRegionType">
    <annotation>
        <documentation>Tabular data in any form is represented with a table
region. Rows and columns may or may not have separator lines; these lines are
not separator regions.</documentation></annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
        <annotation>
            <documentation>The orientation in degrees of the
baseline of the region (Range: -89.999,90).</documentation>
        </annotation></attribute>
    <attribute name="rows" type="int" use="optional">
        <annotation>
            <documentation>The number of rows present in the
table</documentation>
        </annotation></attribute>
    <attribute name="columns" type="int" use="optional">
        <annotation>
            <documentation>The number of columns present in the
table</documentation>
        </annotation></attribute>
    <attribute name="lineColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The colour of the lines used in the
region</documentation>
        </annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The background colour of the region</documentation>
        </annotation>
    </attribute>
    <attribute name="lineSeparators" type="boolean" use="optional">
        <annotation>
            <documentation>Specifies the presence of line
separators</documentation>
        </annotation>
    </attribute>
    <attribute name="embText" type="boolean" use="optional">
        <annotation>
            <documentation>Specifies whether the region also contains
text</documentation>
        </annotation>
    </attribute>
</complexType>
<complexType name="ChartRegionType">

```



```

    <annotation>
      <documentation>Regions containing charts or graphs of any type,
should be marked as chart regions.</documentation></annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
      <annotation>
        <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
      </annotation></attribute>
    <attribute name="type" use="optional" type="pc:ChartTypeSimpleType">
      <annotation>
        <documentation>The type of chart in the
region</documentation></annotation>
      </attribute>
    <attribute name="numColours" type="int" use="optional">
      <annotation>
        <documentation>An approximation of the number of
colours used in the region</documentation></annotation></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
      <annotation>
        <documentation>The background colour of the region</documentation>
      </annotation>
      </attribute>
    <attribute name="embText" type="boolean" use="optional">
      <annotation>
        <documentation>Specifies whether the region also contains
text</documentation></annotation>
      </attribute>
    </complexType>
    <complexType name="SeparatorRegionType">
      <annotation>
        <documentation>Separators are lines that lie between columns and
paragraphs and can be used to logically separate different articles from each
other.</documentation></annotation>
      <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
      </sequence>
      <attribute name="id" type="ID" use="required"></attribute>
      <attribute name="orientation" type="float"
use="optional">
        <annotation>
          <documentation>The orientation in degrees of the
region (Range: -89.999,90)</documentation>
        </annotation></attribute>
      <attribute name="colour" type="pc:ColourSimpleType" use="optional">
        <annotation>
          <documentation>The colour of the separator</documentation>
        </annotation>
      </attribute>
    </complexType>
    <complexType name="MathsRegionType">
      <annotation>
        <documentation>Regions containing equations and mathematical
symbols should be marked as maths regions.</documentation></annotation>
      <sequence>

```

```

        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
        <annotation>
            <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
        </annotation></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The background colour of the
region</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="NoiseRegionType">
        <annotation>
            <documentation>Noise regions are regions where no real data lies,
only false data created by artifacts on the document or scanner
noise.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="UnknownRegionType">
        <annotation>
            <documentation>To be used if the region type cannot be
ascertained.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="FrameRegionType">
        <annotation>
            <documentation>
                A region that surrounds other regions (e.g. a box with
                blue background containing text regions)
            </documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <choice minOccurs="0" maxOccurs="unbounded">
                <element name="TextRegion" type="pc:TextRegionType">
                    </element>
                <element name="ImageRegion"
                    type="pc:ImageRegionType">
                    </element>
                <element name="LineDrawingRegion"
                    type="pc:LineDrawingRegionType">
                    </element>
                <element name="GraphicRegion"
                    type="pc:GraphicRegionType">
                    </element>
                <element name="TableRegion"
                    type="pc:TableRegionType">
                    </element>
            </choice>
        </sequence>
    </complexType>

```

```

        </element>
        <element name="ChartRegion"
            type="pc:ChartRegionType">
        </element>
        <element name="SeparatorRegion"
            type="pc:SeparatorRegionType">
        </element>
        <element name="MathsRegion"
            type="pc:MathsRegionType">
        </element>
        <element name="NoiseRegion"
            type="pc:NoiseRegionType">
        </element>
        <element name="FrameRegion"
            type="pc:FrameRegionType">
        </element>
        <element name="UnknownRegion"
            type="pc:UnknownRegionType"></element>
    </choice>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="bgColour" type="pc:ColourSimpleType"
    use="optional">
</attribute>
<attribute name="borderPresent" type="boolean"
    use="optional"></attribute>
</complexType>

<complexType name="PrintSpaceType">
    <annotation>
        <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.
It does not contain pagenumber (if not part of running title), marginals,
signature mark, preview words.
</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>

<complexType name="ReadingOrderType">
    <annotation>
        <documentation>Definition of the reading order within the page.
Without further grouping, regions are supposed to be unordered on the page.
To express a reading order between elements they have to be included in a
ordered group. Groups may contain further groups.</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
        <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
            type="pc:UnorderedGroupType"></element>
    </choice>
</complexType>

```

```

    <complexType name="RegionRefIndexedType">
      <annotation>
        <documentation>Numbered region</documentation>
      </annotation>
      <attribute name="index" type="int" use="required">
        <annotation>
          <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
        </annotation></attribute>
      <attribute name="regionRef" type="IDREF" use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupIndexedType">
      <annotation>
        <documentation>Indexed group containing ordered
elements</documentation>
      </annotation>
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"></element>
        <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"></element>
        <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"></element>
      </choice>
      <attribute name="id" type="ID" use="required"></attribute>
      <attribute name="index" type="int" use="required">
        <annotation>
          <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
        </annotation></attribute>
    </complexType>

    <complexType name="UnorderedGroupIndexedType">
      <annotation>
        <documentation>Indexed group containing unordered
elements</documentation>
      </annotation>
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
        <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
      </choice>
      <attribute name="id" type="ID" use="required"></attribute>
      <attribute name="index" type="int" use="required">
        <annotation>
          <documentation>
            Position (order number) of this item within the
            current hierarchy level.
          </documentation>
        </annotation>
      </attribute>
    </complexType>

    <complexType name="RegionRefType">
      <attribute name="regionRef" type="IDREF"
use="required"></attribute>

```

```

    </complexType>

    <complexType name="OrderedGroupType">
      <annotation>
        <documentation>Numbered group (contains ordered
elements)</documentation>
      </annotation>
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"></element>
        <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"></element>
        <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"></element>
      </choice>
      <attribute name="id" type="ID" use="required"></attribute>
    </complexType>

    <complexType name="UnorderedGroupType">
      <annotation>
        <documentation>Numbered group (contains unordered
elements)</documentation>
      </annotation>
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
        <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
      </choice>
      <attribute name="id" type="ID" use="required"></attribute>
    </complexType>

    <complexType name="BorderType">
      <annotation>
        <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>
      </annotation>
      <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
      </sequence>
    </complexType>

    <simpleType name="ColourSimpleType">
      <restriction base="string">
        <enumeration value="black"></enumeration>
        <enumeration value="blue"></enumeration>
        <enumeration value="brown"></enumeration>
        <enumeration value="cyan"></enumeration>
        <enumeration value="green"></enumeration>
        <enumeration value="grey"></enumeration>
        <enumeration value="indigo"></enumeration>
        <enumeration value="magenta"></enumeration>
        <enumeration value="orange"></enumeration>
        <enumeration value="pink"></enumeration>
        <enumeration value="red"></enumeration>
        <enumeration value="turquoise"></enumeration>
        <enumeration value="violet"></enumeration>
        <enumeration value="white"></enumeration>
        <enumeration value="yellow"></enumeration>
      </restriction>
    </simpleType>

```

```

    </restriction>
</simpleType>
<simpleType name="ReadingDirectionSimpleType">
  <restriction base="string">
    <enumeration value="left-to-right"></enumeration>
    <enumeration value="right-to-left"></enumeration>
    <enumeration value="top-to-bottom"></enumeration>
    <enumeration value="bottom-to-top"></enumeration>
  </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
  <restriction base="string">
    <enumeration value="paragraph"></enumeration>
    <enumeration value="heading"></enumeration>
    <enumeration value="caption"></enumeration>
    <enumeration value="header"></enumeration>
    <enumeration value="footer"></enumeration>
    <enumeration value="page-number"></enumeration>
    <enumeration value="drop-capital"></enumeration>
    <enumeration value="credit"></enumeration>
    <enumeration value="floating"></enumeration>
  </restriction>
</simpleType>
<simpleType name="LanguageSimpleType">
  <restriction base="string">
    <enumeration value="Afrikaans"></enumeration>
    <enumeration value="Albanian"></enumeration>
    <enumeration value="Amharic"></enumeration>
    <enumeration value="Arabic"></enumeration>
    <enumeration value="Basque"></enumeration>
    <enumeration value="Bengali"></enumeration>
    <enumeration value="Bulgarian"></enumeration>
    <enumeration value="Cambodian"></enumeration>
    <enumeration value="Cantonese"></enumeration>
    <enumeration value="Chinese"></enumeration>
    <enumeration value="Czech"></enumeration>
    <enumeration value="Danish"></enumeration>
    <enumeration value="Dutch"></enumeration>
    <enumeration value="English"></enumeration>
    <enumeration value="Estonian"></enumeration>
    <enumeration value="Finnish"></enumeration>
    <enumeration value="French"></enumeration>
    <enumeration value="German"></enumeration>
    <enumeration value="Greek"></enumeration>
    <enumeration value="Gujarati"></enumeration>
    <enumeration value="Hebrew"></enumeration>
    <enumeration value="Hindi"></enumeration>
    <enumeration value="Hungarian"></enumeration>
    <enumeration value="Icelandic"></enumeration>
    <enumeration value="Gaelic"></enumeration>
    <enumeration value="Italian"></enumeration>
    <enumeration value="Japanese"></enumeration>
    <enumeration value="Korean"></enumeration>
    <enumeration value="Latin"></enumeration>
    <enumeration value="Latvian"></enumeration>
    <enumeration value="Malay"></enumeration>
    <enumeration value="Norwegian"></enumeration>
    <enumeration value="Polish"></enumeration>
    <enumeration value="Portuguese"></enumeration>
  </restriction>
</simpleType>

```

```

        <enumeration value="Punjabi"></enumeration>
        <enumeration value="Russian"></enumeration>
        <enumeration value="Spanish"></enumeration>
        <enumeration value="Swedish"></enumeration>
        <enumeration value="Thai"></enumeration>
        <enumeration value="Turkish"></enumeration>
        <enumeration value="Urdu"></enumeration>
        <enumeration value="Welsh"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ScriptSimpleType">
    <restriction base="string">
        <enumeration value="Arabic"></enumeration>
        <enumeration value="Bengali"></enumeration>
        <enumeration value="Chinese-simplified"></enumeration>
        <enumeration value="Chinese-traditional"></enumeration>
        <enumeration value="Cyrillic"></enumeration>
        <enumeration value="Devangari"></enumeration>
        <enumeration value="Ethiopic"></enumeration>
        <enumeration value="Greek"></enumeration>
        <enumeration value="Gujarati"></enumeration>
        <enumeration value="Gurmukhi"></enumeration>
        <enumeration value="Hebrew"></enumeration>
        <enumeration value="Latin"></enumeration>
        <enumeration value="Thai"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ColourDepthSimpleType">
    <restriction base="string">
        <enumeration value="bilevel"></enumeration>
        <enumeration value="greyscale"></enumeration>
        <enumeration value="colour"></enumeration>
    </restriction>
</simpleType>

<simpleType name="GraphicsTypeSimpleType">
    <restriction base="string">
        <enumeration value="Logo"></enumeration>
        <enumeration value="Letterhead"></enumeration>
        <enumeration value="handwritten-annotation"></enumeration>
        <enumeration value="stamp"></enumeration>
        <enumeration value="signature"></enumeration>
        <enumeration value="paper-grow"></enumeration>
        <enumeration value="punch-hole"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ChartTypeSimpleType">
    <restriction base="string">
        <enumeration value="bar"></enumeration>
        <enumeration value="line"></enumeration>
        <enumeration value="pie"></enumeration>
        <enumeration value="scatter"></enumeration>
        <enumeration value="surface"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

```

```

<complexType name="LayersType">
  <annotation>
    <documentation>
      Can be used to express the z-index of overlapping
      regions. An element with a greater z-index is always in
      front of another element with lower z-index.
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Layer" type="pc:LayerType"></element>
  </sequence>
</complexType>

<complexType name="LayerType">
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="RegionRef" type="pc:RegionRefType"></element>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="zIndex" type="int" use="required"></attribute>
</complexType>

</schema>

```


Pagecontent Format Specification – Third Release

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-03-
19"
    elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2010-03-
19">

    <element name="PcGts" type="pc:PcGtsType">
        <annotation>
            <documentation>Page Content - Ground Truth and
Storage</documentation>
        </annotation></element>

    <complexType name="PcGtsType">
        <sequence>
            <element name="Metadata" type="pc:MetadataType"></element>
            <element name="Page" type="pc:PageType"></element>
        </sequence>
        <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
        <sequence>
            <element name="Creator" type="string"></element>
            <element name="Created" type="dateTime"></element>
            <element name="LastChange" type="dateTime"></element>
            <element name="Comments" type="string" minOccurs="0"
                maxOccurs="1"></element>
        </sequence>
    </complexType>

    <complexType name="PageType">
        <sequence>
            <element name="Border" type="pc:BorderType" minOccurs="0"
                maxOccurs="1">
            </element>
            <element name="PrintSpace" type="pc:PrintSpaceType"
                minOccurs="0" maxOccurs="1">
            </element>
            <element name="ReadingOrder" type="pc:ReadingOrderType"
                minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation>If the reading order element exists, all
regions have to be covered (i.e. all region ids must be mentioned exactly
once)!</documentation>
            </annotation>
            </element>
            <element name="Layers" type="pc:LayersType" minOccurs="0"
                maxOccurs="1">
            <annotation>
                <documentation>If the layers element exists, all regions
have to be covered (i.e. all region ids must be mentioned exactly
once)!</documentation>
            </annotation>
            </element>
        </sequence>
    </complexType>
</schema>
```

```

        </annotation></element>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="TextRegion"
type="pc:TextRegionType"></element>
            <element name="ImageRegion"
                type="pc:ImageRegionType">
            </element>
            <element name="LineDrawingRegion"
                type="pc:LineDrawingRegionType">
            </element>
            <element name="GraphicRegion"
                type="pc:GraphicRegionType">
            </element>
            <element name="TableRegion"
                type="pc:TableRegionType">
            </element>
            <element name="ChartRegion"
                type="pc:ChartRegionType">
            </element>
            <element name="SeparatorRegion"
                type="pc:SeparatorRegionType">
            </element>
            <element name="MathsRegion"
                type="pc:MathsRegionType">
            </element>
            <element name="NoiseRegion"
                type="pc:NoiseRegionType">
            </element>
            <element name="FrameRegion"
                type="pc:FrameRegionType">
            </element>
            <element name="UnknownRegion"
                type="pc:UnknownRegionType">
            </element>
        </choice>
    </sequence>
    <attribute name="imageFilename" type="string"
use="required"></attribute>
    <attribute name="imageWidth" type="int" use="required"></attribute>
    <attribute name="imageHeight" type="int"
use="required"></attribute>
</complexType>
<complexType name="TextRegionType">
    <annotation>
        <documentation>Pure text is represented as a text region. This
includes drop capitals, but practically ornate text may be considered as a
graphic.</documentation></annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="1">
        </element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
    <annotation>

```

```

        <documentation>Individual skew of the region in
degrees (Range: -89.999,90)</documentation>
    </annotation></attribute>
    <attribute name="type" type="pc:TextTypeSimpleType" use="optional">
        <annotation>
            <documentation>The nature of the text in the region</documentation>
        </annotation>
    </attribute>
    <attribute name="textColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The text colour of the
region</documentation></annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The background colour of the
region</documentation></annotation>
    </attribute>
    <attribute name="reverseVideo" type="boolean" use="optional">
        <annotation>
            <documentation>Specifies whether the colour of the text appears
reversed against a background colour</documentation></annotation>
    </attribute>
    <attribute name="fontSize" type="float" use="optional">
        <annotation>
            <documentation>The size of the characters in
points</documentation></annotation></attribute>
    <attribute name="Leading" type="int" use="optional">
        <annotation>
            <documentation>The degree of space in points between
the lines of text</documentation>
    </annotation></attribute>
    <attribute name="kerning" type="int" use="optional">
        <annotation>
            <documentation>The degree of space in points between
the characters in a string of text</documentation>
    </annotation></attribute>
    <attribute name="readingDirection" type="pc:ReadingDirectionSimpleType"
use="optional">
        <annotation>
            <documentation>The direction in which text in a region should be
read (within lines)</documentation>
    </annotation>
    </attribute>
    <attribute name="readingOrientation" type="float" use="optional">
        <annotation>
            <documentation>The degrees by which you need to turn
your head in order to read the text when it is placed on the horizontal (Range:
-89.999,90)</documentation>
    </annotation></attribute>
    <attribute name="indented" type="boolean" use="optional">
        <annotation>
            <documentation>Defines whether a region of text is indented or
not</documentation></annotation>
    </attribute>
    <attribute name="primaryLanguage" type="pc:LanguageSimpleType"
use="optional">
        <annotation>

```

```

        <documentation>The primary language used in the
region</documentation></annotation>
        </attribute>
        <attribute name="secondaryLanguage" type="pc:LanguageSimpleType"
use="optional">
        <annotation>
        <documentation>The secondary language used in the
region</documentation>
        </annotation>
        </attribute>
        <attribute name="primaryScript" type="pc:ScriptSimpleType"
use="optional">
        <annotation>
        <documentation>The primary script used in the
region</documentation>
        </annotation>
        </attribute>
        <attribute name="secondaryScript" type="pc:ScriptSimpleType"
use="optional">
        <annotation>
        <documentation>The secondary script used in the
region</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="CoordsType">
        <sequence>
            <element name="Point" type="pc:PointType" minOccurs="0"
maxOccurs="unbounded"></element>
        </sequence>
    </complexType>
    <complexType name="PointType">
        <attribute name="x" type="int" use="required"></attribute>
        <attribute name="y" type="int" use="required"></attribute>
    </complexType>
    <complexType name="TextLineType">
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Word" type="pc:WordType" minOccurs="0"
maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"
minOccurs="0" maxOccurs="1">
            </element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="WordType">
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Glyph" type="pc:GlyphType" minOccurs="0"
maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"
minOccurs="0" maxOccurs="1">
            </element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="GlyphType">

```

```

    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
      <element name="TextEquiv" type="pc:TextEquivType"
        minOccurs="0" maxOccurs="1">
    </element>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="Ligature" use="optional" type="boolean">
</attribute>
  <attribute name="symbol" use="optional" type="boolean">
</attribute>
</complexType>
<complexType name="TextEquivType">
  <sequence>
    <element name="PlainText" type="string">
      <annotation>
        <documentation>Text in a "simple" form (ASCII
or extended ASCII as mostly used for typing).
I.e. no use of special characters for ligatures (should be stored as two
separate characters)
etc.</documentation>
      </annotation></element>
    <element name="Unicode" type="string">
      <annotation>
        <documentation>Correct encoding of the
original, always using the corresponding Unicode code point.
I.e. ligatures have to be represented as one character
etc.</documentation>
      </annotation></element>
    </sequence>
  </complexType>
  <complexType name="ImageRegionType">
    <annotation>
      <documentation>An image is considered to be more intricate and
complex than a graphic. These can be photos or
drawings.</documentation></annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
      <annotation>
        <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
      </annotation></attribute>
    <attribute name="colourDepth" type="pc:ColourDepthSimpleType"
use="optional">
      <annotation>
        <documentation>The colour bit depth required for the
region</documentation>
      </annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
      <annotation>
        <documentation>The background colour of the region</documentation>
      </annotation>
    </attribute>
    <attribute name="embText" type="boolean" use="optional">

```

```

        <annotation>
            <documentation>Specifies whether the region also contains
text</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="LineDrawingRegionType">
        <annotation>
            <documentation>A line drawing is a single colour illustration
without solid areas.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="orientation" type="float" use="optional">
            <annotation>
                <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
            </annotation></attribute>
        <attribute name="penColour" type="pc:ColourSimpleType" use="optional">
            <annotation>
                <documentation>The pen (foreground) colour of the
region</documentation></annotation>
            </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
            <annotation>
                <documentation>The background colour of the
region</documentation></annotation>
            </attribute>
        <attribute name="embText" type="boolean" use="optional">
            <annotation>
                <documentation>Specifies whether the region also contains
text</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="GraphicRegionType">
        <annotation>
            <documentation>Regions containing simple graphics, such as a
company logo, should be marked as graphic regions.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="orientation" type="float" use="optional">
            <annotation>
                <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90).</documentation>
            </annotation></attribute>
        <attribute name="type" use="optional" type="pc:GraphicsTypeSimpleType">
            <annotation>
                <documentation>The type of graphic in the
region</documentation></annotation>
            </attribute>
        <attribute name="numColours" type="int" use="optional">
            <annotation>

```

```

        <documentation>An approximation of the number of
colours used in the region</documentation>
        </annotation></attribute>
        <attribute name="embText" type="boolean" use="optional">
        <annotation>
        <documentation>Specifies whether the region also contains
text.</documentation>
        </annotation>
        </attribute>
    </complexType>
    <complexType name="TableRegionType">
    <annotation>
    <documentation>Tabular data in any form is represented with a table
region. Rows and columns may or may not have separator lines; these lines are
not separator regions.</documentation></annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
    <annotation>
    <documentation>The orientation in degrees of the
baseline of the region (Range: -89.999,90).</documentation>
    </annotation></attribute>
    <attribute name="rows" type="int" use="optional">
    <annotation>
    <documentation>The number of rows present in the
table</documentation>
    </annotation></attribute>
    <attribute name="columns" type="int" use="optional">
    <annotation>
    <documentation>The number of columns present in the
table</documentation>
    </annotation></attribute>
    <attribute name="lineColour" type="pc:ColourSimpleType" use="optional">
    <annotation>
    <documentation>The colour of the lines used in the
region</documentation>
    </annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
    <annotation>
    <documentation>The background colour of the region</documentation>
    </annotation>
    </attribute>
    <attribute name="lineSeparators" type="boolean" use="optional">
    <annotation>
    <documentation>Specifies the presence of line
separators</documentation>
    </annotation>
    </attribute>
    <attribute name="embText" type="boolean" use="optional">
    <annotation>
    <documentation>Specifies whether the region also contains
text</documentation>
    </annotation>
    </attribute>
    </complexType>
    <complexType name="ChartRegionType">

```

```

    <annotation>
      <documentation>Regions containing charts or graphs of any type,
should be marked as chart regions.</documentation></annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
      <annotation>
        <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
      </annotation></attribute>
    <attribute name="type" use="optional" type="pc:ChartTypeSimpleType">
      <annotation>
        <documentation>The type of chart in the
region</documentation></annotation>
      </attribute>
    <attribute name="numColours" type="int" use="optional">
      <annotation>
        <documentation>An approximation of the number of
colours used in the region</documentation></annotation></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
      <annotation>
        <documentation>The background colour of the region</documentation>
      </annotation>
      </attribute>
    <attribute name="embText" type="boolean" use="optional">
      <annotation>
        <documentation>Specifies whether the region also contains
text</documentation></annotation>
      </attribute>
    </complexType>
    <complexType name="SeparatorRegionType">
      <annotation>
        <documentation>Separators are lines that lie between columns and
paragraphs and can be used to logically separate different articles from each
other.</documentation></annotation>
      <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
      </sequence>
      <attribute name="id" type="ID" use="required"></attribute>
      <attribute name="orientation" type="float"
use="optional">
        <annotation>
          <documentation>The orientation in degrees of the
region (Range: -89.999,90)</documentation>
        </annotation></attribute>
      <attribute name="colour" type="pc:ColourSimpleType" use="optional">
        <annotation>
          <documentation>The colour of the separator</documentation>
        </annotation>
      </attribute>
    </complexType>
    <complexType name="MathsRegionType">
      <annotation>
        <documentation>Regions containing equations and mathematical
symbols should be marked as maths regions.</documentation></annotation>
      <sequence>

```



```

        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="orientation" type="float" use="optional">
        <annotation>
            <documentation>The orientation in degrees of the
baseline of the rectangle that encapsulates the region (Range: -
89.999,90)</documentation>
        </annotation></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType" use="optional">
        <annotation>
            <documentation>The background colour of the
region</documentation></annotation>
        </attribute>
    </complexType>
    <complexType name="NoiseRegionType">
        <annotation>
            <documentation>Noise regions are regions where no real data lies,
only false data created by artifacts on the document or scanner
noise.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="UnknownRegionType">
        <annotation>
            <documentation>To be used if the region type cannot be
ascertained.</documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
    </complexType>
    <complexType name="FrameRegionType">
        <annotation>
            <documentation>
                A region that surrounds other regions (e.g. a box with
                blue background containing text regions)
            </documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"></element>
            <choice minOccurs="0" maxOccurs="unbounded">
                <element name="TextRegion" type="pc:TextRegionType">
                    </element>
                <element name="ImageRegion"
                    type="pc:ImageRegionType">
                    </element>
                <element name="LineDrawingRegion"
                    type="pc:LineDrawingRegionType">
                    </element>
                <element name="GraphicRegion"
                    type="pc:GraphicRegionType">
                    </element>
                <element name="TableRegion"
                    type="pc:TableRegionType">
                    </element>
            </choice>
        </sequence>
    </complexType>

```

```

        </element>
        <element name="ChartRegion"
            type="pc:ChartRegionType">
        </element>
        <element name="SeparatorRegion"
            type="pc:SeparatorRegionType">
        </element>
        <element name="MathsRegion"
            type="pc:MathsRegionType">
        </element>
        <element name="NoiseRegion"
            type="pc:NoiseRegionType">
        </element>
        <element name="FrameRegion"
            type="pc:FrameRegionType">
        </element>
        <element name="UnknownRegion"
            type="pc:UnknownRegionType"></element>
    </choice>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="bgColour" type="pc:ColourSimpleType"
    use="optional">
</attribute>
<attribute name="borderPresent" type="boolean"
    use="optional"></attribute>
</complexType>

<complexType name="PrintSpaceType">
    <annotation>
        <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.
It does not contain pagenumber (if not part of running title), marginals,
signature mark, preview words.
</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>

<complexType name="ReadingOrderType">
    <annotation>
        <documentation>Definition of the reading order within the page. To
express a reading order between elements they have to be included in an
OrderedGroup. Groups may contain further groups.</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="1">
        <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
            type="pc:UnorderedGroupType"></element>
    </choice>
</complexType>

<complexType name="RegionRefIndexedType">
    <annotation>

```

```

        <documentation>Numbered region</documentation>
    </annotation>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
        </annotation></attribute>
    <attribute name="regionRef" type="IDREF" use="required"></attribute>
</complexType>

<complexType name="OrderedGroupIndexedType">
    <annotation>
        <documentation>Indexed group containing ordered
elements</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"></element>
        <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"></element>
        <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"></element>
    </choice>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
        </annotation></attribute>
</complexType>

<complexType name="UnorderedGroupIndexedType">
    <annotation>
        <documentation>Indexed group containing unordered
elements</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
        <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
    </choice>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>
                Position (order number) of this item within the
                current hierarchy level.
            </documentation>
        </annotation>
    </attribute>
</complexType>

<complexType name="RegionRefType">
    <attribute name="regionRef" type="IDREF"
use="required"></attribute>
</complexType>

```

```

    <complexType name="OrderedGroupType">
      <annotation>
        <documentation>Numbered group (contains ordered
elements)</documentation>
      </annotation>
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"></element>
        <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"></element>
        <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"></element>
      </choice>
      <attribute name="id" type="ID" use="required"></attribute>
    </complexType>

    <complexType name="UnorderedGroupType">
      <annotation>
        <documentation>Numbered group (contains unordered
elements)</documentation>
      </annotation>
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
        <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
      </choice>
      <attribute name="id" type="ID" use="required"></attribute>
    </complexType>

    <complexType name="BorderType">
      <annotation>
        <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>
      </annotation>
      <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
      </sequence>
    </complexType>
    <simpleType name="ColourSimpleType">
      <restriction base="string">
        <enumeration value="black"></enumeration>
        <enumeration value="blue"></enumeration>
        <enumeration value="brown"></enumeration>
        <enumeration value="cyan"></enumeration>
        <enumeration value="green"></enumeration>
        <enumeration value="grey"></enumeration>
        <enumeration value="indigo"></enumeration>
        <enumeration value="magenta"></enumeration>
        <enumeration value="orange"></enumeration>
        <enumeration value="pink"></enumeration>
        <enumeration value="red"></enumeration>
        <enumeration value="turquoise"></enumeration>
        <enumeration value="violet"></enumeration>
        <enumeration value="white"></enumeration>
        <enumeration value="yellow"></enumeration>
      </restriction>
    </simpleType>

```

```

<simpleType name="ReadingDirectionSimpleType">
  <restriction base="string">
    <enumeration value="left-to-right"></enumeration>
    <enumeration value="right-to-left"></enumeration>
    <enumeration value="top-to-bottom"></enumeration>
    <enumeration value="bottom-to-top"></enumeration>
  </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
  <restriction base="string">
    <enumeration value="paragraph"></enumeration>
    <enumeration value="heading"></enumeration>
    <enumeration value="caption"></enumeration>
    <enumeration value="header"></enumeration>
    <enumeration value="footer"></enumeration>
    <enumeration value="page-number"></enumeration>
    <enumeration value="drop-capital"></enumeration>
    <enumeration value="credit"></enumeration>
    <enumeration value="floating"></enumeration>
    <enumeration value="signature-mark"></enumeration>
    <enumeration value="catch-word"></enumeration>
    <enumeration value="marginalia"></enumeration>
    <enumeration value="footnote"></enumeration>
    <enumeration value="footnote-continued"></enumeration>
    <enumeration value="TOC-entry"></enumeration>
  </restriction>
</simpleType>
<simpleType name="LanguageSimpleType">
  <restriction base="string">
    <enumeration value="Afrikaans"></enumeration>
    <enumeration value="Albanian"></enumeration>
    <enumeration value="Amharic"></enumeration>
    <enumeration value="Arabic"></enumeration>
    <enumeration value="Basque"></enumeration>
    <enumeration value="Bengali"></enumeration>
    <enumeration value="Bulgarian"></enumeration>
    <enumeration value="Cambodian"></enumeration>
    <enumeration value="Cantonese"></enumeration>
    <enumeration value="Chinese"></enumeration>
    <enumeration value="Czech"></enumeration>
    <enumeration value="Danish"></enumeration>
    <enumeration value="Dutch"></enumeration>
    <enumeration value="English"></enumeration>
    <enumeration value="Estonian"></enumeration>
    <enumeration value="Finnish"></enumeration>
    <enumeration value="French"></enumeration>
    <enumeration value="German"></enumeration>
    <enumeration value="Greek"></enumeration>
    <enumeration value="Gujarati"></enumeration>
    <enumeration value="Hebrew"></enumeration>
    <enumeration value="Hindi"></enumeration>
    <enumeration value="Hungarian"></enumeration>
    <enumeration value="Icelandic"></enumeration>
    <enumeration value="Gaelic"></enumeration>
    <enumeration value="Italian"></enumeration>
    <enumeration value="Japanese"></enumeration>
    <enumeration value="Korean"></enumeration>
    <enumeration value="Latin"></enumeration>
    <enumeration value="Latvian"></enumeration>
  </restriction>
</simpleType>

```

```

        <enumeration value="Malay"></enumeration>
        <enumeration value="Norwegian"></enumeration>
        <enumeration value="Polish"></enumeration>
        <enumeration value="Portuguese"></enumeration>
        <enumeration value="Punjabi"></enumeration>
        <enumeration value="Russian"></enumeration>
        <enumeration value="Spanish"></enumeration>
        <enumeration value="Swedish"></enumeration>
        <enumeration value="Thai"></enumeration>
        <enumeration value="Turkish"></enumeration>
        <enumeration value="Urdu"></enumeration>
        <enumeration value="Welsh"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ScriptSimpleType">
    <restriction base="string">
        <enumeration value="Arabic"></enumeration>
        <enumeration value="Bengali"></enumeration>
        <enumeration value="Chinese-simplified"></enumeration>
        <enumeration value="Chinese-traditional"></enumeration>
        <enumeration value="Cyrillic"></enumeration>
        <enumeration value="Devangari"></enumeration>
        <enumeration value="Ethiopic"></enumeration>
        <enumeration value="Greek"></enumeration>
        <enumeration value="Gujarati"></enumeration>
        <enumeration value="Gurmukhi"></enumeration>
        <enumeration value="Hebrew"></enumeration>
        <enumeration value="Latin"></enumeration>
        <enumeration value="Thai"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ColourDepthSimpleType">
    <restriction base="string">
        <enumeration value="biLevel"></enumeration>
        <enumeration value="greyscale"></enumeration>
        <enumeration value="colour"></enumeration>
    </restriction>
</simpleType>

<simpleType name="GraphicsTypeSimpleType">
    <restriction base="string">
        <enumeration value="Logo"></enumeration>
        <enumeration value="Letterhead"></enumeration>
        <enumeration value="handwritten-annotation"></enumeration>
        <enumeration value="stamp"></enumeration>
        <enumeration value="signature"></enumeration>
        <enumeration value="paper-grow"></enumeration>
        <enumeration value="punch-hole"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="ChartTypeSimpleType">
    <restriction base="string">
        <enumeration value="bar"></enumeration>
        <enumeration value="line"></enumeration>
        <enumeration value="pie"></enumeration>
        <enumeration value="scatter"></enumeration>
    </restriction>
</simpleType>

```

```

        <enumeration value="surface"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<complexType name="LayersType">
    <annotation>
        <documentation>
            Can be used to express the z-index of overlapping
            regions. An element with a greater z-index is always in
            front of another element with lower z-index.
        </documentation>
    </annotation>
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="Layer" type="pc:LayerType"></element>
    </sequence>
</complexType>

<complexType name="LayerType">
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="zIndex" type="int" use="required"></attribute>
</complexType>

</schema>

```

Pagecontent Format Specification – Fourth Release

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2013-07-
15"
    elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2013-07-
15">

    <element name="PcGts" type="pc:PcGtsType">
        <annotation>
            <documentation>Page Content - Ground Truth and
Storage</documentation>
        </annotation></element>

    <complexType name="PcGtsType">
        <sequence>
            <element name="Metadata" type="pc:MetadataType"></element>
            <element name="Page" type="pc:PageType"></element>
        </sequence>
        <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
        <sequence>
            <element name="Creator" type="string"></element>
            <element name="Created" type="dateTime">
                <annotation>
                    <documentation>The timestamp has to be in UTC
(Coordinated Universal Time) and not local
time.</documentation></annotation></element>
            <element name="LastChange" type="dateTime">
                <annotation>
                    <documentation>The timestamp has to be in UTC
(Coordinated Universal Time) and not local
time.</documentation></annotation></element>
            <element name="Comments" type="string" minOccurs="0"
maxOccurs="1"></element>
        </sequence>
    </complexType>

    <complexType name="PageType">
        <sequence>
            <element name="AlternativeImage"
type="pc:AlternativeImageType" minOccurs="0"
maxOccurs="unbounded">
                <annotation>
                    <documentation>
Alternative document page images (e.g.
black-and-white)
                </documentation>
            </annotation>
            </element>
            <element name="Border" type="pc:BorderType" minOccurs="0"
maxOccurs="1">

```



```

    </element>
    <element name="PrintSpace" type="pc:PrintSpaceType"
      minOccurs="0" maxOccurs="1">
    </element>
    <element name="ReadingOrder" type="pc:ReadingOrderType"
      minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation></documentation>
      </annotation>
    </element>
    <element name="Layers" type="pc:LayersType" minOccurs="0"
      maxOccurs="1">
      <annotation>
        <documentation>Unassigned regions are
considered to be in the (virtual) default layer which is to be treated as below
any other layers.</documentation>
      </annotation>
    </element>
    <element name="Relations" type="pc:RelationsType" minOccurs="0">
    </element>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="TextRegion"
type="pc:TextRegionType"></element>
      <element name="ImageRegion" type="pc:ImageRegionType">
    </element>
      <element name="LineDrawingRegion"
type="pc:LineDrawingRegionType">
    </element>
      <element name="GraphicRegion"
type="pc:GraphicRegionType">
    </element>
      <element name="TableRegion" type="pc:TableRegionType">
    </element>
      <element name="ChartRegion" type="pc:ChartRegionType">
    </element>
      <element name="SeparatorRegion"
type="pc:SeparatorRegionType">
    </element>
      <element name="MathsRegion" type="pc:MathsRegionType">
    </element>
      <element name="ChemRegion"
type="pc:ChemRegionType"></element>
      <element name="MusicRegion"
type="pc:MusicRegionType"></element>
      <element name="AdvertRegion" type="pc:AdvertRegionType">
    </element>
      <element name="NoiseRegion" type="pc:NoiseRegionType">
    </element>
      <element name="UnknownRegion"
type="pc:UnknownRegionType">
    </element>
    </choice>

  </sequence>
  <attribute name="imageFilename" type="string"
use="required"></attribute>
  <attribute name="imageWidth" type="int" use="required"></attribute>
  <attribute name="imageHeight" type="int"
use="required"></attribute>

```

```

    <attribute name="custom" type="string">
      <annotation>
        <documentation>For generic use</documentation>
      </annotation>
    </attribute>
    <attribute name="type" type="pc:PageTypeSimpleType">
    <annotation>
      <documentation>Page type</documentation>
    </annotation>
    </attribute>
  </complexType>
  <complexType name="TextRegionType">
    <annotation>
      <documentation>
includes
        Pure text is represented as a text region. This
        drop capitals, but practically ornate text may be
        considered as a graphic.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="pc:RegionType">
        <sequence>
          <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded">
          </element>
          <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="1">
          </element>
          <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0" maxOccurs="1">
          </element>
        </sequence>
        <attribute name="orientation" type="float" use="optional">
          <annotation>
            <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
          </annotation>
        </attribute>
        <attribute name="type" type="pc:TextTypeSimpleType"
          use="optional">
          <annotation>
            <documentation>
              The nature of the text in the region
            </documentation>
          </annotation>
        </attribute>
        <attribute name="Leading" type="int" use="optional">
          <annotation>
            <documentation>
lines of
              The degree of space in points between the
              text (line spacing)
            </documentation>
          </annotation>
        </attribute>
        <attribute name="readingDirection"

```

```

        type="pc:ReadingDirectionSimpleType" use="optional">
        <annotation>
            <documentation>
                The direction in which text in a region
should be
                read (within lines)
            </documentation>
        </annotation>
    </attribute>
    <attribute name="readingOrientation" type="float"
        use="optional">
        <annotation>
            <documentation>The angle the baseline of text
withing a region has to be rotated (relative to the rectangle encapsulating the
region) in clockwise direction in order to correct the present skew (negative
values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
    </attribute>
    <attribute name="indented" type="boolean" use="optional">
        <annotation>
            <documentation>
                Defines whether a region of text is
indented or not
            </documentation>
        </annotation>
    </attribute>
    <attribute name="align" type="pc:AlignSimpleType">
        <annotation>
            <documentation>Text align</documentation>
        </annotation>
    </attribute>
    <attribute name="primaryLanguage"
type="pc:LanguageSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The primary language used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="secondaryLanguage"
type="pc:LanguageSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The secondary language used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="primaryScript" type="pc:ScriptSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The primary script used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType"

```

```

        use="optional">
        <annotation>
            <documentation>
                The secondary script used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="production"
type="pc:ProductionSimpleType"></attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="CoordsType">
    <attribute name="points" use="required" type="pc:PointsType">
    <annotation>
        <documentation>Point list with format "x1,y1 x2,y2
..."</documentation>
    </annotation>
    </attribute>
</complexType>
<complexType name="TextLineType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Baseline" type="pc:BaselineType"
            minOccurs="0">
        <annotation>
            <documentation>
                Multiple connected points that mark the
baseline
                of the glyphs
            </documentation>
        </annotation>
    </element>
    <element name="Word" type="pc:WordType" minOccurs="0"
        maxOccurs="unbounded">
    </element>
    <element name="TextEquiv" type="pc:TextEquivType"
        minOccurs="0" maxOccurs="1">
    </element>
    <element name="TextStyle" type="pc:TextStyleType"
minOccurs="0"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="primaryLanguage"
        type="pc:LanguageSimpleType">
    <annotation>
        <documentation>
            Overrides primaryLanguage attribute of parent
text
            region
        </documentation>
    </annotation>
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType">
    <annotation>
        <documentation>
            Overrides the production attribute of the
parent
            text region

```

```

        </documentation>
    </annotation>
</attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="WordType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Glyph" type="pc:GlyphType" minOccurs="0"
            maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
minOccurs="0"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="Language" type="pc:LanguageSimpleType">
        <annotation>
            <documentation>
                Overrides primaryLanguage attribute of parent
line
                and/or text region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType">
        <annotation>
            <documentation>
                Overrides the production attribute of the
parent
                text line and/or text region.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic use</documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="GlyphType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
minOccurs="0"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="Ligature" use="optional" type="boolean">

```

```

    </attribute>
    <attribute name="symbol" use="optional" type="boolean">
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType">
      <annotation>
        <documentation>
          Overrides the production attribute of the
parent
          word / text line / text region.
        </documentation>
      </annotation>
    </attribute>
    <attribute name="custom" type="string">
      <annotation>
        <documentation>For generic use</documentation>
      </annotation>
    </attribute>
    <attribute name="comments" type="string"></attribute>
  </complexType>
  <complexType name="TextEquivType">
    <sequence>
      <element name="PlainText" type="string" minOccurs="0">
        <annotation>
          <documentation>
            Text in a "simple" form (ASCII or
extended ASCII
            as mostly used for typing). I.e. no use
of
            special characters for ligatures (should
be
            stored as two separate characters) etc.
          </documentation>
        </annotation>
      </element>
      <element name="Unicode" type="string">
        <annotation>
          <documentation>
            Correct encoding of the original, always
using
            the corresponding Unicode code point.
I.e.
            ligatures have to be represented as one
            character etc.
          </documentation>
        </annotation>
      </element>
    </sequence>
    <attribute name="conf">
      <annotation>
        <documentation>OCR confidence value (between 0 and
1)</documentation>
      </annotation>
    </simpleType>
    <restriction base="float">
      <minInclusive value="0"></minInclusive>
      <maxInclusive value="1"></maxInclusive>
    </restriction>
  </simpleType>
</attribute>

```

```

</complexType>
<complexType name="ImageRegionType">
  <annotation>
    <documentation>
      An image is considered to be more intricate and
complex
      than a graphic. These can be photos or drawings.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
      </attribute>
      <attribute name="colourDepth"
        type="pc:ColourDepthSimpleType" use="optional">
        <annotation>
          <documentation>
the region
            The colour bit depth required for
            </documentation>
          </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType"
          use="optional">
          <annotation>
            <documentation>
region
              The background colour of the
              </documentation>
            </annotation>
          </attribute>
          <attribute name="embText" type="boolean"
            use="optional">
            <annotation>
              <documentation>
contains
                Specifies whether the region also
                text
                </documentation>
              </annotation>
            </attribute>
          </extension>
        </complexContent>
      </complexType>
    <complexType name="LineDrawingRegionType">
      <annotation>
        <documentation>
          A line drawing is a single colour illustration without
          solid areas.
        </documentation>
      </annotation>
    </complexContent>
  </complexType>

```

```

        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="penColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The pen (foreground) colour of the
region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="embText" type="boolean"
                use="optional">
                <annotation>
                    <documentation>
                        Specifies whether the region also
contains
                        text
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="GraphicRegionType">
    <annotation>
        <documentation>
            Regions containing simple graphics, such as a company
            logo, should be marked as graphic regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>

```



```

        <attribute name="type" use="optional"
            type="pc:GraphicsTypeSimpleType">
            <annotation>
                <documentation>
                    The type of graphic in the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="numColours" type="int"
            use="optional">
            <annotation>
                <documentation>
                    An approximation of the number of
colours
                    used in the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="embText" type="boolean"
            use="optional">
            <annotation>
                <documentation>
                    Specifies whether the region also
contains
                    text.
                </documentation>
            </annotation>
        </attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="TableRegionType">
    <annotation>
        <documentation>
            Tabular data in any form is represented with a table
            region. Rows and columns may or may not have separator
            lines; these lines are not separator regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="rows" type="int" use="optional">
                <annotation>
                    <documentation>
                        The number of rows present in the
table
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="columns" type="int" use="optional">

```

the table	<pre> <annotation> <documentation> The number of columns present in </documentation> </annotation> </attribute> <attribute name="LineColour" type="pc:ColourSimpleType" use="optional"> <annotation> <documentation> The colour of the lines used in </documentation> </annotation> </attribute> <attribute name="bgColour" type="pc:ColourSimpleType" use="optional"> <annotation> <documentation> The background colour of the </documentation> </annotation> </attribute> <attribute name="LineSeparators" type="boolean" use="optional"> <annotation> <documentation> Specifies the presence of line </documentation> </annotation> </attribute> <attribute name="embText" type="boolean" use="optional"> <annotation> <documentation> Specifies whether the region also </documentation> </annotation> </attribute> </extension> </complexContent> </complexType> <complexType name="ChartRegionType"> <annotation> <documentation> Regions containing charts or graphs of any type, </documentation> </annotation> </complexContent> <extension base="pc:RegionType"> <attribute name="orientation" type="float" </pre>

```

        use="optional">
        <annotation>
            <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
    </attribute>
    <attribute name="type" use="optional"
        type="pc:ChartTypeSimpleType">
        <annotation>
            <documentation>
                The type of chart in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="numColours" type="int"
        use="optional">
        <annotation>
            <documentation>
                An approximation of the number of
colours
                used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The background colour of the
region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="embText" type="boolean"
        use="optional">
        <annotation>
            <documentation>
                Specifies whether the region also
contains
                text
            </documentation>
        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
<complexType name="SeparatorRegionType">
    <annotation>
        <documentation>
            Separators are lines that lie between columns and
            paragraphs and can be used to logically separate
            different articles from each other.
        </documentation>
    </annotation>
</complexContent>
    <extension base="pc:RegionType">
        <attribute name="orientation" type="float"

```

```

        use="optional">
        <annotation>
            <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
    </attribute>
    <attribute name="colour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The colour of the separator
            </documentation>
        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
<complexType name="MathsRegionType">
    <annotation>
        <documentation>
            Regions containing equations and mathematical symbols
            should be marked as maths regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="ChemRegionType">
    <annotation>
        <documentation>
            Regions containing chemical formulas.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">

```

```

        <annotation>
            <documentation>
                The angle the rectangle
encapsulating a
                region has to be rotated in
clockwise
                direction in order to correct the
present
                skew (negative values indicate
                anti-clockwise rotation). Range:
                -179.999,180
            </documentation>
        </annotation>
    </attribute>

    <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The background colour of the
region
            </documentation>
        </annotation>
    </attribute>

    </extension>
</complexContent>
</complexType>
<complexType name="MusicRegionType">
    <annotation>
        <documentation>
            Regions containing musical notations.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="AdvertRegionType">
    <annotation>

```

```

        <documentation>
            Regions containing advertisements.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180
                </documentation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <documentation>
                    The background colour of the
region
                </documentation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="NoiseRegionType">
    <annotation>
        <documentation>
            Noise regions are regions where no real data lies,
only
            false data created by artifacts on the document or
            scanner noise.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType"></extension>
    </complexContent>
</complexType>
<complexType name="UnknownRegionType">
    <annotation>
        <documentation>
            To be used if the region type cannot be ascertained.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType"></extension>
    </complexContent>
</complexType>

<complexType name="PrintSpaceType">
    <annotation>
        <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.

```

It does not contain pagenumber (if not part of running title), marginals, signature mark, preview words.

```

</documentation>
  </annotation>
  <sequence>
    <element name="Coords" type="pc:CoordsType"></element>
  </sequence>
</complexType>

<complexType name="ReadingOrderType">
  <annotation>
    <documentation>Definition of the reading order within the page. To
express a reading order between elements they have to be included in an
OrderedGroup. Groups may contain further groups.</documentation>
  </annotation>
  <choice minOccurs="1" maxOccurs="1">
    <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
    <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
  </choice>
</complexType>

<complexType name="RegionRefIndexedType">
  <annotation>
    <documentation>Numbered region</documentation>
  </annotation>
  <attribute name="index" type="int" use="required">
    <annotation>
      <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
    </annotation>
  </attribute>
  <attribute name="regionRef" type="IDREF" use="required"></attribute>
</complexType>

<complexType name="OrderedGroupIndexedType">
  <annotation>
    <documentation>
      Indexed group containing ordered elements
    </documentation>
  </annotation>
  <choice minOccurs="1" maxOccurs="unbounded">
    <element name="RegionRefIndexed"
      type="pc:RegionRefIndexedType">
    </element>
    <element name="OrderedGroupIndexed"
      type="pc:OrderedGroupIndexedType">
    </element>
    <element name="UnorderedGroupIndexed"
      type="pc:UnorderedGroupIndexedType">
    </element>
  </choice>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="index" type="int" use="required">
    <annotation>
      <documentation>
        Position (order number) of this item within the
        current hierarchy level.
      </documentation>
    </annotation>
  </attribute>

```

```

        </attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="UnorderedGroupIndexedType">
        <annotation>
            <documentation>
                Indexed group containing unordered elements
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef" type="pc:RegionRefType"></element>
            <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
            <element name="UnorderedGroup"
                type="pc:UnorderedGroupType">
            </element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>
                    Position (order number) of this item within the
                    current hierarchy level.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="RegionRefType">
        <attribute name="regionRef" type="IDREF"
use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupType">
        <annotation>
            <documentation>
                Numbered group (contains ordered elements)
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRefIndexed"
                type="pc:RegionRefIndexedType">
            </element>
            <element name="OrderedGroupIndexed"
                type="pc:OrderedGroupIndexedType">
            </element>
            <element name="UnorderedGroupIndexed"
                type="pc:UnorderedGroupIndexedType">
            </element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="UnorderedGroupType">
        <annotation>
            <documentation>

```



```

        Numbered group (contains unordered elements)
    </documentation>
</annotation>
<choice minOccurs="1" maxOccurs="unbounded">
    <element name="RegionRef" type="pc:RegionRefType"></element>
    <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
    <element name="UnorderedGroup"
        type="pc:UnorderedGroupType">
    </element>
</choice>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="caption" type="string"></attribute>
</complexType>

<complexType name="BorderType">
    <annotation>
        <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>
<simpleType name="ColourSimpleType">
    <restriction base="string">
        <enumeration value="black"></enumeration>
        <enumeration value="blue"></enumeration>
        <enumeration value="brown"></enumeration>
        <enumeration value="cyan"></enumeration>
        <enumeration value="green"></enumeration>
        <enumeration value="grey"></enumeration>
        <enumeration value="indigo"></enumeration>
        <enumeration value="magenta"></enumeration>
        <enumeration value="orange"></enumeration>
        <enumeration value="pink"></enumeration>
        <enumeration value="red"></enumeration>
        <enumeration value="turquoise"></enumeration>
        <enumeration value="violet"></enumeration>
        <enumeration value="white"></enumeration>
        <enumeration value="yellow"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="ReadingDirectionSimpleType">
    <restriction base="string">
        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
        <enumeration value="top-to-bottom"></enumeration>
        <enumeration value="bottom-to-top"></enumeration>
    </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
    <restriction base="string">
        <enumeration value="paragraph"></enumeration>
        <enumeration value="heading"></enumeration>
        <enumeration value="caption"></enumeration>
        <enumeration value="header"></enumeration>
        <enumeration value="footer"></enumeration>
    </restriction>
</simpleType>

```

```

        <enumeration value="page-number"></enumeration>
        <enumeration value="drop-capital"></enumeration>
        <enumeration value="credit"></enumeration>
        <enumeration value="floating"></enumeration>
        <enumeration value="signature-mark"></enumeration>
        <enumeration value="catch-word"></enumeration>
        <enumeration value="marginalia"></enumeration>
        <enumeration value="footnote"></enumeration>
        <enumeration value="footnote-continued"></enumeration>
        <enumeration value="endnote"></enumeration>
        <enumeration value="TOC-entry"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="PageTypeSimpleType">
    <restriction base="string">
        <enumeration value="front-cover"></enumeration>
        <enumeration value="back-cover"></enumeration>
        <enumeration value="title"></enumeration>
        <enumeration value="table-of-contents"></enumeration>
        <enumeration value="index"></enumeration>
        <enumeration value="content"></enumeration>
        <enumeration value="blank"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="LanguageSimpleType">
    <restriction base="string">
<enumeration value="Abkhaz"></enumeration>
<enumeration value="Afar"></enumeration>
<enumeration value="Afrikaans"></enumeration>
<enumeration value="Akan"></enumeration>
<enumeration value="Albanian"></enumeration>
<enumeration value="Amharic"></enumeration>
<enumeration value="Arabic"></enumeration>
<enumeration value="Aragonese"></enumeration>
<enumeration value="Armenian"></enumeration>
<enumeration value="Assamese"></enumeration>
<enumeration value="Avaric"></enumeration>
<enumeration value="Avestan"></enumeration>
<enumeration value="Aymara"></enumeration>
<enumeration value="Azerbaijani"></enumeration>
<enumeration value="Bambara"></enumeration>
<enumeration value="Bashkir"></enumeration>
<enumeration value="Basque"></enumeration>
<enumeration value="Belarusian"></enumeration>
<enumeration value="Bengali"></enumeration>
<enumeration value="Bihari"></enumeration>
<enumeration value="Bislama"></enumeration>
<enumeration value="Bosnian"></enumeration>
<enumeration value="Breton"></enumeration>
<enumeration value="Bulgarian"></enumeration>
<enumeration value="Burmese"></enumeration>
<enumeration value="Cambodian"></enumeration>
<enumeration value="Cantonese"></enumeration>
<enumeration value="Catalan"></enumeration>
<enumeration value="Chamorro"></enumeration>
<enumeration value="Chechen"></enumeration>

```

```

<enumeration value="Chichewa"></enumeration>
<enumeration value="Chinese"></enumeration>
<enumeration value="Chuvash"></enumeration>
<enumeration value="Cornish"></enumeration>
<enumeration value="Corsican"></enumeration>
<enumeration value="Cree"></enumeration>
<enumeration value="Croatian"></enumeration>
<enumeration value="Czech"></enumeration>
<enumeration value="Danish"></enumeration>
<enumeration value="Divehi"></enumeration>
<enumeration value="Dutch"></enumeration>
<enumeration value="Dzongkha"></enumeration>
<enumeration value="English"></enumeration>
<enumeration value="Esperanto"></enumeration>
<enumeration value="Estonian"></enumeration>
<enumeration value="Ewe"></enumeration>
<enumeration value="Faroese"></enumeration>
<enumeration value="Fijian"></enumeration>
<enumeration value="Finnish"></enumeration>
<enumeration value="French"></enumeration>
<enumeration value="Fula"></enumeration>
<enumeration value="Gaelic"></enumeration>
<enumeration value="Galician"></enumeration>
<enumeration value="Ganda"></enumeration>
<enumeration value="Georgian"></enumeration>
<enumeration value="German"></enumeration>
<enumeration value="Greek"></enumeration>
<enumeration value="Guarani"></enumeration>
<enumeration value="Gujarati"></enumeration>
<enumeration value="Haitian"></enumeration>
<enumeration value="Hausa"></enumeration>
<enumeration value="Hebrew"></enumeration>
<enumeration value="Herero"></enumeration>
<enumeration value="Hindi"></enumeration>
<enumeration value="Hiri Motu"></enumeration>
<enumeration value="Hungarian"></enumeration>
<enumeration value="Icelandic"></enumeration>
<enumeration value="Ido"></enumeration>
<enumeration value="Igbo"></enumeration>
<enumeration value="Indonesian"></enumeration>
<enumeration value="Interlingua"></enumeration>
<enumeration value="Interlingue"></enumeration>
<enumeration value="Inuktitut"></enumeration>
<enumeration value="Inupiaq"></enumeration>
<enumeration value="Irish"></enumeration>
<enumeration value="Italian"></enumeration>
<enumeration value="Japanese"></enumeration>
<enumeration value="Javanese"></enumeration>
<enumeration value="Kalaallisut"></enumeration>
<enumeration value="Kannada"></enumeration>
<enumeration value="Kanuri"></enumeration>
<enumeration value="Kashmiri"></enumeration>
<enumeration value="Kazakh"></enumeration>
<enumeration value="Khmer"></enumeration>
<enumeration value="Kikuyu"></enumeration>
<enumeration value="Kinyarwanda"></enumeration>
<enumeration value="Kirundi"></enumeration>
<enumeration value="Komi"></enumeration>
<enumeration value="Kongo"></enumeration>

```

```

<enumeration value="Korean"></enumeration>
<enumeration value="Kurdish"></enumeration>
<enumeration value="Kwanyama"></enumeration>
<enumeration value="Kyrgyz"></enumeration>
<enumeration value="Lao"></enumeration>
<enumeration value="Latin"></enumeration>
<enumeration value="Latvian"></enumeration>
<enumeration value="Limburgish"></enumeration>
<enumeration value="Lingala"></enumeration>
<enumeration value="Lithuanian"></enumeration>
<enumeration value="Luba-Katanga"></enumeration>
<enumeration value="Luxembourgish"></enumeration>
<enumeration value="Macedonian"></enumeration>
<enumeration value="Malagasy"></enumeration>
<enumeration value="Malay"></enumeration>
<enumeration value="Malayalam"></enumeration>
<enumeration value="Maltese"></enumeration>
<enumeration value="Manx"></enumeration>
<enumeration value="Māori"></enumeration>
<enumeration value="Marathi"></enumeration>
<enumeration value="Marshallese"></enumeration>
<enumeration value="Mongolian"></enumeration>
<enumeration value="Nauru"></enumeration>
<enumeration value="Navajo"></enumeration>
<enumeration value="Ndonga"></enumeration>
<enumeration value="Nepali"></enumeration>
<enumeration value="North Ndebele"></enumeration>
<enumeration value="Northern Sami"></enumeration>
<enumeration value="Norwegian"></enumeration>
<enumeration value="Norwegian Bokmål"></enumeration>
<enumeration value="Norwegian Nynorsk"></enumeration>
<enumeration value="Nuosu"></enumeration>
<enumeration value="Occitan"></enumeration>
<enumeration value="Ojibwe"></enumeration>
<enumeration value="Old Church Slavonic"></enumeration>
<enumeration value="Oriya"></enumeration>
<enumeration value="Oromo"></enumeration>
<enumeration value="Ossetian"></enumeration>
<enumeration value="Pāli"></enumeration>
<enumeration value="Panjabi"></enumeration>
<enumeration value="Pashto"></enumeration>
<enumeration value="Persian"></enumeration>
<enumeration value="Polish"></enumeration>
<enumeration value="Portuguese"></enumeration>
<enumeration value="Punjabi"></enumeration>
<enumeration value="Quechua"></enumeration>
<enumeration value="Romanian"></enumeration>
<enumeration value="Romansh"></enumeration>
<enumeration value="Russian"></enumeration>
<enumeration value="Samoan"></enumeration>
<enumeration value="Sango"></enumeration>
<enumeration value="Sanskrit"></enumeration>
<enumeration value="Sardinian"></enumeration>
<enumeration value="Serbian"></enumeration>
<enumeration value="Shona"></enumeration>
<enumeration value="Sindhi"></enumeration>
<enumeration value="Sinhala"></enumeration>
<enumeration value="Slovak"></enumeration>
<enumeration value="Slovene"></enumeration>

```

```

<enumeration value="Somali"></enumeration>
<enumeration value="South Ndebele"></enumeration>
<enumeration value="Southern Sotho"></enumeration>
<enumeration value="Spanish"></enumeration>
<enumeration value="Sundanese"></enumeration>
<enumeration value="Swahili"></enumeration>
<enumeration value="Swati"></enumeration>
<enumeration value="Swedish"></enumeration>
<enumeration value="Tagalog"></enumeration>
<enumeration value="Tahitian"></enumeration>
<enumeration value="Tajik"></enumeration>
<enumeration value="Tamil"></enumeration>
<enumeration value="Tatar"></enumeration>
<enumeration value="Telugu"></enumeration>
<enumeration value="Thai"></enumeration>
<enumeration value="Tibetan"></enumeration>
<enumeration value="Tigrinya"></enumeration>
<enumeration value="Tonga"></enumeration>
<enumeration value="Tsonga"></enumeration>
<enumeration value="Tswana"></enumeration>
<enumeration value="Turkish"></enumeration>
<enumeration value="Turkmen"></enumeration>
<enumeration value="Twi"></enumeration>
<enumeration value="Uighur"></enumeration>
<enumeration value="Ukrainian"></enumeration>
<enumeration value="Urdu"></enumeration>
<enumeration value="Uzbek"></enumeration>
<enumeration value="Venda"></enumeration>
<enumeration value="Vietnamese"></enumeration>
<enumeration value="Volapük"></enumeration>
<enumeration value="Walloon"></enumeration>
<enumeration value="Welsh"></enumeration>
<enumeration value="Western Frisian"></enumeration>
<enumeration value="WoLoF"></enumeration>
<enumeration value="Xhosa"></enumeration>
<enumeration value="Yiddish"></enumeration>
<enumeration value="Yoruba"></enumeration>
<enumeration value="Zhuang"></enumeration>
<enumeration value="Zulu"></enumeration>
<enumeration value="other"></enumeration>
</restriction>
</simpleType>

<simpleType name="ScriptSimpleType">
  <restriction base="string">
    <enumeration value="Arabic"></enumeration>
    <enumeration value="Bengali"></enumeration>
    <enumeration value="Chinese-simplified"></enumeration>
    <enumeration value="Chinese-traditional"></enumeration>
    <enumeration value="Cyrillic"></enumeration>
    <enumeration value="Devangari"></enumeration>
    <enumeration value="Ethiopic"></enumeration>
    <enumeration value="Greek"></enumeration>
    <enumeration value="Gujarati"></enumeration>
    <enumeration value="Gurmukhi"></enumeration>
    <enumeration value="Hebrew"></enumeration>
    <enumeration value="Latin"></enumeration>
    <enumeration value="Thai"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>

```

```

    </restriction>
</simpleType>
<simpleType name="ColourDepthSimpleType">
  <restriction base="string">
    <enumeration value="bilevel"></enumeration>
    <enumeration value="greyscale"></enumeration>
    <enumeration value="colour"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="GraphicsTypeSimpleType">
  <restriction base="string">
    <enumeration value="Logo"></enumeration>
    <enumeration value="Letterhead"></enumeration>
    <enumeration value="decoration"></enumeration>
    <enumeration value="frame"></enumeration>
    <enumeration value="handwritten-annotation"></enumeration>
    <enumeration value="stamp"></enumeration>
    <enumeration value="signature"></enumeration>
    <enumeration value="barcode"></enumeration>
    <enumeration value="paper-grow"></enumeration>
    <enumeration value="punch-hole"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="ChartTypeSimpleType">
  <restriction base="string">
    <enumeration value="bar"></enumeration>
    <enumeration value="line"></enumeration>
    <enumeration value="pie"></enumeration>
    <enumeration value="scatter"></enumeration>
    <enumeration value="surface"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<complexType name="LayersType">
  <annotation>
    <documentation>
      Can be used to express the z-index of overlapping
      regions. An element with a greater z-index is always in
      front of another element with lower z-index.
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Layer" type="pc:LayerType"></element>
  </sequence>
</complexType>

<complexType name="LayerType">
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="RegionRef" type="pc:RegionRefType"></element>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="zIndex" type="int" use="required"></attribute>
  <attribute name="caption" type="string"></attribute>
</complexType>

```

```

<complexType name="BaselineType">
  <attribute name="points" type="pc:PointsType" use="required"></attribute>
</complexType>

<simpleType name="PointsType">
  <annotation>
    <documentation>Point list with format "x1,y1 x2,y2
..."</documentation>
  </annotation>
  <restriction base="string">
    <pattern value="([0-9]+,[0-9]+ )+([0-9]+,[0-9]+)"></pattern>
  </restriction>
</simpleType>

<complexType name="RelationsType">
  <annotation>
    <documentation>
      Container for one-to-one relations between layout
      objects (for example: DropCap - paragraph, caption -
      image)
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Relation" type="pc:RelationType"></element>
  </sequence>
</complexType>

<complexType name="RelationType">
  <annotation>
    <documentation>
      One-to-one relation between to layout object. Use 'link'
      for loose relations and 'join' for strong relations
      (where something is fragmented for instance).

      Examples for 'link': caption - image floating -
      paragraph paragraph - paragraph (when a paragraph is
      split across columns and the last word of the first
      paragraph DOES NOT continue in the second paragraph)
      drop-cap - paragraph (when the drop-cap is a whole word)

      Examples for 'join': word - word (separated word at the
      end of a line) drop-cap - paragraph (when the drop-cap
      is not a whole word) paragraph - paragraph (when a
      paragraph is split across columns and the last word of
      the first paragraph DOES continue in the second
      paragraph)
    </documentation>
  </annotation>
  <sequence minOccurs="2" maxOccurs="2">
    <element name="RegionRef" type="pc:RegionRefType"></element>
  </sequence>
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="string">
        <enumeration value="Link"></enumeration>
        <enumeration value="Join"></enumeration>
      </restriction>
    </simpleType>
  </attribute>

```



```

        </simpleType>
    </attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic
use</documentation></annotation></attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>

<simpleType name="ProductionSimpleType">
    <annotation>
        <documentation>Text production type</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="printed"></enumeration>
        <enumeration value="typewritten"></enumeration>
        <enumeration value="handwritten-cursive"></enumeration>
        <enumeration value="handwritten-printschrift"></enumeration>
        <enumeration value="medieval-manuscript"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<complexType name="TextStyleType">
    <annotation>
        <documentation>
            Monospace (fixed-pitch, non-proportional) or
            proportional font
        </documentation>
    </annotation>
    <attribute name="fontFamily" type="string">
        <annotation>
            <documentation>
                For instance: Arial, Times New Roman. Add more
                information if necessary (e.g. blackletter,
                antiqua).
            </documentation>
        </annotation>
    </attribute>
    <attribute name="serif" type="boolean">
        <annotation>
            <documentation>
                Serif or sans-serif typeface
            </documentation>
        </annotation>
    </attribute>
    <attribute name="monospace" type="boolean"></attribute>
    <attribute name="fontSize" type="float">
        <annotation>
            <documentation>
                The size of the characters in points
            </documentation>
        </annotation>
    </attribute>
    <attribute name="kerning" type="int">
        <annotation>
            <documentation>
                The degree of space (in points) between the
                characters in a string of text
            </documentation>
        </annotation>
    </attribute>

```



```

        </documentation>
    </annotation>
</attribute>
<attribute name="textColour" type="pc:ColourSimpleType"></attribute>
<attribute name="bgColour" type="pc:ColourSimpleType">
    <annotation>
        <documentation>Background colour</documentation>
    </annotation>
</attribute>
<attribute name="reverseVideo" type="boolean">
    <annotation>
        <documentation>
            Specifies whether the colour of the text appears
            reversed against a background colour
        </documentation>
    </annotation>
</attribute>
<attribute name="bold" type="boolean"></attribute>
<attribute name="italic" type="boolean"></attribute>
<attribute name="underlined" type="boolean"></attribute>
<attribute name="subscript" type="boolean"></attribute>
<attribute name="superscript" type="boolean"></attribute>
<attribute name="striketrough" type="boolean"></attribute>
    <attribute name="smallCaps" type="boolean"></attribute>
    <attribute name="LetterSpaced" type="boolean"></attribute>
</complexType>

<complexType name="RegionType" abstract="true">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="TextRegion"
type="pc:TextRegionType"></element>
            <element name="ImageRegion"
type="pc:ImageRegionType"></element>
            <element name="LineDrawingRegion"
                type="pc:LineDrawingRegionType">
            </element>
            <element name="GraphicRegion"
                type="pc:GraphicRegionType">
            </element>
            <element name="TableRegion"
type="pc:TableRegionType"></element>
            <element name="ChartRegion"
type="pc:ChartRegionType"></element>
            <element name="SeparatorRegion"
                type="pc:SeparatorRegionType">
            </element>
            <element name="MathsRegion"
type="pc:MathsRegionType"></element>
            <element name="ChemRegion"
type="pc:ChemRegionType"></element>
            <element name="MusicRegion"
type="pc:MusicRegionType"></element>
            <element name="AdvertRegion" type="pc:AdvertRegionType">
            </element>
            <element name="NoiseRegion" type="pc:NoiseRegionType"></element>
            <element name="UnknownRegion"
type="pc:UnknownRegionType"></element>
        </choice>
    </sequence>
</complexType>

```

```

        </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic use</documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="AlternativeImageType">
    <attribute name="filename" type="string" use="required"></attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>

<simpleType name="AlignSimpleType">
    <restriction base="string">
        <enumeration value="left"></enumeration>
        <enumeration value="centre"></enumeration>
        <enumeration value="right"></enumeration>
        <enumeration value="justify"></enumeration>
    </restriction>
</simpleType>
</schema>

```

Pagecontent Format Specification – Fifth Release

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2016-07-
15"
    elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2016-07-
15">

    <element name="PcGts" type="pc:PcGtsType">
        <annotation>
            <documentation>Page Content - Ground Truth and
Storage</documentation>
        </annotation></element>

    <complexType name="PcGtsType">
        <sequence>
            <element name="Metadata" type="pc:MetadataType"></element>
            <element name="Page" type="pc:PageType"></element>
        </sequence>
        <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
        <sequence>
            <element name="Creator" type="string"></element>
            <element name="Created" type="dateTime">
                <annotation>
                    <documentation>The timestamp has to be in UTC
(Coordinated Universal Time) and not local
time.</documentation></annotation></element>
            <element name="LastChange" type="dateTime">
                <annotation>
                    <documentation>The timestamp has to be in UTC
(Coordinated Universal Time) and not local
time.</documentation></annotation></element>
            <element name="Comments" type="string" minOccurs="0"
maxOccurs="1"></element>
        </sequence>
        <attribute name="externalRef" type="string" use="optional">
            <annotation>
                <documentation>External reference of any
kind</documentation>
            </annotation></attribute>
    </complexType>

    <complexType name="PageType">
        <sequence>
            <element name="AlternativeImage"
type="pc:AlternativeImageType" minOccurs="0"
maxOccurs="unbounded">
                <annotation>
                    <documentation>
                        Alternative document page images (e.g.
                        black-and-white)
                    </documentation>
                </annotation>
            </element>
        </sequence>
    </complexType>
</schema>
```

```

        </documentation>
    </annotation>
</element>
<element name="Border" type="pc:BorderType" minOccurs="0"
    maxOccurs="1">
</element>
<element name="PrintSpace" type="pc:PrintSpaceType"
    minOccurs="0" maxOccurs="1">
</element>
<element name="ReadingOrder" type="pc:ReadingOrderType"
    minOccurs="0" maxOccurs="1">
    <annotation>
        <documentation></documentation>
    </annotation>
</element>
<element name="Layers" type="pc:LayersType" minOccurs="0"
    maxOccurs="1">
    <annotation>
        <documentation>Unassigned regions are
considered to be in the (virtual) default layer which is to be treated as below
any other layers.</documentation>
    </annotation>
</element>
<element name="Relations" type="pc:RelationsType" minOccurs="0">
</element>
<choice minOccurs="0" maxOccurs="unbounded">
    <element name="TextRegion"
type="pc:TextRegionType"></element>
    <element name="ImageRegion" type="pc:ImageRegionType">
</element>
    <element name="LineDrawingRegion"
        type="pc:LineDrawingRegionType">
</element>
    <element name="GraphicRegion"
        type="pc:GraphicRegionType">
</element>
    <element name="TableRegion" type="pc:TableRegionType">
</element>
    <element name="ChartRegion" type="pc:ChartRegionType">
</element>
    <element name="SeparatorRegion"
        type="pc:SeparatorRegionType">
</element>
    <element name="MathsRegion" type="pc:MathsRegionType">
</element>
    <element name="ChemRegion"
type="pc:ChemRegionType"></element>
    <element name="MusicRegion"
type="pc:MusicRegionType"></element>
    <element name="AdvertRegion" type="pc:AdvertRegionType">
</element>
    <element name="NoiseRegion" type="pc:NoiseRegionType">
</element>
    <element name="UnknownRegion"
        type="pc:UnknownRegionType">
</element>
</choice>
</sequence>

```

```

        <attribute name="imageFilename" type="string"
use="required"></attribute>
        <attribute name="imageWidth" type="int" use="required"></attribute>
        <attribute name="imageHeight" type="int"
use="required"></attribute>
        <attribute name="custom" type="string">
            <annotation>
                <documentation>For generic use</documentation>
            </annotation>
        </attribute>
        <attribute name="type" type="pc:PageTypeSimpleType">
<annotation>
        <documentation>Page type</documentation>
</annotation>
        </attribute>
        <attribute name="primaryLanguage" type="pc:LanguageSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The primary language used in the page (lower-
level definitions override the page-level definition)
                </documentation>
            </annotation>
        </attribute>
        <attribute name="secondaryLanguage" type="pc:LanguageSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The secondary language used in the page (lower-
level definitions override the page-level definition)
                </documentation>
            </annotation>
        </attribute>
        <attribute name="primaryScript" type="pc:ScriptSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The primary script used in the page (lower-
level definitions override the page-level definition)
                </documentation>
            </annotation>
        </attribute>
        <attribute name="secondaryScript" type="pc:ScriptSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The secondary script used in the page (lower-
level definitions override the page-level definition)
                </documentation>
            </annotation>
        </attribute>
        <attribute name="readingDirection"
            type="pc:ReadingDirectionSimpleType" use="optional">
            <annotation>
                <documentation>
                    The direction in which text in a region should
be
                    read (within lines) (lower-level definitions
override the page-level definition)
                </documentation>
            </annotation>
        </attribute>
    </element>

```

```

        </documentation>
      </annotation>
    </attribute>
    <attribute name="textLineOrder"
      type="pc:TextLineOrderSimpleType" use="optional">
      <annotation>
        <documentation>Inner-block order of text lines (in
addition to "readingDirection" which is the inner-text line order of words and
characters) (lower-level definitions override the page-level
definition)</documentation>
      </annotation>
    </attribute>
  </complexType>
  <complexType name="TextRegionType">
    <annotation>
      <documentation>
        Pure text is represented as a text region. This
includes
        drop capitals, but practically ornate text may be
        considered as a graphic.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="pc:RegionType">
        <sequence>
          <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded">
          </element>
          <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
          </element>
          <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0" maxOccurs="1">
          </element>
        </sequence>
        <attribute name="orientation" type="float" use="optional">
          <annotation>
            <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
          </annotation>
        </attribute>
        <attribute name="type" type="pc:TextTypeSimpleType"
          use="optional">
          <annotation>
            <documentation>
              The nature of the text in the region
            </documentation>
          </annotation>
        </attribute>
        <attribute name="leading" type="int" use="optional">
          <annotation>
            <documentation>
              The degree of space in points between the
lines of
              text (line spacing)
            </documentation>
          </annotation>

```

```

    </attribute>
    <attribute name="readingDirection"
      type="pc:ReadingDirectionSimpleType" use="optional">
      <annotation>
        <documentation>
          The direction in which text in a region
should be
          read (within lines)
        </documentation>
      </annotation>
    </attribute>
    <attribute name="textLineOrder"
      type="pc:TextLineOrderSimpleType" use="optional">
      <annotation>
        <documentation>Inner-block order of text lines
(in addition to "readingDirection" which is the inner-text line order of words
and characters)</documentation>
      </annotation>
    </attribute>
    <attribute name="readingOrientation" type="float"
      use="optional">
      <annotation>
        <documentation>The angle the baseline of text
withing a region has to be rotated (relative to the rectangle encapsulating the
region) in clockwise direction in order to correct the present skew (negative
values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
      </annotation>
    </attribute>
    <attribute name="indented" type="boolean" use="optional">
      <annotation>
        <documentation>
          Defines whether a region of text is
indented or not
        </documentation>
      </annotation>
    </attribute>
    <attribute name="align" type="pc:AlignSimpleType">
      <annotation>
        <documentation>Text align</documentation>
      </annotation>
    </attribute>
    <attribute name="primaryLanguage"
type="pc:LanguageSimpleType"
      use="optional">
      <annotation>
        <documentation>
          The primary language used in the region
        </documentation>
      </annotation>
    </attribute>
    <attribute name="secondaryLanguage"
type="pc:LanguageSimpleType"
      use="optional">
      <annotation>
        <documentation>
          The secondary language used in the region
        </documentation>
      </annotation>
    </attribute>

```

```

        </attribute>
        <attribute name="primaryScript" type="pc:ScriptSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The primary script used in the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="secondaryScript" type="pc:ScriptSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The secondary script used in the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="production" type="pc:ProductionSimpleType"
use="optional"></attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="CoordsType">
    <attribute name="points" use="required" type="pc:PointsType">
    <annotation>
        <documentation>Point list with format "x1,y1 x2,y2
..."</documentation>
    </annotation>
    </attribute>
</complexType>
<complexType name="TextLineType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Baseline" type="pc:BaselineType"
            minOccurs="0">
            <annotation>
                <documentation>
baseline
                    Multiple connected points that mark the
                        of the glyphs
                </documentation>
            </annotation>
        </element>
        <element name="Word" type="pc:WordType" minOccurs="0"
            maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
minOccurs="0"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="primaryLanguage"
        type="pc:LanguageSimpleType">
    <annotation>
        <documentation>
text
            Overrides primaryLanguage attribute of parent

```



```

        region
        </documentation>
    </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The primary script used in the text line
        </documentation>
    </annotation>
</attribute>
<attribute name="secondaryScript" type="pc:ScriptSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The secondary script used in the text line
        </documentation>
    </annotation>
</attribute>
<attribute name="readingDirection"
    type="pc:ReadingDirectionSimpleType" use="optional">
    <annotation>
        <documentation>
            The direction in which text in a text line
should be read
        </documentation>
    </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType"
use="optional">
    <annotation>
        <documentation>
            Overrides the production attribute of the
parent
            text region
        </documentation>
    </annotation>
</attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="WordType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Glyph" type="pc:GlyphType" minOccurs="0"
            maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
minOccurs="0"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>

```

```

<attribute name="language" type="pc:LanguageSimpleType">
  <annotation>
    <documentation>
      Overrides primaryLanguage attribute of parent
line
      and/or text region
    </documentation>
  </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The primary script used in the word
    </documentation>
  </annotation>
</attribute>
<attribute name="secondaryScript" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The secondary script used in the word
    </documentation>
  </annotation>
</attribute>
<attribute name="readingDirection"
  type="pc:ReadingDirectionSimpleType" use="optional">
  <annotation>
    <documentation>
      The direction in which characters in a word
should be read
    </documentation>
  </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType"
use="optional">
  <annotation>
    <documentation>
      Overrides the production attribute of the
parent
      text line and/or text region.
    </documentation>
  </annotation>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic use</documentation>
  </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="GlyphType">
  <sequence>
    <element name="Coords" type="pc:CoordsType"></element>
    <element name="TextEquiv" type="pc:TextEquivType"
      minOccurs="0" maxOccurs="unbounded">
    </element>
    <element name="TextStyle" type="pc:TextStyleType"
minOccurs="0"></element>

```

```

</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="ligature" use="optional" type="boolean">
</attribute>
<attribute name="symbol" use="optional" type="boolean">
</attribute>
<attribute name="script" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The script used for the glyph
    </documentation>
  </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType"
  use="optional">
  <annotation>
    <documentation>
      Overrides the production attribute of the
parent
      word / text line / text region.
    </documentation>
  </annotation>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic use</documentation>
  </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="TextEquivType">
  <sequence>
    <element name="PlainText" type="string" minOccurs="0">
      <annotation>
        <documentation>
          Text in a "simple" form (ASCII or
extended ASCII
          as mostly used for typing). I.e. no use
of
          special characters for ligatures (should
be
          stored as two separate characters) etc.
        </documentation>
      </annotation>
    </element>
    <element name="Unicode" type="string">
      <annotation>
        <documentation>
          Correct encoding of the original, always
using
          the corresponding Unicode code point.
I.e.
          ligatures have to be represented as one
          character etc.
        </documentation>
      </annotation>
    </element>
  </sequence>

```

```

        <attribute name="index" use="optional">
            <annotation>
                <documentation>Used for sort order in case multiple
TextEquivs are defined. The text content with the lowest index should be
interpreted as the main text content.</documentation>
            </annotation>
        </simpleType>
        <restriction base="integer">
            <minInclusive value="0"></minInclusive>
        </restriction>
    </simpleType>
</attribute>
<attribute name="conf">
<annotation>
    <documentation>OCR confidence value (between 0 and
1)</documentation>
</annotation>
<simpleType>
    <restriction base="float">
        <minInclusive value="0"></minInclusive>
        <maxInclusive value="1"></maxInclusive>
    </restriction>
</simpleType>
</attribute>
    <attribute name="dataType" type="pc:TextDataTypeSimpleType"
use="optional">
        <annotation>
            <documentation>Type of text content (is it free
text or a number, for instance)
This is only a descriptive attribute, the text type is not checked during XML
validation</documentation>
        </annotation>
    </attribute>
    <attribute name="dataTypeDetails" type="string" use="optional">
        <annotation>
            <documentation>Refinement for dataType
attribute. Can be a regular expression, for instance.</documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string" use="optional"></attribute>
</complexType>
<complexType name="ImageRegionType">
    <annotation>
        <documentation>
            An image is considered to be more intricate and
complex
            than a graphic. These can be photos or drawings.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>

```

```

        </attribute>
        <attribute name="colourDepth"
            type="pc:ColourDepthSimpleType" use="optional">
            <annotation>
                <documentation>
                    The colour bit depth required for
the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The background colour of the
region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="embText" type="boolean"
            use="optional">
            <annotation>
                <documentation>
                    Specifies whether the region also
contains
                    text
                </documentation>
            </annotation>
        </attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="LineDrawingRegionType">
    <annotation>
        <documentation>
            A line drawing is a single colour illustration without
            solid areas.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="penColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The pen (foreground) colour of the
region
                    </documentation>
                </annotation>
            </attribute>

```

```

        <attribute name="bgColour" type="pc:ColourSimpleType"
            use="optional">
            <annotation>
                <documentation>
                    The background colour of the
region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="embText" type="boolean"
            use="optional">
            <annotation>
                <documentation>
                    Specifies whether the region also
contains
                    text
                </documentation>
            </annotation>
        </attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="GraphicRegionType">
    <annotation>
        <documentation>
            Regions containing simple graphics, such as a company
            logo, should be marked as graphic regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="type" use="optional"
                type="pc:GraphicsTypeSimpleType">
                <annotation>
                    <documentation>
                        The type of graphic in the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="numColours" type="int"
                use="optional">
                <annotation>
                    <documentation>
                        An approximation of the number of
colours
                        used in the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="embText" type="boolean"

```

```

        use="optional">
        <annotation>
            <documentation>
                Specifies whether the region also
contains
                text.
            </documentation>
        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
<complexType name="TableRegionType">
    <annotation>
        <documentation>
            Tabular data in any form is represented with a table
            region. Rows and columns may or may not have separator
            lines; these lines are not separator regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
            use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="rows" type="int" use="optional">
                <annotation>
                    <documentation>
                        The number of rows present in the
table
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="columns" type="int" use="optional">
                <annotation>
                    <documentation>
                        The number of columns present in
the table
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="LineColour"
type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The colour of the lines used in
the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
            use="optional">

```

```

        <annotation>
            <documentation>
                The background colour of the
region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="LineSeparators" type="boolean"
        use="optional">
        <annotation>
            <documentation>
                Specifies the presence of line
separators
            </documentation>
        </annotation>
    </attribute>
    <attribute name="embText" type="boolean"
        use="optional">
        <annotation>
            <documentation>
                Specifies whether the region also
contains
                text
            </documentation>
        </annotation>
    </attribute>
</extension>
</complexType>
<complexType name="ChartRegionType">
    <annotation>
        <documentation>
            Regions containing charts or graphs of any type,
should
            be marked as chart regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="type" use="optional"
                type="pc:ChartTypeSimpleType">
                <annotation>
                    <documentation>
                        The type of chart in the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="numColours" type="int"
                use="optional">
                <annotation>

```



```

        <documentation>
            An approximation of the number of
colours        used in the region
        </documentation>
    </annotation>
</attribute>
<attribute name="bgColour" type="pc:ColourSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The background colour of the
region
        </documentation>
    </annotation>
</attribute>
<attribute name="embText" type="boolean"
    use="optional">
    <annotation>
        <documentation>
            Specifies whether the region also
contains
            text
        </documentation>
    </annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="SeparatorRegionType">
    <annotation>
        <documentation>
            Separators are lines that lie between columns and
            paragraphs and can be used to logically separate
            different articles from each other.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="colour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The colour of the separator
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>

```

```

<complexType name="MathsRegionType">
  <annotation>
    <documentation>
      Regions containing equations and mathematical symbols
      should be marked as maths regions.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
        </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>
            The background colour of the
region
          </documentation>
        </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
<complexType name="ChemRegionType">
  <annotation>
    <documentation>
      Regions containing chemical formulas.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>
            The angle the rectangle
encapsulating a
region has to be rotated in
clockwise
direction in order to correct the
present
skew (negative values indicate
anti-clockwise rotation). Range:
-179.999,180
          </documentation>
        </annotation>
        </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>

```

```

region
The background colour of the
    </documentation>
    </annotation>
    </attribute>

    </extension>
    </complexContent>
</complexType>
<complexType name="MusicRegionType">
    <annotation>
        <documentation>
            Regions containing musical notations.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="AdvertRegionType">
    <annotation>
        <documentation>
            Regions containing advertisements.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180
                </documentation>
            </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>

```

```

region
    <documentation>
        The background colour of the
    </documentation>
</annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="NoiseRegionType">
    <annotation>
        <documentation>
            Noise regions are regions where no real data lies,
only
            false data created by artifacts on the document or
            scanner noise.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType"></extension>
    </complexContent>
</complexType>
<complexType name="UnknownRegionType">
    <annotation>
        <documentation>
            To be used if the region type cannot be ascertained.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType"></extension>
    </complexContent>
</complexType>

<complexType name="PrintSpaceType">
    <annotation>
        <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.
It does not contain pagenumber (if not part of running title), marginals,
signature mark, preview words.
</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>

<complexType name="ReadingOrderType">
    <annotation>
        <documentation>Definition of the reading order within the page. To
express a reading order between elements they have to be included in an
OrderedGroup. Groups may contain further groups.</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="1">
        <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>

```

```

        </choice>
    </complexType>

    <complexType name="RegionRefIndexedType">
        <annotation>
            <documentation>Numbered region</documentation>
        </annotation>
        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
            </annotation>
        </attribute>
        <attribute name="regionRef" type="IDREF" use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupIndexedType">
        <annotation>
            <documentation>
                Indexed group containing ordered elements
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRefIndexed"
                type="pc:RegionRefIndexedType">
            </element>
            <element name="OrderedGroupIndexed"
                type="pc:OrderedGroupIndexedType">
            </element>
            <element name="UnorderedGroupIndexed"
                type="pc:UnorderedGroupIndexedType">
            </element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>
                    Position (order number) of this item within the
                    current hierarchy level.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="UnorderedGroupIndexedType">
        <annotation>
            <documentation>
                Indexed group containing unordered elements
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef" type="pc:RegionRefType"></element>
            <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
            <element name="UnorderedGroup"
                type="pc:UnorderedGroupType">
            </element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>

```

```

        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>
                    Position (order number) of this item within the
                    current hierarchy level.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="RegionRefType">
        <attribute name="regionRef" type="IDREF"
use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupType">
        <annotation>
            <documentation>
                Numbered group (contains ordered elements)
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRefIndexed"
                type="pc:RegionRefIndexedType">
            </element>
            <element name="OrderedGroupIndexed"
                type="pc:OrderedGroupIndexedType">
            </element>
            <element name="UnorderedGroupIndexed"
                type="pc:UnorderedGroupIndexedType">
            </element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="UnorderedGroupType">
        <annotation>
            <documentation>
                Numbered group (contains unordered elements)
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef" type="pc:RegionRefType"></element>
            <element name="OrderedGroup"
type="pc:OrderedGroupType"></element>
            <element name="UnorderedGroup"
                type="pc:UnorderedGroupType">
            </element>
        </choice>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="caption" type="string"></attribute>
    </complexType>

    <complexType name="BorderType">
        <annotation>
            <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>

```

```

    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>
<simpleType name="ColourSimpleType">
    <restriction base="string">
        <enumeration value="black"></enumeration>
        <enumeration value="blue"></enumeration>
        <enumeration value="brown"></enumeration>
        <enumeration value="cyan"></enumeration>
        <enumeration value="green"></enumeration>
        <enumeration value="grey"></enumeration>
        <enumeration value="indigo"></enumeration>
        <enumeration value="magenta"></enumeration>
        <enumeration value="orange"></enumeration>
        <enumeration value="pink"></enumeration>
        <enumeration value="red"></enumeration>
        <enumeration value="turquoise"></enumeration>
        <enumeration value="violet"></enumeration>
        <enumeration value="white"></enumeration>
        <enumeration value="yellow"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="ReadingDirectionSimpleType">
    <restriction base="string">
        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
        <enumeration value="top-to-bottom"></enumeration>
        <enumeration value="bottom-to-top"></enumeration>
    </restriction>
</simpleType>
<simpleType name="TextLineOrderSimpleType">
    <restriction base="string">
        <enumeration value="top-to-bottom"></enumeration>
        <enumeration value="bottom-to-top"></enumeration>
        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
    </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
    <restriction base="string">
        <enumeration value="paragraph"></enumeration>
        <enumeration value="heading"></enumeration>
        <enumeration value="caption"></enumeration>
        <enumeration value="header"></enumeration>
        <enumeration value="footer"></enumeration>
        <enumeration value="page-number"></enumeration>
        <enumeration value="drop-capital"></enumeration>
        <enumeration value="credit"></enumeration>
        <enumeration value="floating"></enumeration>
        <enumeration value="signature-mark"></enumeration>
        <enumeration value="catch-word"></enumeration>
        <enumeration value="marginalia"></enumeration>
        <enumeration value="footnote"></enumeration>
        <enumeration value="footnote-continued"></enumeration>
        <enumeration value="endnote"></enumeration>
        <enumeration value="TOC-entry"></enumeration>
    </restriction>
</simpleType>

```

```

        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="PageTypeSimpleType">
    <restriction base="string">
        <enumeration value="front-cover"></enumeration>
        <enumeration value="back-cover"></enumeration>
        <enumeration value="title"></enumeration>
        <enumeration value="table-of-contents"></enumeration>
        <enumeration value="index"></enumeration>
        <enumeration value="content"></enumeration>
        <enumeration value="blank"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="LanguageSimpleType">
    <annotation>
        <documentation>iso15924 2016-07-14</documentation>
    </annotation>
    <restriction base="string">
<enumeration value="Abkhaz"></enumeration>
<enumeration value="Afar"></enumeration>
<enumeration value="Afrikaans"></enumeration>
<enumeration value="Akan"></enumeration>
<enumeration value="Albanian"></enumeration>
<enumeration value="Amharic"></enumeration>
<enumeration value="Arabic"></enumeration>
<enumeration value="Aragonese"></enumeration>
<enumeration value="Armenian"></enumeration>
<enumeration value="Assamese"></enumeration>
<enumeration value="Avaric"></enumeration>
<enumeration value="Avestan"></enumeration>
<enumeration value="Aymara"></enumeration>
<enumeration value="Azerbaijani"></enumeration>
<enumeration value="Bambara"></enumeration>
<enumeration value="Bashkir"></enumeration>
<enumeration value="Basque"></enumeration>
<enumeration value="Belarusian"></enumeration>
<enumeration value="Bengali"></enumeration>
<enumeration value="Bihari"></enumeration>
<enumeration value="Bislama"></enumeration>
<enumeration value="Bosnian"></enumeration>
<enumeration value="Breton"></enumeration>
<enumeration value="Bulgarian"></enumeration>
<enumeration value="Burmese"></enumeration>
<enumeration value="Cambodian"></enumeration>
<enumeration value="Cantonese"></enumeration>
<enumeration value="Catalan"></enumeration>
<enumeration value="Chamorro"></enumeration>
<enumeration value="Chechen"></enumeration>
<enumeration value="Chichewa"></enumeration>
<enumeration value="Chinese"></enumeration>
<enumeration value="Chuvash"></enumeration>
<enumeration value="Cornish"></enumeration>
<enumeration value="Corsican"></enumeration>
<enumeration value="Cree"></enumeration>
<enumeration value="Croatian"></enumeration>
<enumeration value="Czech"></enumeration>

```



```

<enumeration value="Danish"></enumeration>
<enumeration value="Divehi"></enumeration>
<enumeration value="Dutch"></enumeration>
<enumeration value="Dzongkha"></enumeration>
<enumeration value="English"></enumeration>
<enumeration value="Esperanto"></enumeration>
<enumeration value="Estonian"></enumeration>
<enumeration value="Ewe"></enumeration>
<enumeration value="Faroese"></enumeration>
<enumeration value="Fijian"></enumeration>
<enumeration value="Finnish"></enumeration>
<enumeration value="French"></enumeration>
<enumeration value="Fula"></enumeration>
<enumeration value="Gaelic"></enumeration>
<enumeration value="Galician"></enumeration>
<enumeration value="Ganda"></enumeration>
<enumeration value="Georgian"></enumeration>
<enumeration value="German"></enumeration>
<enumeration value="Greek"></enumeration>
<enumeration value="Guarani"></enumeration>
<enumeration value="Gujarati"></enumeration>
<enumeration value="Haitian"></enumeration>
<enumeration value="Hausa"></enumeration>
<enumeration value="Hebrew"></enumeration>
<enumeration value="Herero"></enumeration>
<enumeration value="Hindi"></enumeration>
<enumeration value="Hiri Motu"></enumeration>
<enumeration value="Hungarian"></enumeration>
<enumeration value="Icelandic"></enumeration>
<enumeration value="Ido"></enumeration>
<enumeration value="Igbo"></enumeration>
<enumeration value="Indonesian"></enumeration>
<enumeration value="Interlingua"></enumeration>
<enumeration value="Interlingue"></enumeration>
<enumeration value="Inuktitut"></enumeration>
<enumeration value="Inupiaq"></enumeration>
<enumeration value="Irish"></enumeration>
<enumeration value="Italian"></enumeration>
<enumeration value="Japanese"></enumeration>
<enumeration value="Javanese"></enumeration>
<enumeration value="Kalaallisut"></enumeration>
<enumeration value="Kannada"></enumeration>
<enumeration value="Kanuri"></enumeration>
<enumeration value="Kashmiri"></enumeration>
<enumeration value="Kazakh"></enumeration>
<enumeration value="Khmer"></enumeration>
<enumeration value="Kikuyu"></enumeration>
<enumeration value="Kinyarwanda"></enumeration>
<enumeration value="Kirundi"></enumeration>
<enumeration value="Komi"></enumeration>
<enumeration value="Kongo"></enumeration>
<enumeration value="Korean"></enumeration>
<enumeration value="Kurdish"></enumeration>
<enumeration value="Kwanyama"></enumeration>
<enumeration value="Kyrgyz"></enumeration>
<enumeration value="Lao"></enumeration>
<enumeration value="Latin"></enumeration>
<enumeration value="Latvian"></enumeration>
<enumeration value="Limburgish"></enumeration>

```

```

<enumeration value="Lingala"></enumeration>
<enumeration value="Lithuanian"></enumeration>
<enumeration value="Luba-Katanga"></enumeration>
<enumeration value="Luxembourgish"></enumeration>
<enumeration value="Macedonian"></enumeration>
<enumeration value="Malagasy"></enumeration>
<enumeration value="Malay"></enumeration>
<enumeration value="Malayalam"></enumeration>
<enumeration value="Maltese"></enumeration>
<enumeration value="Manx"></enumeration>
<enumeration value="Māori"></enumeration>
<enumeration value="Marathi"></enumeration>
<enumeration value="Marshallese"></enumeration>
<enumeration value="Mongolian"></enumeration>
<enumeration value="Nauru"></enumeration>
<enumeration value="Navajo"></enumeration>
<enumeration value="Ndonga"></enumeration>
<enumeration value="Nepali"></enumeration>
<enumeration value="North Ndebele"></enumeration>
<enumeration value="Northern Sami"></enumeration>
<enumeration value="Norwegian"></enumeration>
<enumeration value="Norwegian Bokmål"></enumeration>
<enumeration value="Norwegian Nynorsk"></enumeration>
<enumeration value="Nuosu"></enumeration>
<enumeration value="Occitan"></enumeration>
<enumeration value="Ojibwe"></enumeration>
<enumeration value="Old Church Slavonic"></enumeration>
<enumeration value="Oriya"></enumeration>
<enumeration value="Oromo"></enumeration>
<enumeration value="Ossetian"></enumeration>
<enumeration value="Pāli"></enumeration>
<enumeration value="Panjabi"></enumeration>
<enumeration value="Pashto"></enumeration>
<enumeration value="Persian"></enumeration>
<enumeration value="Polish"></enumeration>
<enumeration value="Portuguese"></enumeration>
<enumeration value="Punjabi"></enumeration>
<enumeration value="Quechua"></enumeration>
<enumeration value="Romanian"></enumeration>
<enumeration value="Romansh"></enumeration>
<enumeration value="Russian"></enumeration>
<enumeration value="Samoan"></enumeration>
<enumeration value="Sango"></enumeration>
<enumeration value="Sanskrit"></enumeration>
<enumeration value="Sardinian"></enumeration>
<enumeration value="Serbian"></enumeration>
<enumeration value="Shona"></enumeration>
<enumeration value="Sindhi"></enumeration>
<enumeration value="Sinhala"></enumeration>
<enumeration value="Slovak"></enumeration>
<enumeration value="Slovene"></enumeration>
<enumeration value="Somali"></enumeration>
<enumeration value="South Ndebele"></enumeration>
<enumeration value="Southern Sotho"></enumeration>
<enumeration value="Spanish"></enumeration>
<enumeration value="Sundanese"></enumeration>
<enumeration value="Swahili"></enumeration>
<enumeration value="Swati"></enumeration>
<enumeration value="Swedish"></enumeration>

```

```

<enumeration value="Tagalog"></enumeration>
<enumeration value="Tahitian"></enumeration>
<enumeration value="Tajik"></enumeration>
<enumeration value="Tamil"></enumeration>
<enumeration value="Tatar"></enumeration>
<enumeration value="Telugu"></enumeration>
<enumeration value="Thai"></enumeration>
<enumeration value="Tibetan"></enumeration>
<enumeration value="Tigrinya"></enumeration>
<enumeration value="Tonga"></enumeration>
<enumeration value="Tsonga"></enumeration>
<enumeration value="Tswana"></enumeration>
<enumeration value="Turkish"></enumeration>
<enumeration value="Turkmen"></enumeration>
<enumeration value="Twi"></enumeration>
<enumeration value="Uighur"></enumeration>
<enumeration value="Ukrainian"></enumeration>
<enumeration value="Urdu"></enumeration>
<enumeration value="Uzbek"></enumeration>
<enumeration value="Venda"></enumeration>
<enumeration value="Vietnamese"></enumeration>
<enumeration value="Volapük"></enumeration>
<enumeration value="Walloon"></enumeration>
<enumeration value="Welsh"></enumeration>
<enumeration value="Western Frisian"></enumeration>
<enumeration value="Wolof"></enumeration>
<enumeration value="Xhosa"></enumeration>
<enumeration value="Yiddish"></enumeration>
<enumeration value="Yoruba"></enumeration>
<enumeration value="Zhuang"></enumeration>
<enumeration value="Zulu"></enumeration>
<enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<simpleType name="ScriptSimpleType">
  <restriction base="string">
<enumeration value="Adlm - Adlam"></enumeration>
<enumeration value="Afak - Afaka"></enumeration>
<enumeration value="Aghb - Caucasian Albanian"></enumeration>
<enumeration value="Ahom - Ahom, Tai Ahom"></enumeration>
<enumeration value="Arab - Arabic"></enumeration>
<enumeration value="Aran - Arabic (Nastaliq variant)"></enumeration>
<enumeration value="Armi - Imperial Aramaic"></enumeration>
<enumeration value="Armn - Armenian"></enumeration>
<enumeration value="Avst - Avestan"></enumeration>
<enumeration value="Bali - Balinese"></enumeration>
<enumeration value="Bamu - Bamum"></enumeration>
<enumeration value="Bass - Bassa Vah"></enumeration>
<enumeration value="Batk - Batak"></enumeration>
<enumeration value="Beng - Bengali"></enumeration>
<enumeration value="Bhks - Bhaiksuki"></enumeration>
<enumeration value="Blis - Blissymbols"></enumeration>
<enumeration value="Bopo - Bopomofo"></enumeration>
<enumeration value="Brah - Brahmi"></enumeration>
<enumeration value="Brai - Braille"></enumeration>
<enumeration value="Bugi - Buginese"></enumeration>
<enumeration value="Buhd - Buhid"></enumeration>
<enumeration value="Cakm - Chakma"></enumeration>

```

```

<enumeration value="Cans - Unified Canadian Aboriginal Syllabics"></enumeration>
<enumeration value="Cari - Carian"></enumeration>
<enumeration value="Cham - Cham"></enumeration>
<enumeration value="Cher - Cherokee"></enumeration>
<enumeration value="Cirt - Cirth"></enumeration>
<enumeration value="Copt - Coptic"></enumeration>
<enumeration value="Cprt - Cypriot"></enumeration>
<enumeration value="Cyrl - Cyrillic"></enumeration>
<enumeration value="Cyrs - Cyrillic (Old Church Slavonic
variant)"></enumeration>
<enumeration value="Deva - Devanagari (Nagari)"></enumeration>
<enumeration value="Dsrt - Deseret (Mormon)"></enumeration>
<enumeration value="Dupl - Duplicoyan shorthand, Duplicoyan
stenography"></enumeration>
<enumeration value="Egyd - Egyptian demotic"></enumeration>
<enumeration value="Egyh - Egyptian hieratic"></enumeration>
<enumeration value="Egyp - Egyptian hieroglyphs"></enumeration>
<enumeration value="Elba - Elbasan"></enumeration>
<enumeration value="Ethi - Ethiopic"></enumeration>
<enumeration value="Geok - Khutsuri (Asomtavruli and Nuskhuri)"></enumeration>
<enumeration value="Geor - Georgian (Mkhedruli)"></enumeration>
<enumeration value="Glag - Glagolitic"></enumeration>
<enumeration value="Goth - Gothic"></enumeration>
<enumeration value="Gran - Grantha"></enumeration>
<enumeration value="GreK - Greek"></enumeration>
<enumeration value="Gujr - Gujarati"></enumeration>
<enumeration value="Guru - Gurmukhi"></enumeration>
<enumeration value="Hanb - Han with Bopomofo"></enumeration>
<enumeration value="Hang - Hangul"></enumeration>
<enumeration value="Hani - Han (Hanzi, Kanji, Hanja)"></enumeration>
<enumeration value="Hano - Hanunoo (Hanunóo)"></enumeration>
<enumeration value="Hans - Han (Simplified variant)"></enumeration>
<enumeration value="Hant - Han (Traditional variant)"></enumeration>
<enumeration value="Hatr - Hatran"></enumeration>
<enumeration value="Hebr - Hebrew"></enumeration>
<enumeration value="Hira - Hiragana"></enumeration>
<enumeration value="HLuw - Anatolian Hieroglyphs"></enumeration>
<enumeration value="Hmng - Pahawh Hmong"></enumeration>
<enumeration value="Hrkt - Japanese syllabaries"></enumeration>
<enumeration value="Hung - Old Hungarian (Hungarian Runic)"></enumeration>
<enumeration value="Inds - Indus (Harappan)"></enumeration>
<enumeration value="Ital - Old Italic (Etruscan, Oscan etc.)"></enumeration>
<enumeration value="Jamo - Jamo"></enumeration>
<enumeration value="Java - Javanese"></enumeration>
<enumeration value="Jpan - Japanese"></enumeration>
<enumeration value="Jurc - Jurchen"></enumeration>
<enumeration value="Kali - Kayah Li"></enumeration>
<enumeration value="Kana - Katakana"></enumeration>
<enumeration value="Khar - Kharoshthi"></enumeration>
<enumeration value="Khmr - Khmer"></enumeration>
<enumeration value="Khoj - Khojki"></enumeration>
<enumeration value="Kitl - Khitan large script"></enumeration>
<enumeration value="Kits - Khitan small script"></enumeration>
<enumeration value="Knda - Kannada"></enumeration>
<enumeration value="Kore - Korean (alias for Hangul + Han)"></enumeration>
<enumeration value="Kpel - Kpelle"></enumeration>
<enumeration value="Kthi - Kaithi"></enumeration>
<enumeration value="Lana - Tai Tham (Lanna)"></enumeration>
<enumeration value="Laoo - Lao"></enumeration>

```

```

<enumeration value="Latf - Latin (Fraktur variant)"></enumeration>
<enumeration value="Latg - Latin (Gaelic variant)"></enumeration>
<enumeration value="Latn - Latin"></enumeration>
<enumeration value="Leke - Leke"></enumeration>
<enumeration value="Lepc - Lepcha (Róng)"></enumeration>
<enumeration value="Limb - Limbu"></enumeration>
<enumeration value="Lina - Linear A"></enumeration>
<enumeration value="Linb - Linear B"></enumeration>
<enumeration value="Lisu - Lisu (Fraser)"></enumeration>
<enumeration value="Loma - Loma"></enumeration>
<enumeration value="Lyci - Lycian"></enumeration>
<enumeration value="Lydi - Lydian"></enumeration>
<enumeration value="Mahj - Mahajani"></enumeration>
<enumeration value="Mand - Mandaic, Mandaean"></enumeration>
<enumeration value="Mani - Manichaeen"></enumeration>
<enumeration value="Marc - Marchen"></enumeration>
<enumeration value="Maya - Mayan hieroglyphs"></enumeration>
<enumeration value="Mend - Mende Kikakui"></enumeration>
<enumeration value="Merc - Meroitic Cursive"></enumeration>
<enumeration value="Mero - Meroitic Hieroglyphs"></enumeration>
<enumeration value="Mlym - Malayalam"></enumeration>
<enumeration value="Modi - Modi, Modī"></enumeration>
<enumeration value="Mong - Mongolian"></enumeration>
<enumeration value="Moon - Moon (Moon code, Moon script, Moon
type)"></enumeration>
<enumeration value="Mroo - Mro, Mru"></enumeration>
<enumeration value="Mtei - Meitei Mayek (Meithei, Meetei)"></enumeration>
<enumeration value="Mult - Multani"></enumeration>
<enumeration value="Mymr - Myanmar (Burmese)"></enumeration>
<enumeration value="Narb - Old North Arabian (Ancient North
Arabian)"></enumeration>
<enumeration value="Nbat - Nabataean"></enumeration>
<enumeration value="Newa - Newa, Newar, Newari"></enumeration>
<enumeration value="Nkgb - Nakhi Geba"></enumeration>
<enumeration value="Nkoo - N'Ko"></enumeration>
<enumeration value="Nshu - Nüshu"></enumeration>
<enumeration value="Ogam - Ogham"></enumeration>
<enumeration value="Olck - Ol Chiki (Ol Cemet', Ol, Santali)"></enumeration>
<enumeration value="Orkh - Old Turkic, Orkhon Runic"></enumeration>
<enumeration value="Orya - Oriya"></enumeration>
<enumeration value="Osge - Osage"></enumeration>
<enumeration value="Osma - Osmanya"></enumeration>
<enumeration value="Palm - Palmyrene"></enumeration>
<enumeration value="Pauc - Pau Cin Hau"></enumeration>
<enumeration value="Perm - Old Permic"></enumeration>
<enumeration value="Phag - Phags-pa"></enumeration>
<enumeration value="Phli - Inscriptional Pahlavi"></enumeration>
<enumeration value="Phlp - Psalter Pahlavi"></enumeration>
<enumeration value="Phlv - Book Pahlavi"></enumeration>
<enumeration value="Phnx - Phoenician"></enumeration>
<enumeration value="Piqd - Klingon (KLI pIqaD)"></enumeration>
<enumeration value="PLrd - Miao (Pollard)"></enumeration>
<enumeration value="Prti - Inscriptional Parthian"></enumeration>
<enumeration value="Rjng - Rejang (Redjang, Kaganga)"></enumeration>
<enumeration value="Roro - Rongorongo"></enumeration>
<enumeration value="Runr - Runic"></enumeration>
<enumeration value="Samr - Samaritan"></enumeration>
<enumeration value="Sara - Sarati"></enumeration>
<enumeration value="Sarb - Old South Arabian"></enumeration>

```



```

<enumeration value="Saur - Saurashtra"></enumeration>
<enumeration value="Sgnw - SignWriting"></enumeration>
<enumeration value="Shaw - Shavian (Shaw)"></enumeration>
<enumeration value="Shrd - Sharada, Śārādā"></enumeration>
<enumeration value="Sidd - Siddham"></enumeration>
<enumeration value="Sind - Khudawadi, Sindhi"></enumeration>
<enumeration value="Sinh - Sinhala"></enumeration>
<enumeration value="Sora - Sora Sompeng"></enumeration>
<enumeration value="Sund - Sundanese"></enumeration>
<enumeration value="Sylo - Syloti Nagri"></enumeration>
<enumeration value="Syrc - Syriac"></enumeration>
<enumeration value="Syre - Syriac (Estrangelo variant)"></enumeration>
<enumeration value="Syrj - Syriac (Western variant)"></enumeration>
<enumeration value="Syrn - Syriac (Eastern variant)"></enumeration>
<enumeration value="Tagb - Tagbanwa"></enumeration>
<enumeration value="Takr - Takri"></enumeration>
<enumeration value="Tale - Tai Le"></enumeration>
<enumeration value="Talu - New Tai Lue"></enumeration>
<enumeration value="Taml - Tamil"></enumeration>
<enumeration value="Tang - Tangut"></enumeration>
<enumeration value="Tavt - Tai Viet"></enumeration>
<enumeration value="Telu - Telugu"></enumeration>
<enumeration value="Teng - Tengwar"></enumeration>
<enumeration value="Tfng - Tifinagh (Berber)"></enumeration>
<enumeration value="Tglg - Tagalog (Baybayin, Alibata)"></enumeration>
<enumeration value="Thaa - Thaana"></enumeration>
<enumeration value="Thai - Thai"></enumeration>
<enumeration value="Tibt - Tibetan"></enumeration>
<enumeration value="Tirh - Tirhuta"></enumeration>
<enumeration value="Ugar - Ugaritic"></enumeration>
<enumeration value="Vaii - Vai"></enumeration>
<enumeration value="Visp - Visible Speech"></enumeration>
<enumeration value="Wara - Warang Citi (Varang Kshiti)"></enumeration>
<enumeration value="Wole - Woleai"></enumeration>
<enumeration value="Xpeo - Old Persian"></enumeration>
<enumeration value="Xsux - Cuneiform, Sumero-Akkadian"></enumeration>
<enumeration value="Yiii - Yi"></enumeration>
<enumeration value="Zinh - Code for inherited script"></enumeration>
<enumeration value="Zmth - Mathematical notation"></enumeration>
<enumeration value="Zsye - Symbols (Emoji variant)"></enumeration>
<enumeration value="Zsym - Symbols"></enumeration>
<enumeration value="Zxxx - Code for unwritten documents"></enumeration>
<enumeration value="Zyyy - Code for undetermined script"></enumeration>
<enumeration value="Zzzz - Code for uncoded script"></enumeration>
<enumeration value="other"></enumeration>
</restriction>
</simpleType>
<simpleType name="ColourDepthSimpleType">
  <restriction base="string">
    <enumeration value="bilevel"></enumeration>
    <enumeration value="greyscale"></enumeration>
    <enumeration value="colour"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="GraphicsTypeSimpleType">
  <restriction base="string">
    <enumeration value="Logo"></enumeration>
    <enumeration value="Letterhead"></enumeration>
  </restriction>
</simpleType>

```

```

        <enumeration value="decoration"></enumeration>
        <enumeration value="frame"></enumeration>
        <enumeration value="handwritten-annotation"></enumeration>
        <enumeration value="stamp"></enumeration>
        <enumeration value="signature"></enumeration>
        <enumeration value="barcode"></enumeration>
        <enumeration value="paper-grow"></enumeration>
        <enumeration value="punch-hole"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="ChartTypeSimpleType">
    <restriction base="string">
        <enumeration value="bar"></enumeration>
        <enumeration value="line"></enumeration>
        <enumeration value="pie"></enumeration>
        <enumeration value="scatter"></enumeration>
        <enumeration value="surface"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<complexType name="LayersType">
    <annotation>
        <documentation>
            Can be used to express the z-index of overlapping
            regions. An element with a greater z-index is always in
            front of another element with lower z-index.
        </documentation>
    </annotation>
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="Layer" type="pc:LayerType"></element>
    </sequence>
</complexType>

<complexType name="LayerType">
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="zIndex" type="int" use="required"></attribute>
    <attribute name="caption" type="string"></attribute>
</complexType>

<complexType name="BaselineType">
    <attribute name="points" type="pc:PointsType" use="required"></attribute>
</complexType>

<simpleType name="PointsType">
    <annotation>
        <documentation>Point list with format "x1,y1 x2,y2
...</documentation>
    </annotation>
    <restriction base="string">
        <pattern value="([0-9]+,[0-9]+ )+([0-9]+,[0-9]+)"></pattern>
    </restriction>

```

```

</simpleType>

<complexType name="RelationsType">
  <annotation>
    <documentation>
      Container for one-to-one relations between layout
      objects (for example: DropCap - paragraph, caption -
      image)
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Relation" type="pc:RelationType"></element>
  </sequence>
</complexType>

<complexType name="RelationType">
  <annotation>
    <documentation>
      One-to-one relation between to layout object. Use 'link'
      for loose relations and 'join' for strong relations
      (where something is fragmented for instance).

      Examples for 'link': caption - image floating -
      paragraph paragraph - paragraph (when a paragraph is
      split across columns and the last word of the first
      paragraph DOES NOT continue in the second paragraph)
      drop-cap - paragraph (when the drop-cap is a whole word)

      Examples for 'join': word - word (separated word at the
      end of a line) drop-cap - paragraph (when the drop-cap
      is not a whole word) paragraph - paragraph (when a
      paragraph is split across columns and the last word of
      the first paragraph DOES continue in the second
      paragraph)
    </documentation>
  </annotation>
  <sequence minOccurs="2" maxOccurs="2">
    <element name="RegionRef" type="pc:RegionRefType"></element>
  </sequence>
  <attribute name="type" use="required">
    <simpleType>
      <restriction base="string">
        <enumeration value="link"></enumeration>
        <enumeration value="join"></enumeration>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="custom" type="string">
    <annotation>
      <documentation>For generic
use</documentation></annotation></attribute>
    <attribute name="comments" type="string"></attribute>
  </complexType>

<simpleType name="ProductionSimpleType">
  <annotation>
    <documentation>Text production type</documentation>
  </annotation>
  <restriction base="string">

```



```

        <enumeration value="printed"></enumeration>
        <enumeration value="typewritten"></enumeration>
        <enumeration value="handwritten-cursive"></enumeration>
        <enumeration value="handwritten-printschrift"></enumeration>
        <enumeration value="medieval-manuscript"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<complexType name="TextStyleType">
    <annotation>
        <documentation>
            Monospace (fixed-pitch, non-proportional) or
            proportional font
        </documentation>
    </annotation>
    <attribute name="fontFamily" type="string">
        <annotation>
            <documentation>
                For instance: Arial, Times New Roman. Add more
                information if necessary (e.g. blackletter,
                antiqua).
            </documentation>
        </annotation>
    </attribute>
    <attribute name="serif" type="boolean">
        <annotation>
            <documentation>
                Serif or sans-serif typeface
            </documentation>
        </annotation>
    </attribute>
    <attribute name="monospace" type="boolean"></attribute>
    <attribute name="fontSize" type="float">
        <annotation>
            <documentation>
                The size of the characters in points
            </documentation>
        </annotation>
    </attribute>
    <attribute name="xHeight" type="integer" use="optional">
        <annotation>
            <documentation>The x-height or corpus size refers to the
distance between the baseline and the mean line of lower-case letters in a
typeface. The unit is assumed to be pixels.</documentation>
        </annotation>
    </attribute>
    <attribute name="kerning" type="int">
        <annotation>
            <documentation>
                The degree of space (in points) between the
                characters in a string of text
            </documentation>
        </annotation>
    </attribute>
    <attribute name="textColour" type="pc:ColourSimpleType"></attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType">
        <annotation>
            <documentation>Background colour</documentation>

```

```

        </annotation>
    </attribute>
    <attribute name="reverseVideo" type="boolean">
        <annotation>
            <documentation>
                Specifies whether the colour of the text appears
                reversed against a background colour
            </documentation>
        </annotation>
    </attribute>
    <attribute name="bold" type="boolean"></attribute>
    <attribute name="italic" type="boolean"></attribute>
    <attribute name="underlined" type="boolean"></attribute>
    <attribute name="subscript" type="boolean"></attribute>
    <attribute name="superscript" type="boolean"></attribute>
    <attribute name="strikethrough" type="boolean"></attribute>
    <attribute name="smallCaps" type="boolean"></attribute>
    <attribute name="LetterSpaced" type="boolean"></attribute>
</complexType>

<complexType name="RegionType" abstract="true">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="TextRegion"
type="pc:TextRegionType"></element>
            <element name="ImageRegion"
type="pc:ImageRegionType"></element>
            <element name="LineDrawingRegion"
type="pc:LineDrawingRegionType"></element>
            <element name="GraphicRegion"
type="pc:GraphicRegionType"></element>
            <element name="TableRegion"
type="pc:TableRegionType"></element>
            <element name="ChartRegion"
type="pc:ChartRegionType"></element>
            <element name="SeparatorRegion"
type="pc:SeparatorRegionType"></element>
            <element name="MathsRegion"
type="pc:MathsRegionType"></element>
            <element name="ChemRegion"
type="pc:ChemRegionType"></element>
            <element name="MusicRegion"
type="pc:MusicRegionType"></element>
            <element name="AdvertRegion" type="pc:AdvertRegionType"></element>
            <element name="NoiseRegion" type="pc:NoiseRegionType"></element>
            <element name="UnknownRegion"
type="pc:UnknownRegionType"></element>
        </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic use</documentation>
        </annotation>
    </attribute>

```

```

    </attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="AlternativeImageType">
  <attribute name="filename" type="string" use="required"></attribute>
  <attribute name="comments" type="string"></attribute>
</complexType>

<simpleType name="AlignSimpleType">
  <restriction base="string">
    <enumeration value="left"></enumeration>
    <enumeration value="centre"></enumeration>
    <enumeration value="right"></enumeration>
    <enumeration value="justify"></enumeration>
  </restriction>
</simpleType>

<simpleType name="TextDataTypeSimpleType">
  <restriction base="string">
    <enumeration value="xsd:decimal">
      <annotation>
        <documentation>Examples: "123.456", "+1234.456", "-
1234.456", "-.456", "-456"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:float">
      <annotation>
        <documentation>Examples: "123.456", "+1234.456", "-
1.2344e56", "-.45E-6", "INF", "-INF", "NaN"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:integer">
      <annotation>
        <documentation>Examples: "123456", "+00000012", "-1",
"-456"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:boolean">
      <annotation>
        <documentation>Examples: "true", "false", "1",
"0"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:date">
      <annotation>
        <documentation>Examples: "2001-10-26", "2001-10-
26+02:00", "2001-10-26Z", "2001-10-26+00:00", "-2001-10-26", "-20000-04-
01"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:time">
      <annotation>
        <documentation>Examples: "21:32:52", "21:32:52+02:00",
"19:32:52Z", "19:32:52+00:00", "21:32:52.12679"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:dateTime">
      <annotation>
        <documentation>Examples: "2001-10-26T21:32:52", "2001-
10-26T21:32:52+02:00", "2001-10-26T19:32:52Z", "2001-10-26T19:32:52+00:00", "-
2001-10-26T21:32:52", "2001-10-26T21:32:52.12679"</documentation>
      </annotation>
    </enumeration>
    <enumeration value="xsd:string">
      <annotation>
        <documentation>Generic text string</documentation>
      </annotation>
    </enumeration>
  </restriction>
</simpleType>

```

```

        </annotation></enumeration>
    <enumeration value="other">
        <annotation>
            <documentation>An XSD type that is not listed or a
custom type (use dataTypeDetails attribute)</documentation>
        </annotation></enumeration>
    </restriction>
</simpleType>
</schema>

```

Pagecontent Format Specification – Sixth Release

```

<?xml version="1.0" encoding="UTF-8" ?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2017-07-
15"
    elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2017-07-
15">

    <element name="PcGts" type="pc:PcGtsType">
        <annotation>
            <documentation>Page Content - Ground Truth and
Storage</documentation>
        </annotation></element>

    <complexType name="PcGtsType">
        <sequence>
            <element name="Metadata" type="pc:MetadataType"></element>
            <element name="Page" type="pc:PageType"></element>
        </sequence>
        <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
        <sequence>
            <element name="Creator" type="string"></element>
            <element name="Created" type="dateTime">
                <annotation>
                    <documentation>
                        The timestamp has to be in UTC
                        (Coordinated
                        Universal Time) and not local time.
                    </documentation>
                </annotation>
            </element>
            <element name="LastChange" type="dateTime">
                <annotation>
                    <documentation>
                        The timestamp has to be in UTC
                        (Coordinated
                        Universal Time) and not local time.
                    </documentation>
                </annotation>
            </element>
            <element name="Comments" type="string" minOccurs="0"
                maxOccurs="1">
            </element>
            <element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
        </sequence>
        <attribute name="externalRef" type="string" use="optional">
            <annotation>
                <documentation>External reference of any
kind</documentation>
            </annotation></attribute>
    </complexType>

```

```

<complexType name="PageType">
  <sequence>
    <element name="AlternativeImage"
      type="pc:AlternativeImageType" minOccurs="0"
      maxOccurs="unbounded">
      <annotation>
        <documentation>
          Alternative document page images (e.g.
          black-and-white)
        </documentation>
      </annotation>
    </element>
    <element name="Border" type="pc:BorderType" minOccurs="0"
      maxOccurs="1">
    </element>
    <element name="PrintSpace" type="pc:PrintSpaceType"
      minOccurs="0" maxOccurs="1">
    </element>
    <element name="ReadingOrder" type="pc:ReadingOrderType"
      minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation></documentation>
      </annotation>
    </element>
    <element name="Layers" type="pc:LayersType" minOccurs="0"
      maxOccurs="1">
      <annotation>
        <documentation>
          Unassigned regions are considered to be
          (virtual) default layer which is to be
          as below any other layers.
        </documentation>
      </annotation>
    </element>
    <element name="Relations" type="pc:RelationsType"
      minOccurs="0">
    </element>
    <element name="UserDefined" type="pc:UserDefinedType" minOccurs="0"
      maxOccurs="1"></element>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="TextRegion"
        type="pc:TextRegionType"></element>
      <element name="ImageRegion" type="pc:ImageRegionType">
      </element>
      <element name="LineDrawingRegion"
        type="pc:LineDrawingRegionType">
      </element>
      <element name="GraphicRegion"
        type="pc:GraphicRegionType">
      </element>
      <element name="TableRegion" type="pc:TableRegionType">
      </element>
      <element name="ChartRegion" type="pc:ChartRegionType">
      </element>
      <element name="SeparatorRegion"
        type="pc:SeparatorRegionType">
      </element>
    </choice>
  </sequence>

```

in the

treated

```

        <element name="MathsRegion" type="pc:MathsRegionType">
        </element>
        <element name="ChemRegion"
type="pc:ChemRegionType"></element>
        <element name="MusicRegion"
type="pc:MusicRegionType"></element>
        <element name="AdvertRegion"
            type="pc:AdvertRegionType">
        </element>
        <element name="NoiseRegion" type="pc:NoiseRegionType">
        </element>
        <element name="UnknownRegion"
            type="pc:UnknownRegionType">
        </element>
    </choice>

</sequence>
<attribute name="imageFilename" type="string"
use="required"></attribute>
<attribute name="imageWidth" type="int" use="required"></attribute>
<attribute name="imageHeight" type="int"
use="required"></attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="type" type="pc:PageTypeSimpleType">
<annotation>
    <documentation>Page type</documentation>
</annotation>
</attribute>
<attribute name="primaryLanguage" type="pc:LanguageSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The primary language used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>
<attribute name="secondaryLanguage" type="pc:LanguageSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The secondary language used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The primary script used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>

```

```

    <attribute name="secondaryScript" type="pc:ScriptSimpleType"
      use="optional">
      <annotation>
        <documentation>
          The secondary script used in the page (lower-
level definitions override the page-level definition)
        </documentation>
      </annotation>
    </attribute>
    <attribute name="readingDirection"
      type="pc:ReadingDirectionSimpleType" use="optional">
      <annotation>
        <documentation>
          The direction in which text in a region should
be
          read (within lines) (lower-level definitions
override the page-level definition)
        </documentation>
      </annotation>
    </attribute>
    <attribute name="textLineOrder"
      type="pc:TextLineOrderSimpleType" use="optional">
      <annotation>
        <documentation>Inner-block order of text lines (in
addition to "readingDirection" which is the inner-text line order of words and
characters) (lower-level definitions override the page-level
definition)</documentation>
      </annotation>
    </attribute>
  </complexType>
  <complexType name="TextRegionType">
    <annotation>
      <documentation>
        Pure text is represented as a text region. This
includes
        drop capitals, but practically ornate text may be
        considered as a graphic.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="pc:RegionType">
        <sequence>
          <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded">
          </element>
          <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
          </element>
          <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0" maxOccurs="1">
          </element>
        </sequence>
        <attribute name="orientation" type="float" use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>

```



```

</attribute>
<attribute name="type" type="pc:TextTypeSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The nature of the text in the region
    </documentation>
  </annotation>
</attribute>
<attribute name="Leading" type="int" use="optional">
  <annotation>
    <documentation>
      The degree of space in points between the
lines of
      text (line spacing)
    </documentation>
  </annotation>
</attribute>
<attribute name="readingDirection"
  type="pc:ReadingDirectionSimpleType" use="optional">
  <annotation>
    <documentation>
      The direction in which text in a region
should be
      read (within lines)
    </documentation>
  </annotation>
</attribute>
<attribute name="textLineOrder"
  type="pc:TextLineOrderSimpleType" use="optional">
  <annotation>
    <documentation>Inner-block order of text lines
(in addition to "readingDirection" which is the inner-text line order of words
and characters)</documentation>
  </annotation>
</attribute>
<attribute name="readingOrientation" type="float"
  use="optional">
  <annotation>
    <documentation>The angle the baseline of text
withing a region has to be rotated (relative to the rectangle encapsulating the
region) in clockwise direction in order to correct the present skew (negative
values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
  </annotation>
</attribute>
<attribute name="indented" type="boolean" use="optional">
  <annotation>
    <documentation>
      Defines whether a region of text is
indented or not
    </documentation>
  </annotation>
</attribute>
<attribute name="align" type="pc:AlignSimpleType">
  <annotation>
    <documentation>Text align</documentation>
  </annotation>
</attribute>

```

```

        <attribute name="primaryLanguage"
type="pc:LanguageSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The primary language used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="secondaryLanguage"
type="pc:LanguageSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The secondary language used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="primaryScript" type="pc:ScriptSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The primary script used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The secondary script used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType"
use="optional"></attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="CoordsType">
    <attribute name="points" use="required" type="pc:PointsType">
    <annotation>
        <documentation>Point list with format "x1,y1 x2,y2
..."</documentation>
    </annotation>
    </attribute>
</complexType>
<complexType name="TextLineType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Baseline" type="pc:BaselineType"
            minOccurs="0">
        <annotation>
            <documentation>
                Multiple connected points that mark the
baseline
                of the glyphs
            </documentation>
        </annotation>
    </sequence>

```

```

        </element>
        <element name="Word" type="pc:WordType" minOccurs="0"
            maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0">
        </element>
        <element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="primaryLanguage"
        type="pc:LanguageSimpleType">
    <annotation>
        <documentation>
            Overrides primaryLanguage attribute of parent
text
            region
        </documentation>
    </annotation>
</attribute>
    <attribute name="primaryScript" type="pc:ScriptSimpleType"
        use="optional">
    <annotation>
        <documentation>
            The primary script used in the text line
        </documentation>
    </annotation>
</attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType"
        use="optional">
    <annotation>
        <documentation>
            The secondary script used in the text line
        </documentation>
    </annotation>
</attribute>
    <attribute name="readingDirection"
        type="pc:ReadingDirectionSimpleType" use="optional">
    <annotation>
        <documentation>
            The direction in which text in a text line
should be read
        </documentation>
    </annotation>
</attribute>
    <attribute name="production" type="pc:ProductionSimpleType"
use="optional">
    <annotation>
        <documentation>
            Overrides the production attribute of the
parent
            text region
        </documentation>
    </annotation>
</attribute>

```

```

    <attribute name="custom" type="string">
      <annotation>
        <documentation>For generic use</documentation>
      </annotation>
    </attribute>
    <attribute name="comments" type="string"></attribute>
  </complexType>
  <complexType name="WordType">
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
      <element name="Glyph" type="pc:GlyphType" minOccurs="0"
        maxOccurs="unbounded">
      </element>
      <element name="TextEquiv" type="pc:TextEquivType"
        minOccurs="0" maxOccurs="unbounded">
      </element>
      <element name="TextStyle" type="pc:TextStyleType"
        minOccurs="0">
      </element>
      <element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="Language" type="pc:LanguageSimpleType">
      <annotation>
        <documentation>
          Overrides primaryLanguage attribute of parent
line
          and/or text region
        </documentation>
      </annotation>
    </attribute>
    <attribute name="primaryScript" type="pc:ScriptSimpleType"
      use="optional">
      <annotation>
        <documentation>
          The primary script used in the word
        </documentation>
      </annotation>
    </attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType"
      use="optional">
      <annotation>
        <documentation>
          The secondary script used in the word
        </documentation>
      </annotation>
    </attribute>
    <attribute name="readingDirection"
      type="pc:ReadingDirectionSimpleType" use="optional">
      <annotation>
        <documentation>
          The direction in which characters in a word
should be read
        </documentation>
      </annotation>
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType"
      use="optional">

```

```

        <annotation>
            <documentation>
                Overrides the production attribute of the
parent
                text line and/or text region.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic use</documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="GlyphType">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Graphemes" type="pc:GraphemesType"
            minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation>
                    Container for graphemes, grapheme groups
and
                    non-printing characters
                </documentation>
            </annotation>
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0">
        </element>
        <element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="Ligature" use="optional" type="boolean">
    </attribute>
    <attribute name="symbol" use="optional" type="boolean">
    </attribute>
    <attribute name="script" type="pc:ScriptSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The script used for the glyph
            </documentation>
        </annotation>
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType"
use="optional">
        <annotation>
            <documentation>
                Overrides the production attribute of the
parent
                word / text line / text region.
            </documentation>
        </annotation>
    </attribute>

```

```

    </attribute>
    <attribute name="custom" type="string">
      <annotation>
        <documentation>For generic use</documentation>
      </annotation>
    </attribute>
    <attribute name="comments" type="string"></attribute>
  </complexType>
  <complexType name="TextEquivType">
    <sequence>
      <element name="PlainText" type="string" minOccurs="0">
        <annotation>
          <documentation>
            Text in a "simple" form (ASCII or
extended ASCII
of
be
            as mostly used for typing). I.e. no use
            special characters for ligatures (should
            be
            stored as two separate characters) etc.
          </documentation>
        </annotation>
      </element>
      <element name="Unicode" type="string">
        <annotation>
          <documentation>
            Correct encoding of the original, always
using
            the corresponding Unicode code point.
I.e.
            ligatures have to be represented as one
            character etc.
          </documentation>
        </annotation>
      </element>
    </sequence>
    <attribute name="index" use="optional">
      <annotation>
        <documentation>Used for sort order in case multiple
TextEquivs are defined. The text content with the lowest index should be
interpreted as the main text content.</documentation>
      </annotation>
      <simpleType>
        <restriction base="integer">
          <minInclusive value="0"></minInclusive>
        </restriction>
      </simpleType>
    </attribute>
    <attribute name="conf">
      <annotation>
        <documentation>OCR confidence value (between 0 and
1)</documentation>
      </annotation>
      <simpleType>
        <restriction base="float">
          <minInclusive value="0"></minInclusive>
          <maxInclusive value="1"></maxInclusive>
        </restriction>
      </simpleType>
    </attribute>
  </complexType>

```

```

        </attribute>
        <attribute name="dataType" type="pc:TextDataTypeSimpleType"
use="optional">
            <annotation>
                <documentation>Type of text content (is it free
text or a number, for instance)
This is only a descriptive attribute, the text type is not checked during XML
validation</documentation>
            </annotation>
        </attribute>
        <attribute name="dataTypeDetails" type="string" use="optional">
            <annotation>
                <documentation>Refinement for dataType
attribute. Can be a regular expression, for instance.</documentation>
            </annotation>
        </attribute>
        <!-- <attribute name="mergeWithNextRule"
type="pc:TextMergeRuleSimpleType" use="optional">
            <annotation>
                <documentation>Rule for merging consecutive
text objects. The rule applies to the first object of a pair (i.e. 'remove-last'
removes the last
character of the first region, can be used to remove hyphen, for
example)</documentation>
            </annotation>
        </attribute>
        <attribute name="mergeWithNextRuleData" type="string"
use="optional">
            <annotation>
                <documentation>Custom data for mergeRule
attribute. Can number of characters to be removed, for example.</documentation>
            </annotation>
        </attribute> -->
        <attribute name="comments" type="string" use="optional"></attribute>
    </complexType>
    <complexType name="ImageRegionType">
        <annotation>
            <documentation>
                An image is considered to be more intricate and
complex
                than a graphic. These can be photos or drawings.
            </documentation>
        </annotation>
        <complexContent>
            <extension base="pc:RegionType">
                <attribute name="orientation" type="float"
use="optional">
                    <annotation>
                        <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                    </annotation>
                </attribute>
                <attribute name="colourDepth"
type="pc:ColourDepthSimpleType" use="optional">
                    <annotation>
                        <documentation>

```


region

The background colour of the

```

        </documentation>
      </annotation>
    </attribute>
    <attribute name="embText" type="boolean"
      use="optional">
      <annotation>
        <documentation>
          Specifies whether the region also
contains
          text
        </documentation>
      </annotation>
    </attribute>
  </extension>
</complexContent>
</complexType>
<complexType name="GraphicRegionType">
  <annotation>
    <documentation>
      Regions containing simple graphics, such as a company
      logo, should be marked as graphic regions.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
      </attribute>
      <attribute name="type" use="optional"
        type="pc:GraphicsTypeSimpleType">
        <annotation>
          <documentation>
            The type of graphic in the region
          </documentation>
        </annotation>
      </attribute>
      <attribute name="numColours" type="int"
        use="optional">
        <annotation>
          <documentation>
            An approximation of the number of
colours
            used in the region
          </documentation>
        </annotation>
      </attribute>
      <attribute name="embText" type="boolean"
        use="optional">
        <annotation>
          <documentation>

```

contains

Specifies whether the region also

```

    text.
  </documentation>
</annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="TableRegionType">
  <annotation>
    <documentation>
      Tabular data in any form is represented with a table
      region. Rows and columns may or may not have separator
      lines; these lines are not separator regions.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
          </annotation>
        </attribute>
        <attribute name="rows" type="int" use="optional">
          <annotation>
            <documentation>
              The number of rows present in the
table
            </documentation>
          </annotation>
        </attribute>
        <attribute name="columns" type="int" use="optional">
          <annotation>
            <documentation>
              The number of columns present in
the table
            </documentation>
          </annotation>
        </attribute>
        <attribute name="LineColour"
type="pc:ColourSimpleType"
          use="optional">
          <annotation>
            <documentation>
              The colour of the lines used in
the region
            </documentation>
          </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType"
          use="optional">
          <annotation>
            <documentation>

```

region The background colour of the

```

        </documentation>
      </annotation>
    </attribute>
    <attribute name="lineSeparators" type="boolean"
      use="optional">
      <annotation>
        <documentation>
          Specifies the presence of line
separators
        </documentation>
      </annotation>
    </attribute>
    <attribute name="embText" type="boolean"
      use="optional">
      <annotation>
        <documentation>
          Specifies whether the region also
contains
          text
        </documentation>
      </annotation>
    </attribute>
  </extension>
</complexContent>
</complexType>
<complexType name="ChartRegionType">
  <annotation>
    <documentation>
      Regions containing charts or graphs of any type,
should
      be marked as chart regions.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
      <annotation>
        <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
      </annotation>
    </attribute>
    <attribute name="type" use="optional"
      type="pc:ChartTypeSimpleType">
    <annotation>
      <documentation>
        The type of chart in the region
      </documentation>
    </annotation>
  </attribute>
  <attribute name="numColours" type="int"
    use="optional">
  <annotation>
    <documentation>

```

```

colours
    An approximation of the number of
    used in the region
    </documentation>
    </annotation>
</attribute>
<attribute name="bgColour" type="pc:ColourSimpleType"
    use="optional">
    <annotation>
    <documentation>
    The background colour of the
region
    </documentation>
    </annotation>
</attribute>
<attribute name="embText" type="boolean"
    use="optional">
    <annotation>
    <documentation>
    Specifies whether the region also
contains
    text
    </documentation>
    </annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="SeparatorRegionType">
    <annotation>
    <documentation>
    Separators are lines that lie between columns and
    paragraphs and can be used to logically separate
    different articles from each other.
    </documentation>
    </annotation>
    <complexContent>
    <extension base="pc:RegionType">
    <attribute name="orientation" type="float"
        use="optional">
        <annotation>
        <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
        </attribute>
        <attribute name="colour" type="pc:ColourSimpleType"
            use="optional">
            <annotation>
            <documentation>
            The colour of the separator
            </documentation>
            </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="MathsRegionType">

```

```

    <annotation>
      <documentation>
        Regions containing equations and mathematical symbols
        should be marked as maths regions.
      </documentation>
    </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
          </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType"
          use="optional">
          <annotation>
            <documentation>
              The background colour of the
region
            </documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
<complexType name="ChemRegionType">
  <annotation>
    <documentation>
      Regions containing chemical formulas.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>
            The angle the rectangle
encapsulating a
region has to be rotated in
clockwise
direction in order to correct the
present
skew (negative values indicate
anti-clockwise rotation). Range:
-179.999,180
          </documentation>
        </annotation>
      </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>

```

```

region
The background colour of the
    </documentation>
    </annotation>
    </attribute>

    </extension>
    </complexContent>
</complexType>
<complexType name="MusicRegionType">
    <annotation>
        <documentation>
            Regions containing musical notations.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="AdvertRegionType">
    <annotation>
        <documentation>
            Regions containing advertisements.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180
                </documentation>
            </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>

```

```

region
    <documentation>
        The background colour of the
    </documentation>
</annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="NoiseRegionType">
    <annotation>
        <documentation>
            Noise regions are regions where no real data lies,
only
            false data created by artifacts on the document or
            scanner noise.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType"></extension>
    </complexContent>
</complexType>
<complexType name="UnknownRegionType">
    <annotation>
        <documentation>
            To be used if the region type cannot be ascertained.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType"></extension>
    </complexContent>
</complexType>

<complexType name="PrintSpaceType">
    <annotation>
        <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.
It does not contain pagenumber (if not part of running title), marginals,
signature mark, preview words.
</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>

<complexType name="ReadingOrderType">
    <annotation>
        <documentation>Definition of the reading order within the page. To
express a reading order between elements they have to be included in an
OrderedGroup. Groups may contain further groups.</documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="1">
        <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
        <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>

```

```

        </choice>
    </complexType>

    <complexType name="RegionRefIndexedType">
        <annotation>
            <documentation>Numbered region</documentation>
        </annotation>
        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
            </annotation></attribute>
        <attribute name="regionRef" type="IDREF" use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupIndexedType">
        <annotation>
            <documentation>
                Indexed group containing ordered elements
            </documentation>
        </annotation>
        <sequence>
            <element name="UserDefined" type="pc:UserDefinedType"
                minOccurs="0" maxOccurs="1">
            </element>

            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="RegionRefIndexed"
                    type="pc:RegionRefIndexedType">
                </element>
                <element name="OrderedGroupIndexed"
                    type="pc:OrderedGroupIndexedType">
                </element>
                <element name="UnorderedGroupIndexed"
                    type="pc:UnorderedGroupIndexedType">
                </element>
            </choice>
        </sequence>
        <attribute name="id" type="ID" use="required"></attribute>
        <attribute name="regionRef" type="IDREF" use="optional">
            <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
        </attribute>
        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>
                    Position (order number) of this item within the
                    current hierarchy level.
                </documentation>
            </annotation>
        </attribute>
        <attribute name="caption" type="string"></attribute>
        <attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
        <attribute name="continuation" type="boolean">
            <annotation>
                <documentation>
                    Is this group a continuation of another group

```

(from


```

        previous column or page, for example)?
    </documentation>
</annotation>
</attribute>
<attribute name="custom" type="string"></attribute>
<attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="UnorderedGroupIndexedType">
    <annotation>
        <documentation>
            Indexed group containing unordered elements
        </documentation>
    </annotation>
    <sequence>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef"
type="pc:RegionRefType"></element>
            <element name="OrderedGroup"
                type="pc:OrderedGroupType">
            </element>
            <element name="UnorderedGroup"
                type="pc:UnorderedGroupType">
            </element>
        </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="regionRef" type="IDREF" use="optional">
        <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
    </attribute>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>
                Position (order number) of this item within the
                current hierarchy level.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="caption" type="string"></attribute>
    <attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
    <attribute name="continuation" type="boolean">
        <annotation>
            <documentation>Is this group a continuation of another group
(from previous column or page, for
example)?</documentation></annotation></attribute>
        <attribute name="custom" type="string"></attribute>
        <attribute name="comments" type="string"></attribute>
    </complexType>

    <complexType name="RegionRefType">
        <attribute name="regionRef" type="IDREF"
use="required"></attribute>
    </complexType>

```

```

<complexType name="OrderedGroupType">
  <annotation>
    <documentation>
      Numbered group (contains ordered elements)
    </documentation>
  </annotation>
  <sequence>
    <element name="UserDefined" type="pc:UserDefinedType"
      minOccurs="0" maxOccurs="1">
    </element>
    <choice minOccurs="1" maxOccurs="unbounded">
      <element name="RegionRefIndexed"
        type="pc:RegionRefIndexedType">
      </element>
      <element name="OrderedGroupIndexed"
        type="pc:OrderedGroupIndexedType">
      </element>
      <element name="UnorderedGroupIndexed"
        type="pc:UnorderedGroupIndexedType">
      </element>
    </choice>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="regionRef" type="IDREF" use="optional">
    <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
  </attribute>
  <attribute name="caption" type="string"></attribute>
  <attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
  <attribute name="continuation" type="boolean">
    <annotation>
      <documentation>Is this group a continuation of another group
(from previous column or page, for example)?</documentation>
    </annotation></attribute>
  <attribute name="custom" type="string"></attribute>
  <attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="UnorderedGroupType">
  <annotation>
    <documentation>
      Numbered group (contains unordered elements)
    </documentation>
  </annotation>
  <sequence>
    <element name="UserDefined" type="pc:UserDefinedType"
      minOccurs="0" maxOccurs="1">
    </element>
    <choice minOccurs="1" maxOccurs="unbounded">
      <element name="RegionRef"
type="pc:RegionRefType"></element>
      <element name="OrderedGroup"
        type="pc:OrderedGroupType">
      </element>
      <element name="UnorderedGroup"
        type="pc:UnorderedGroupType">
      </element>
    </choice>
  </sequence>

```

```

    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="regionRef" type="IDREF" use="optional">
      <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
    </attribute>
    <attribute name="caption" type="string"></attribute>
    <attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
    <attribute name="continuation" type="boolean">
      <annotation>
        <documentation>Is this group a continuation of another group
(from previous column or page, for
example)?</documentation></annotation></attribute>
    <attribute name="custom" type="string"></attribute>
    <attribute name="comments" type="string"></attribute>
  </complexType>

  <complexType name="BorderType">
    <annotation>
      <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>
    </annotation>
    <sequence>
      <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
  </complexType>

  <simpleType name="ColourSimpleType">
    <restriction base="string">
      <enumeration value="black"></enumeration>
      <enumeration value="blue"></enumeration>
      <enumeration value="brown"></enumeration>
      <enumeration value="cyan"></enumeration>
      <enumeration value="green"></enumeration>
      <enumeration value="grey"></enumeration>
      <enumeration value="indigo"></enumeration>
      <enumeration value="magenta"></enumeration>
      <enumeration value="orange"></enumeration>
      <enumeration value="pink"></enumeration>
      <enumeration value="red"></enumeration>
      <enumeration value="turquoise"></enumeration>
      <enumeration value="violet"></enumeration>
      <enumeration value="white"></enumeration>
      <enumeration value="yellow"></enumeration>
      <enumeration value="other"></enumeration>
    </restriction>
  </simpleType>

  <simpleType name="ReadingDirectionSimpleType">
    <restriction base="string">
      <enumeration value="left-to-right"></enumeration>
      <enumeration value="right-to-left"></enumeration>
      <enumeration value="top-to-bottom"></enumeration>
      <enumeration value="bottom-to-top"></enumeration>
    </restriction>
  </simpleType>

  <simpleType name="TextLineOrderSimpleType">
    <restriction base="string">
      <enumeration value="top-to-bottom"></enumeration>
      <enumeration value="bottom-to-top"></enumeration>
    </restriction>
  </simpleType>

```

```

        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
    </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
    <restriction base="string">
        <enumeration value="paragraph"></enumeration>
        <enumeration value="heading"></enumeration>
        <enumeration value="caption"></enumeration>
        <enumeration value="header"></enumeration>
        <enumeration value="footer"></enumeration>
        <enumeration value="page-number"></enumeration>
        <enumeration value="drop-capital"></enumeration>
        <enumeration value="credit"></enumeration>
        <enumeration value="floating"></enumeration>
        <enumeration value="signature-mark"></enumeration>
        <enumeration value="catch-word"></enumeration>
        <enumeration value="marginalia"></enumeration>
        <enumeration value="footnote"></enumeration>
        <enumeration value="footnote-continued"></enumeration>
        <enumeration value="endnote"></enumeration>
        <enumeration value="TOC-entry"></enumeration>
        <enumeration value="List-Label"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="PageTypeSimpleType">
    <restriction base="string">
        <enumeration value="front-cover"></enumeration>
        <enumeration value="back-cover"></enumeration>
        <enumeration value="title"></enumeration>
        <enumeration value="table-of-contents"></enumeration>
        <enumeration value="index"></enumeration>
        <enumeration value="content"></enumeration>
        <enumeration value="blank"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>

<simpleType name="LanguageSimpleType">
    <annotation>
        <documentation>iso15924 2016-07-14</documentation>
    </annotation>
    <restriction base="string">
<enumeration value="Abkhaz"></enumeration>
<enumeration value="Afar"></enumeration>
<enumeration value="Afrikaans"></enumeration>
<enumeration value="Akan"></enumeration>
<enumeration value="Albanian"></enumeration>
<enumeration value="Amharic"></enumeration>
<enumeration value="Arabic"></enumeration>
<enumeration value="Aragonese"></enumeration>
<enumeration value="Armenian"></enumeration>
<enumeration value="Assamese"></enumeration>
<enumeration value="Avaric"></enumeration>
<enumeration value="Avestan"></enumeration>
<enumeration value="Aymara"></enumeration>
<enumeration value="Azerbaijani"></enumeration>
<enumeration value="Bambara"></enumeration>

```

```

<enumeration value="Bashkir"></enumeration>
<enumeration value="Basque"></enumeration>
<enumeration value="Belarusian"></enumeration>
<enumeration value="Bengali"></enumeration>
<enumeration value="Bihari"></enumeration>
<enumeration value="Bislama"></enumeration>
<enumeration value="Bosnian"></enumeration>
<enumeration value="Breton"></enumeration>
<enumeration value="Bulgarian"></enumeration>
<enumeration value="Burmese"></enumeration>
<enumeration value="Cambodian"></enumeration>
<enumeration value="Cantonese"></enumeration>
<enumeration value="Catalan"></enumeration>
<enumeration value="Chamorro"></enumeration>
<enumeration value="Chechen"></enumeration>
<enumeration value="Chichewa"></enumeration>
<enumeration value="Chinese"></enumeration>
<enumeration value="Chuvash"></enumeration>
<enumeration value="Cornish"></enumeration>
<enumeration value="Corsican"></enumeration>
<enumeration value="Cree"></enumeration>
<enumeration value="Croatian"></enumeration>
<enumeration value="Czech"></enumeration>
<enumeration value="Danish"></enumeration>
<enumeration value="Divehi"></enumeration>
<enumeration value="Dutch"></enumeration>
<enumeration value="Dzongkha"></enumeration>
<enumeration value="English"></enumeration>
<enumeration value="Esperanto"></enumeration>
<enumeration value="Estonian"></enumeration>
<enumeration value="Ewe"></enumeration>
<enumeration value="Faroese"></enumeration>
<enumeration value="Fijian"></enumeration>
<enumeration value="Finnish"></enumeration>
<enumeration value="French"></enumeration>
<enumeration value="Fula"></enumeration>
<enumeration value="Gaelic"></enumeration>
<enumeration value="Galician"></enumeration>
<enumeration value="Ganda"></enumeration>
<enumeration value="Georgian"></enumeration>
<enumeration value="German"></enumeration>
<enumeration value="Greek"></enumeration>
<enumeration value="Guaraní"></enumeration>
<enumeration value="Gujarati"></enumeration>
<enumeration value="Haitian"></enumeration>
<enumeration value="Hausa"></enumeration>
<enumeration value="Hebrew"></enumeration>
<enumeration value="Herero"></enumeration>
<enumeration value="Hindi"></enumeration>
<enumeration value="Hiri Motu"></enumeration>
<enumeration value="Hungarian"></enumeration>
<enumeration value="Icelandic"></enumeration>
<enumeration value="Ido"></enumeration>
<enumeration value="Igbo"></enumeration>
<enumeration value="Indonesian"></enumeration>
<enumeration value="Interlingua"></enumeration>
<enumeration value="Interlingue"></enumeration>
<enumeration value="Inuktitut"></enumeration>
<enumeration value="Inupiaq"></enumeration>

```

```

<enumeration value="Irish"></enumeration>
<enumeration value="Italian"></enumeration>
<enumeration value="Japanese"></enumeration>
<enumeration value="Javanese"></enumeration>
<enumeration value="Kalaallisut"></enumeration>
<enumeration value="Kannada"></enumeration>
<enumeration value="Kanuri"></enumeration>
<enumeration value="Kashmiri"></enumeration>
<enumeration value="Kazakh"></enumeration>
<enumeration value="Khmer"></enumeration>
<enumeration value="Kikuyu"></enumeration>
<enumeration value="Kinyarwanda"></enumeration>
<enumeration value="Kirundi"></enumeration>
<enumeration value="Komi"></enumeration>
<enumeration value="Kongo"></enumeration>
<enumeration value="Korean"></enumeration>
<enumeration value="Kurdish"></enumeration>
<enumeration value="Kwanyama"></enumeration>
<enumeration value="Kyrgyz"></enumeration>
<enumeration value="Lao"></enumeration>
<enumeration value="Latin"></enumeration>
<enumeration value="Latvian"></enumeration>
<enumeration value="Limburgish"></enumeration>
<enumeration value="Lingala"></enumeration>
<enumeration value="Lithuanian"></enumeration>
<enumeration value="Luba-Katanga"></enumeration>
<enumeration value="Luxembourgish"></enumeration>
<enumeration value="Macedonian"></enumeration>
<enumeration value="Malagasy"></enumeration>
<enumeration value="Malay"></enumeration>
<enumeration value="Malayalam"></enumeration>
<enumeration value="Maltese"></enumeration>
<enumeration value="Manx"></enumeration>
<enumeration value="Māori"></enumeration>
<enumeration value="Marathi"></enumeration>
<enumeration value="Marshallese"></enumeration>
<enumeration value="Mongolian"></enumeration>
<enumeration value="Nauru"></enumeration>
<enumeration value="Navajo"></enumeration>
<enumeration value="Ndonga"></enumeration>
<enumeration value="Nepali"></enumeration>
<enumeration value="North Ndebele"></enumeration>
<enumeration value="Northern Sami"></enumeration>
<enumeration value="Norwegian"></enumeration>
<enumeration value="Norwegian Bokmål"></enumeration>
<enumeration value="Norwegian Nynorsk"></enumeration>
<enumeration value="Nuosu"></enumeration>
<enumeration value="Occitan"></enumeration>
<enumeration value="Ojibwe"></enumeration>
<enumeration value="Old Church Slavonic"></enumeration>
<enumeration value="Oriya"></enumeration>
<enumeration value="Oromo"></enumeration>
<enumeration value="Ossetian"></enumeration>
<enumeration value="Pāli"></enumeration>
<enumeration value="Panjabi"></enumeration>
<enumeration value="Pashto"></enumeration>
<enumeration value="Persian"></enumeration>
<enumeration value="Polish"></enumeration>
<enumeration value="Portuguese"></enumeration>

```

```

<enumeration value="Punjabi"></enumeration>
<enumeration value="Quechua"></enumeration>
<enumeration value="Romanian"></enumeration>
<enumeration value="Romansh"></enumeration>
<enumeration value="Russian"></enumeration>
<enumeration value="Samoan"></enumeration>
<enumeration value="Sango"></enumeration>
<enumeration value="Sanskrit"></enumeration>
<enumeration value="Sardinian"></enumeration>
<enumeration value="Serbian"></enumeration>
<enumeration value="Shona"></enumeration>
<enumeration value="Sindhi"></enumeration>
<enumeration value="Sinhala"></enumeration>
<enumeration value="Slovak"></enumeration>
<enumeration value="Slovene"></enumeration>
<enumeration value="Somali"></enumeration>
<enumeration value="South Ndebele"></enumeration>
<enumeration value="Southern Sotho"></enumeration>
<enumeration value="Spanish"></enumeration>
<enumeration value="Sundanese"></enumeration>
<enumeration value="Swahili"></enumeration>
<enumeration value="Swati"></enumeration>
<enumeration value="Swedish"></enumeration>
<enumeration value="Tagalog"></enumeration>
<enumeration value="Tahitian"></enumeration>
<enumeration value="Tajik"></enumeration>
<enumeration value="Tamil"></enumeration>
<enumeration value="Tatar"></enumeration>
<enumeration value="Telugu"></enumeration>
<enumeration value="Thai"></enumeration>
<enumeration value="Tibetan"></enumeration>
<enumeration value="Tigrinya"></enumeration>
<enumeration value="Tonga"></enumeration>
<enumeration value="Tsonga"></enumeration>
<enumeration value="Tswana"></enumeration>
<enumeration value="Turkish"></enumeration>
<enumeration value="Turkmen"></enumeration>
<enumeration value="Twi"></enumeration>
<enumeration value="Uighur"></enumeration>
<enumeration value="Ukrainian"></enumeration>
<enumeration value="Urdu"></enumeration>
<enumeration value="Uzbek"></enumeration>
<enumeration value="Venda"></enumeration>
<enumeration value="Vietnamese"></enumeration>
<enumeration value="Volapük"></enumeration>
<enumeration value="Walloon"></enumeration>
<enumeration value="Welsh"></enumeration>
<enumeration value="Western Frisian"></enumeration>
<enumeration value="Wolof"></enumeration>
<enumeration value="Xhosa"></enumeration>
<enumeration value="Yiddish"></enumeration>
<enumeration value="Yoruba"></enumeration>
<enumeration value="Zhuang"></enumeration>
<enumeration value="Zulu"></enumeration>
<enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<simpleType name="ScriptSimpleType">

```



```

    <restriction base="string">
<enumeration value="Adlm - Adlam"></enumeration>
<enumeration value="Afak - Afaka"></enumeration>
<enumeration value="Aghb - Caucasian Albanian"></enumeration>
<enumeration value="Ahom - Ahom, Tai Ahom"></enumeration>
<enumeration value="Arab - Arabic"></enumeration>
<enumeration value="Aran - Arabic (Nastaliq variant)"></enumeration>
<enumeration value="Armi - Imperial Aramaic"></enumeration>
<enumeration value="Armn - Armenian"></enumeration>
<enumeration value="Avst - Avestan"></enumeration>
<enumeration value="Bali - Balinese"></enumeration>
<enumeration value="Bamu - Bamum"></enumeration>
<enumeration value="Bass - Bassa Vah"></enumeration>
<enumeration value="Batk - Batak"></enumeration>
<enumeration value="Beng - Bengali"></enumeration>
<enumeration value="Bhks - Bhaiksuki"></enumeration>
<enumeration value="Blis - Blissymbols"></enumeration>
<enumeration value="Bopo - Bopomofo"></enumeration>
<enumeration value="Brah - Brahmi"></enumeration>
<enumeration value="Brai - Braille"></enumeration>
<enumeration value="Bugi - Buginese"></enumeration>
<enumeration value="Buhd - Buhid"></enumeration>
<enumeration value="Cakm - Chakma"></enumeration>
<enumeration value="Cans - Unified Canadian Aboriginal Syllabics"></enumeration>
<enumeration value="Cari - Carian"></enumeration>
<enumeration value="Cham - Cham"></enumeration>
<enumeration value="Cher - Cherokee"></enumeration>
<enumeration value="Cirt - Cirth"></enumeration>
<enumeration value="Copt - Coptic"></enumeration>
<enumeration value="Cprt - Cypriot"></enumeration>
<enumeration value="Cyrl - Cyrillic"></enumeration>
<enumeration value="Cyrs - Cyrillic (Old Church Slavonic
variant)"></enumeration>
<enumeration value="Deva - Devanagari (Nagari)"></enumeration>
<enumeration value="Dsrt - Deseret (Mormon)"></enumeration>
<enumeration value="Dupl - Duployan shorthand, Duployan
stenography"></enumeration>
<enumeration value="Egyd - Egyptian demotic"></enumeration>
<enumeration value="Egyh - Egyptian hieratic"></enumeration>
<enumeration value="Egyp - Egyptian hieroglyphs"></enumeration>
<enumeration value="Elba - Elbasan"></enumeration>
<enumeration value="Ethi - Ethiopic"></enumeration>
<enumeration value="Geok - Khutsuri (Asomtavruli and Nuskhuri)"></enumeration>
<enumeration value="Geor - Georgian (Mkhedruli)"></enumeration>
<enumeration value="Glag - Glagolitic"></enumeration>
<enumeration value="Goth - Gothic"></enumeration>
<enumeration value="Gran - Grantha"></enumeration>
<enumeration value="Grek - Greek"></enumeration>
<enumeration value="Gujr - Gujarati"></enumeration>
<enumeration value="Guru - Gurmukhi"></enumeration>
<enumeration value="Hanb - Han with Bopomofo"></enumeration>
<enumeration value="Hang - Hangul"></enumeration>
<enumeration value="Hani - Han (Hanzi, Kanji, Hanja)"></enumeration>
<enumeration value="Hano - Hanunoo (Hanunóo)"></enumeration>
<enumeration value="Hans - Han (Simplified variant)"></enumeration>
<enumeration value="Hant - Han (Traditional variant)"></enumeration>
<enumeration value="Hatr - Hatran"></enumeration>
<enumeration value="Hebr - Hebrew"></enumeration>
<enumeration value="Hira - Hiragana"></enumeration>

```



```

<enumeration value="HLuw - Anatolian Hieroglyphs"></enumeration>
<enumeration value="Hmng - Pahawh Hmong"></enumeration>
<enumeration value="Hrkt - Japanese syllabaries"></enumeration>
<enumeration value="Hung - Old Hungarian (Hungarian Runic)"></enumeration>
<enumeration value="Inds - Indus (Harappan)"></enumeration>
<enumeration value="Ital - Old Italic (Etruscan, Oscan etc.)"></enumeration>
<enumeration value="Jamo - Jamo"></enumeration>
<enumeration value="Java - Javanese"></enumeration>
<enumeration value="Jpan - Japanese"></enumeration>
<enumeration value="Jurc - Jurchen"></enumeration>
<enumeration value="Kali - Kayah Li"></enumeration>
<enumeration value="Kana - Katakana"></enumeration>
<enumeration value="Khar - Kharoshthi"></enumeration>
<enumeration value="Khmr - Khmer"></enumeration>
<enumeration value="Khoj - Khojki"></enumeration>
<enumeration value="Kitl - Khitan large script"></enumeration>
<enumeration value="Kits - Khitan small script"></enumeration>
<enumeration value="Knda - Kannada"></enumeration>
<enumeration value="Kore - Korean (alias for Hangul + Han)"></enumeration>
<enumeration value="Kpel - Kpelle"></enumeration>
<enumeration value="Kthi - Kaithi"></enumeration>
<enumeration value="Lana - Tai Tham (Lanna)"></enumeration>
<enumeration value="Laoo - Lao"></enumeration>
<enumeration value="Latf - Latin (Fraktur variant)"></enumeration>
<enumeration value="Latg - Latin (Gaelic variant)"></enumeration>
<enumeration value="Latn - Latin"></enumeration>
<enumeration value="Leke - Leke"></enumeration>
<enumeration value="Lepc - Lepcha (Róng)"></enumeration>
<enumeration value="Limb - Limbu"></enumeration>
<enumeration value="Lina - Linear A"></enumeration>
<enumeration value="Linb - Linear B"></enumeration>
<enumeration value="Lisu - Lisu (Fraser)"></enumeration>
<enumeration value="Loma - Loma"></enumeration>
<enumeration value="Lyci - Lycian"></enumeration>
<enumeration value="Lydi - Lydian"></enumeration>
<enumeration value="Mahj - Mahajani"></enumeration>
<enumeration value="Mand - Mandaic, Mandaean"></enumeration>
<enumeration value="Mani - Manichaean"></enumeration>
<enumeration value="Marc - Marchen"></enumeration>
<enumeration value="Maya - Mayan hieroglyphs"></enumeration>
<enumeration value="Mend - Mende Kikakui"></enumeration>
<enumeration value="Merc - Meroitic Cursive"></enumeration>
<enumeration value="Mero - Meroitic Hieroglyphs"></enumeration>
<enumeration value="MLym - Malayalam"></enumeration>
<enumeration value="Modi - Modi, Modī"></enumeration>
<enumeration value="Mong - Mongolian"></enumeration>
<enumeration value="Moon - Moon (Moon code, Moon script, Moon
type)"></enumeration>
<enumeration value="Mroo - Mro, Mru"></enumeration>
<enumeration value="Mtei - Meitei Mayek (Meithei, Meetei)"></enumeration>
<enumeration value="Mult - Multani"></enumeration>
<enumeration value="Mymr - Myanmar (Burmese)"></enumeration>
<enumeration value="Narb - Old North Arabian (Ancient North
Arabian)"></enumeration>
<enumeration value="Nbat - Nabataean"></enumeration>
<enumeration value="Newa - Newa, Newar, Newari"></enumeration>
<enumeration value="Nkgb - Nakhi Geba"></enumeration>
<enumeration value="Nkoo - N'Ko"></enumeration>
<enumeration value="Nshu - Nüshu"></enumeration>

```

<enumeration value="Ogam - Ogham"></enumeration>
 <enumeration value="Olck - Ol Chiki (Ol Cemet', Ol, Santali)"></enumeration>
 <enumeration value="Orkh - Old Turkic, Orkhon Runic"></enumeration>
 <enumeration value="Orya - Oriya"></enumeration>
 <enumeration value="Osge - Osage"></enumeration>
 <enumeration value="Osma - Osmanya"></enumeration>
 <enumeration value="Palm - Palmyrene"></enumeration>
 <enumeration value="Pauc - Pau Cin Hau"></enumeration>
 <enumeration value="Perm - Old Permic"></enumeration>
 <enumeration value="Phag - Phags-pa"></enumeration>
 <enumeration value="Phli - Inscriptional Pahlavi"></enumeration>
 <enumeration value="Phlp - Psalter Pahlavi"></enumeration>
 <enumeration value="Phlv - Book Pahlavi"></enumeration>
 <enumeration value="Phnx - Phoenician"></enumeration>
 <enumeration value="Piqd - Klingon (KLI pIqaD)"></enumeration>
 <enumeration value="PLrd - Miao (Pollard)"></enumeration>
 <enumeration value="Prti - Inscriptional Parthian"></enumeration>
 <enumeration value="Rjng - Rejang (Redjang, Kaganga)"></enumeration>
 <enumeration value="Roro - Rongorongo"></enumeration>
 <enumeration value="Runr - Runic"></enumeration>
 <enumeration value="Samr - Samaritan"></enumeration>
 <enumeration value="Sara - Sarati"></enumeration>
 <enumeration value="Sarb - Old South Arabian"></enumeration>
 <enumeration value="Saur - Saurashtra"></enumeration>
 <enumeration value="Sgnw - SignWriting"></enumeration>
 <enumeration value="Shaw - Shavian (Shaw)"></enumeration>
 <enumeration value="Shrd - Sharada, Śārādā"></enumeration>
 <enumeration value="Sidd - Siddham"></enumeration>
 <enumeration value="Sind - Khudawadi, Sindhi"></enumeration>
 <enumeration value="Sinh - Sinhala"></enumeration>
 <enumeration value="Sora - Sora Sompeng"></enumeration>
 <enumeration value="Sund - Sundanese"></enumeration>
 <enumeration value="Sylo - Syloti Nagri"></enumeration>
 <enumeration value="SyrC - Syriac"></enumeration>
 <enumeration value="Syre - Syriac (Estrangelo variant)"></enumeration>
 <enumeration value="Syrj - Syriac (Western variant)"></enumeration>
 <enumeration value="Syrn - Syriac (Eastern variant)"></enumeration>
 <enumeration value="Tagb - Tagbanwa"></enumeration>
 <enumeration value="Takr - Takri"></enumeration>
 <enumeration value="Tale - Tai Le"></enumeration>
 <enumeration value="Talu - New Tai Lue"></enumeration>
 <enumeration value="Taml - Tamil"></enumeration>
 <enumeration value="Tang - Tangut"></enumeration>
 <enumeration value="Tavt - Tai Viet"></enumeration>
 <enumeration value="Telu - Telugu"></enumeration>
 <enumeration value="Teng - Tengwar"></enumeration>
 <enumeration value="Tfng - Tifinagh (Berber)"></enumeration>
 <enumeration value="Tglg - Tagalog (Baybayin, Alibata)"></enumeration>
 <enumeration value="Thaa - Thaana"></enumeration>
 <enumeration value="Thai - Thai"></enumeration>
 <enumeration value="Tibt - Tibetan"></enumeration>
 <enumeration value="Tirh - Tirhuta"></enumeration>
 <enumeration value="Ugar - Ugaritic"></enumeration>
 <enumeration value="Vaii - Vai"></enumeration>
 <enumeration value="Visp - Visible Speech"></enumeration>
 <enumeration value="Wara - Warang Citi (Varang Kshiti)"></enumeration>
 <enumeration value="Wole - Woleai"></enumeration>
 <enumeration value="Xpeo - Old Persian"></enumeration>
 <enumeration value="Xsux - Cuneiform, Sumero-Akkadian"></enumeration>

```

<enumeration value="Yiii - Yi"></enumeration>
<enumeration value="Zinh - Code for inherited script"></enumeration>
<enumeration value="Zmth - Mathematical notation"></enumeration>
<enumeration value="Zsye - Symbols (Emoji variant)"></enumeration>
<enumeration value="Zsym - Symbols"></enumeration>
<enumeration value="Zxxx - Code for unwritten documents"></enumeration>
<enumeration value="Zyyy - Code for undetermined script"></enumeration>
<enumeration value="Zzzz - Code for uncoded script"></enumeration>
<enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="ColourDepthSimpleType">
  <restriction base="string">
    <enumeration value="bilevel"></enumeration>
    <enumeration value="greyscale"></enumeration>
    <enumeration value="colour"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="GraphicsTypeSimpleType">
  <restriction base="string">
    <enumeration value="Logo"></enumeration>
    <enumeration value="Letterhead"></enumeration>
    <enumeration value="decoration"></enumeration>
    <enumeration value="frame"></enumeration>
    <enumeration value="handwritten-annotation"></enumeration>
    <enumeration value="stamp"></enumeration>
    <enumeration value="signature"></enumeration>
    <enumeration value="barcode"></enumeration>
    <enumeration value="paper-grow"></enumeration>
    <enumeration value="punch-hole"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="ChartTypeSimpleType">
  <restriction base="string">
    <enumeration value="bar"></enumeration>
    <enumeration value="line"></enumeration>
    <enumeration value="pie"></enumeration>
    <enumeration value="scatter"></enumeration>
    <enumeration value="surface"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<complexType name="LayersType">
  <annotation>
    <documentation>
      Can be used to express the z-index of overlapping
      regions. An element with a greater z-index is always in
      front of another element with lower z-index.
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Layer" type="pc:LayerType"></element>
  </sequence>
</complexType>

```

```

<complexType name="LayerType">
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="RegionRef" type="pc:RegionRefType"></element>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="zIndex" type="int" use="required"></attribute>
  <attribute name="caption" type="string"></attribute>
</complexType>

<complexType name="BaselineType">
  <attribute name="points" type="pc:PointsType" use="required"></attribute>
</complexType>

<simpleType name="PointsType">
  <annotation>
    <documentation>Point list with format "x1,y1 x2,y2
...</documentation>
  </annotation>
  <restriction base="string">
    <pattern value="([0-9]+,[0-9]+ )+([0-9]+,[0-9]+)"></pattern>
  </restriction>
</simpleType>

<complexType name="RelationsType">
  <annotation>
    <documentation>
      Container for one-to-one relations between layout
      objects (for example: DropCap - paragraph, caption -
      image)
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Relation" type="pc:RelationType"></element>
  </sequence>
</complexType>

<complexType name="RelationType">
  <annotation>
    <documentation>
      One-to-one relation between to layout object. Use 'link'
      for loose relations and 'join' for strong relations
      (where something is fragmented for instance).

      Examples for 'link': caption - image floating -
      paragraph paragraph - paragraph (when a paragraph is
      split across columns and the last word of the first
      paragraph DOES NOT continue in the second paragraph)
      drop-cap - paragraph (when the drop-cap is a whole word)

      Examples for 'join': word - word (separated word at the
      end of a line) drop-cap - paragraph (when the drop-cap
      is not a whole word) paragraph - paragraph (when a
      paragraph is split across columns and the last word of
      the first paragraph DOES continue in the second
      paragraph)
    </documentation>
  </annotation>

```

```

<sequence minOccurs="2" maxOccurs="2">
  <element name="RegionRef" type="pc:RegionRefType"></element>
</sequence>
<attribute name="type" use="required">
  <simpleType>
    <restriction base="string">
      <enumeration value="Link"></enumeration>
      <enumeration value="join"></enumeration>
    </restriction>
  </simpleType>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic
use</documentation></annotation></attribute>
  <attribute name="comments" type="string"></attribute>
</complexType>

<simpleType name="ProductionSimpleType">
  <annotation>
    <documentation>Text production type</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="printed"></enumeration>
    <enumeration value="typewritten"></enumeration>
    <enumeration value="handwritten-cursive"></enumeration>
    <enumeration value="handwritten-printschrift"></enumeration>
    <enumeration value="medieval-manuscript"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<complexType name="TextStyleType">
  <annotation>
    <documentation>
      Monospace (fixed-pitch, non-proportional) or
      proportional font
    </documentation>
  </annotation>
  <attribute name="fontFamily" type="string">
    <annotation>
      <documentation>
        For instance: Arial, Times New Roman. Add more
        information if necessary (e.g. blackletter,
        antiqua).
      </documentation>
    </annotation>
  </attribute>
  <attribute name="serif" type="boolean">
    <annotation>
      <documentation>
        Serif or sans-serif typeface
      </documentation>
    </annotation>
  </attribute>
  <attribute name="monospace" type="boolean"></attribute>
  <attribute name="fontSize" type="float">
    <annotation>
      <documentation>

```

```

        The size of the characters in points
    </documentation>
</annotation>
</attribute>
<attribute name="xHeight" type="integer" use="optional">
    <annotation>
        <documentation>The x-height or corpus size refers to the
distance between the baseline and the mean line of lower-case letters in a
typeface. The unit is assumed to be pixels.</documentation>
    </annotation>
</attribute>
<attribute name="kerning" type="int">
    <annotation>
        <documentation>
            The degree of space (in points) between the
            characters in a string of text
        </documentation>
    </annotation>
</attribute>
<attribute name="textColour" type="pc:ColourSimpleType"></attribute>
<attribute name="textColourRgb" type="integer">
    <annotation>
        <documentation>Text colour in RGB encoded format (red value)
+ (256 x green value) + (65536 x blue value)</documentation>
    </annotation>
</attribute>
<attribute name="bgColour" type="pc:ColourSimpleType">
    <annotation>
        <documentation>Background colour</documentation>
    </annotation>
</attribute>
<attribute name="bgColourRgb" type="integer">
    <annotation>
        <documentation>Background colour in RGB encoded format (red
value) + (256 x green value) + (65536 x blue value)</documentation>
    </annotation>
</attribute>
<attribute name="reverseVideo" type="boolean">
    <annotation>
        <documentation>
            Specifies whether the colour of the text appears
            reversed against a background colour
        </documentation>
    </annotation>
</attribute>
<attribute name="bold" type="boolean"></attribute>
<attribute name="italic" type="boolean"></attribute>
<attribute name="underlined" type="boolean"></attribute>
<attribute name="subscript" type="boolean"></attribute>
<attribute name="superscript" type="boolean"></attribute>
<attribute name="strikethrough" type="boolean"></attribute>
    <attribute name="smallCaps" type="boolean"></attribute>
    <attribute name="LetterSpaced" type="boolean"></attribute>
</complexType>

<complexType name="RegionType" abstract="true">
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="UserDefined" type="pc:UserDefinedType"

```

```

        minOccurs="0" maxOccurs="1">
    </element>
    <element name="Roles" type="pc:RolesType" minOccurs="0"
        maxOccurs="1">
        <annotation>
            <documentation>
                Roles the region takes (e.g. in context of a
                parent region)
            </documentation>
        </annotation>
    </element>
    <choice minOccurs="0" maxOccurs="unbounded">
        <element name="TextRegion"
type="pc:TextRegionType"></element>
        <element name="ImageRegion"
type="pc:ImageRegionType"></element>
        <element name="LineDrawingRegion"
            type="pc:LineDrawingRegionType">
        </element>
        <element name="GraphicRegion"
            type="pc:GraphicRegionType">
        </element>
        <element name="TableRegion"
type="pc:TableRegionType"></element>
        <element name="ChartRegion"
type="pc:ChartRegionType"></element>
        <element name="SeparatorRegion"
            type="pc:SeparatorRegionType">
        </element>
        <element name="MathsRegion"
type="pc:MathsRegionType"></element>
        <element name="ChemRegion"
type="pc:ChemRegionType"></element>
        <element name="MusicRegion"
type="pc:MusicRegionType"></element>
        <element name="AdvertRegion"
            type="pc:AdvertRegionType">
        </element>
        <element name="NoiseRegion"
type="pc:NoiseRegionType"></element>
        <element name="UnknownRegion"
            type="pc:UnknownRegionType">
        </element>
    </choice>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
<attribute name="continuation" type="boolean">
    <annotation>
        <documentation>Is this region a continuation of another
        region (in previous column or page, for example)?</documentation>
    </annotation></attribute>
</complexType>

```



```

<complexType name="AlternativeImageType">
  <attribute name="filename" type="string" use="required"></attribute>
  <attribute name="comments" type="string"></attribute>
</complexType>

<simpleType name="AlignSimpleType">
  <restriction base="string">
    <enumeration value="left"></enumeration>
    <enumeration value="centre"></enumeration>
    <enumeration value="right"></enumeration>
    <enumeration value="justify"></enumeration>
  </restriction>
</simpleType>

<simpleType name="GroupTypeSimpleType">
  <restriction base="string">
    <enumeration value="paragraph"></enumeration>
    <enumeration value="list"></enumeration>
    <enumeration value="list-item"></enumeration>
    <enumeration value="figure"></enumeration>
    <enumeration value="article"></enumeration>
    <enumeration value="div"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<simpleType name="TextDataTypeSimpleType">
  <restriction base="string">
    <enumeration value="xsd:decimal">
      <annotation>
        <documentation>Examples: "123.456", "+1234.456", "-
1234.456", "-.456", "-456"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:float">
      <annotation>
        <documentation>Examples: "123.456", "+1234.456", "-
1.2344e56", "-.45E-6", "INF", "-INF", "NaN"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:integer">
      <annotation>
        <documentation>Examples: "123456", "+00000012", "-1",
"-456"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:boolean">
      <annotation>
        <documentation>Examples: "true", "false", "1",
"0"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:date">
      <annotation>
        <documentation>Examples: "2001-10-26", "2001-10-
26+02:00", "2001-10-26Z", "2001-10-26+00:00", "-2001-10-26", "-20000-04-
01"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:time">
      <annotation>
        <documentation>Examples: "21:32:52", "21:32:52+02:00",
"19:32:52Z", "19:32:52+00:00", "21:32:52.12679"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:dateTime">

```



```

        <annotation>
            <documentation>Examples: "2001-10-26T21:32:52", "2001-
10-26T21:32:52+02:00", "2001-10-26T19:32:52Z", "2001-10-26T19:32:52+00:00", "-
2001-10-26T21:32:52", "2001-10-26T21:32:52.12679"</documentation>
        </annotation></enumeration>
    <enumeration value="xsd:string">
        <annotation>
            <documentation>Generic text string</documentation>
        </annotation></enumeration>
    <enumeration value="other">
        <annotation>
            <documentation>An XSD type that is not listed or a
custom type (use dataTypeDetails attribute)</documentation>
        </annotation></enumeration>
</restriction>
</simpleType>

    <!-- <simpleType name="TextMergeRuleSimpleType">
<restriction base="string">
    <enumeration value="no-whitespace">
        <annotation>
            <documentation>Concatenate text without adding
anything</documentation>
        </annotation></enumeration>
    <enumeration value="add-space">
        <annotation>
            <documentation>Adds a single space character between
the text two parts</documentation>
        </annotation></enumeration>
    <enumeration value="add-tab">
        <annotation>
            <documentation>Adds a single tab character between the
text two parts</documentation>
        </annotation></enumeration>
    <enumeration value="add-line-break">
        <annotation>
            <documentation>Adds a single line break character
between the text two parts</documentation>
        </annotation></enumeration>
    <enumeration value="add-custom">
        <annotation>
            <documentation>Adds a user-defined string between the
two text parts (specify in merge rule data)</documentation>
        </annotation></enumeration>
    <enumeration value="remove-last">
        <annotation>
            <documentation>Removes the last character of the first
text part (e.g. remove hyphen)</documentation>
        </annotation></enumeration>
    <enumeration value="remove-multiple">
        <annotation>
            <documentation>Removes the last x characters from the
first text part (specify number in merge rule data)</documentation>
        </annotation></enumeration>
    <enumeration value="remove-all">
        <annotation>
            <documentation>Ignores the first text part completely
(e.g. for drop caps that are repeated in the following
paragraph)</documentation>

```

```

        </annotation></enumeration>
    </restriction>
</simpleType> -->

<complexType name="GraphemesType">
    <annotation>
        <documentation>
            Container for graphemes, grapheme groups and
            non-printing characters
        </documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="Grapheme" type="pc:GraphemeType"></element>
        <element name="NonPrintingChar"
type="pc:NonPrintingCharType"></element>
        <element name="GraphemeGroup"
type="pc:GraphemeGroupType"></element>
    </choice>
</complexType>

<complexType name="GraphemeBaseType">
    <annotation>
        <documentation>Base type for graphemes, grapheme groups and non-
printing characters</documentation>
    </annotation>
    <sequence>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="index" use="required">
        <annotation>
            <documentation>Order index of grapheme, group, or non-printing
character within the parent container (graphemes or glyph or grapheme
group)</documentation>
        </annotation>
        <simpleType>
            <restriction base="int">
                <minInclusive value="0"></minInclusive>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="Ligature" type="boolean"></attribute>
    <attribute name="charType">
        <annotation>
            <documentation>Type of character represented by the
grapheme/group/non-printing character element</documentation>
        </annotation>
        <simpleType>
            <restriction base="string">
                <enumeration value="base"></enumeration>
                <enumeration value="combining"></enumeration>
            </restriction>
        </simpleType>
    </attribute>
    <attribute name="custom" type="string">
        <annotation>

```

```

        <documentation>For generic
use</documentation></annotation></attribute>
        <attribute name="comments" type="string">
        <annotation>
        <documentation>For generic
use</documentation></annotation></attribute>
    </complexType>

    <complexType name="GraphemeType">
        <annotation>
        <documentation>Represents a sub-element of a glyph. Smallest
graphical unit that can be assigned a Unicode code point</documentation>
        </annotation>
        <complexContent>
            <extension base="pc:GraphemeBaseType">
                <sequence>
                    <element name="Coords" type="pc:CoordsType"></element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

    <complexType name="NonPrintingCharType">
        <annotation>
        <documentation>A glyph component without visual representation but
with Unicode code point. Non-visual / non-printing / control character. Part of
grapheme container (of glyph) or grapheme sub group.</documentation>
        </annotation>
        <complexContent>
            <extension base="pc:GraphemeBaseType"></extension>
        </complexContent>
    </complexType>

    <complexType name="GraphemeGroupType">
        <complexContent>
            <extension base="pc:GraphemeBaseType">
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="Grapheme"
type="pc:GraphemeType"></element>
                    <element name="NonPrintingChar"
                        type="pc:NonPrintingCharType">
                    </element>
                </choice>
            </extension>
        </complexContent>
    </complexType>

    <complexType name="UserDefinedType">
        <annotation>
        <documentation>Container for user-defined
attributes</documentation>
        </annotation>
        <sequence>
            <element name="UserAttribute" type="pc:UserAttributeType"
minOccurs="1" maxOccurs="unbounded"></element>
        </sequence>
    </complexType>

    <complexType name="UserAttributeType">

```

```

        <annotation>
            <documentation>Structured custom data defined by name, type and
value.</documentation>
        </annotation>
        <attribute name="name" type="string"></attribute>
        <attribute name="description" type="string"></attribute>
        <attribute name="type">
            <simpleType>
                <restriction base="string">
                    <enumeration value="xsd:string"></enumeration>
                    <enumeration value="xsd:integer"></enumeration>
                    <enumeration value="xsd:boolean"></enumeration>
                    <enumeration value="xsd:float"></enumeration>
                </restriction>
            </simpleType>
        </attribute>
        <attribute name="value" type="string"></attribute>
    </complexType>

    <complexType name="TableCellRoleType">
        <attribute name="rowIndex" type="int" use="required">
            <annotation>
                <documentation>Cell position in table starting with row
0</documentation>
            </annotation></attribute>
        <attribute name="columnIndex" type="int" use="required">
            <annotation>
                <documentation>Cell position in table starting with column
0</documentation>
            </annotation></attribute>
        <attribute name="rowSpan" type="int" use="optional">
            <annotation>
                <documentation>Number of rows the cell spans (optional;
default is 1)</documentation>
            </annotation></attribute>
        <attribute name="colSpan" type="int" use="optional">
            <annotation>
                <documentation>Number of columns the cell spans (optional;
default is 1)</documentation>
            </annotation></attribute>
    </complexType>

    <complexType name="RolesType">
        <sequence>
            <element name="TableCellRole" type="pc:TableCellRoleType"
minOccurs="0" maxOccurs="1">
                <annotation>
                    <documentation>Data for a region that takes on the
role of a table cell within a parent table region</documentation>
                </annotation></element>
        </sequence>
    </complexType>
</schema>

```

Pagecontent Format Specification – Seventh Release

```

<?xml version="1.0" encoding="UTF-8" ?>
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2018-07-
15"
    elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2018-07-
15" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <element name="PcGts" type="pc:PcGtsType">
        <annotation>
            <documentation>Page Content - Ground Truth and
Storage</documentation>
        </annotation></element>

    <complexType name="PcGtsType">
        <sequence>
            <element name="Metadata" type="pc:MetadataType"></element>
            <element name="Page" type="pc:PageType"></element>
        </sequence>
        <attribute name="pcGtsId" type="ID" use="optional"></attribute>
    </complexType>

    <complexType name="MetadataType">
        <sequence>
            <element name="Creator" type="string"></element>
            <element name="Created" type="dateTime">
                <annotation>
                    <documentation>
                        The timestamp has to be in UTC
                        (Coordinated
                        Universal Time) and not local time.
                    </documentation>
                </annotation>
            </element>
            <element name="LastChange" type="dateTime">
                <annotation>
                    <documentation>
                        The timestamp has to be in UTC
                        (Coordinated
                        Universal Time) and not local time.
                    </documentation>
                </annotation>
            </element>
            <element name="Comments" type="string" minOccurs="0"
                maxOccurs="1">
            </element>
            <element name="UserDefined" type="pc:UserDefinedType"
                minOccurs="0" maxOccurs="1">
            </element>

            <element name="MetadataItem" type="pc:MetadataItemType"
minOccurs="0" maxOccurs="unbounded"></element>
        </sequence>
        <attribute name="externalRef" type="string" use="optional">
            <annotation>

```

```

        <documentation>External reference of any
kind</documentation>
        </annotation></attribute>
    </complexType>

    <complexType name="MetadataItemType">
        <sequence>
            <element name="Labels" type="pc:LabelType" minOccurs="0"
maxOccurs="unbounded">
                <annotation>
                    <documentation>Semantic labels /
tags</documentation>
                </annotation></element>
            </sequence>
            <attribute name="type" use="optional">
                <annotation>
                    <documentation>
                        Type of metadata (e.g. author)
                    </documentation>
                </annotation>
                <simpleType>
                    <restriction base="string">
                        <enumeration value="author"></enumeration>
                        <enumeration
value="imageProperties"></enumeration>
                        <enumeration
value="processingStep"></enumeration>
                        <enumeration value="other"></enumeration>
                    </restriction>
                </simpleType>
            </attribute>
            <attribute name="name" type="string" use="optional">
                <annotation>
                    <documentation>
                        E.g. imagePhotometricInterpretation
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="value" type="string" use="required">
                <annotation>
                    <documentation>E.g. RGB</documentation>
                </annotation>
            </attribute>
            <attribute name="date" type="dateTime"></attribute>
        </complexType>
    <complexType name="LabelType">
        <sequence>
            <element name="Label" type="pc:LabelType" minOccurs="0"
maxOccurs="unbounded">
                <annotation>
                    <documentation>
                        A semantic label / tag
                    </documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="externalModel" type="string" use="optional">
            <annotation>
                <documentation>

```

```

        Reference to external model / ontology / schema
    </documentation>
</annotation>
</attribute>

<attribute name="externalId" type="string" use="optional">
    <annotation>
        <documentation>E.g. an RDF resource identifier (to be
used as subject or object of an RDF triple)</documentation>
    </annotation></attribute>
<attribute name="prefix" type="string" use="optional">
    <annotation>
        <documentation>
            Prefix for all labels (e.g. first part of an
URI)
        </documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string" use="optional">
</attribute>

</complexType>
<complexType name="LabelType">
    <annotation>
        <documentation>Semantic label</documentation>
    </annotation>
    <attribute name="value" type="string" use="required">
        <annotation>
            <documentation>
                The label / tag (e.g. 'person'). Can be an RDF
resource identifier (e.g. object of an RDF triple).
            </documentation>
        </annotation>
    </attribute>
    <attribute name="type" type="string" use="optional">
        <annotation>
            <documentation>
                Additional information on the label (e.g.
'YYYY-mm-dd' for a date label). Can be used as predicate of an RDF triple.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string" use="optional">
</attribute>
</complexType>
<complexType name="PageType">
    <sequence>
        <element name="AlternativeImage"
            type="pc:AlternativeImageType" minOccurs="0"
            maxOccurs="unbounded">
            <annotation>
                <documentation>
                    Alternative document page images (e.g.
                    black-and-white)
                </documentation>
            </annotation>
        </element>
        <element name="Border" type="pc:BorderType" minOccurs="0"
            maxOccurs="1">

```

```

</element>
<element name="PrintSpace" type="pc:PrintSpaceType"
  minOccurs="0" maxOccurs="1">
</element>
<element name="ReadingOrder" type="pc:ReadingOrderType"
  minOccurs="0" maxOccurs="1">
  <annotation>
    <documentation></documentation>
  </annotation>
</element>
<element name="Layers" type="pc:LayersType" minOccurs="0"
  maxOccurs="1">
  <annotation>
    <documentation>
      Unassigned regions are considered to be
      (virtual) default layer which is to be
      as below any other layers.
    </documentation>
  </annotation>
</element>
<element name="Relations" type="pc:RelationsType"
  minOccurs="0">
</element>
<element name="UserDefined" type="pc:UserDefinedType" minOccurs="0"
maxOccurs="1"></element>
  <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
    <annotation>
      <documentation>Semantic labels /
tags</documentation>
    </annotation>
  </element>

  <choice minOccurs="0" maxOccurs="unbounded">
    <element name="TextRegion"
type="pc:TextRegionType"></element>
    <element name="ImageRegion" type="pc:ImageRegionType">
</element>
    <element name="LineDrawingRegion"
      type="pc:LineDrawingRegionType">
</element>
    <element name="GraphicRegion"
      type="pc:GraphicRegionType">
</element>
    <element name="TableRegion" type="pc:TableRegionType">
</element>
    <element name="ChartRegion" type="pc:ChartRegionType">
</element>
    <element name="MapRegion"
type="pc:MapRegionType"></element>
    <element name="SeparatorRegion"
      type="pc:SeparatorRegionType">
</element>
    <element name="MathsRegion" type="pc:MathsRegionType">
</element>
    <element name="ChemRegion"
type="pc:ChemRegionType"></element>

```



```

        <element name="MusicRegion"
type="pc:MusicRegionType"></element>
        <element name="AdvertRegion"
            type="pc:AdvertRegionType">
        </element>
        <element name="NoiseRegion" type="pc:NoiseRegionType">
        </element>
        <element name="UnknownRegion"
            type="pc:UnknownRegionType">
        </element>
        <element name="CustomRegion"
            type="pc:CustomRegionType">
        </element>
    </choice>

    </sequence>
    <attribute name="imageFilename" type="string" use="required">
    <annotation>
        <documentation>
            Contains the image file name including the file extension.
        </documentation>
    </annotation>
    </attribute>
    <attribute name="imageWidth" type="int" use="required">
    <annotation>
        <documentation>Specifies the width of the image.</documentation>
    </annotation>
    </attribute>
    <attribute name="imageHeight" type="int" use="required">
    <annotation>
        <documentation>Specifies the height of the image.</documentation>
    </annotation>
    </attribute>
    <attribute name="imageXResolution" type="float" use="optional">
    <annotation>
        <documentation>Specifies the image resolution in
width.</documentation>
    </annotation>
    </attribute>
    <attribute name="imageYResolution" type="float" use="optional">
    <annotation>
        <documentation>Specifies the image resolution in
height.</documentation>
    </annotation>
    </attribute>
    <attribute name="imageResolutionUnit" use="optional">
    <annotation>
        <documentation>
            Specifies the unit of the resolution information
            referring to a standardised unit of measurement
            (pixels per inch, pixels per centimeter or other).
        </documentation>
    </annotation>
    <simpleType>
        <restriction base="string">

        <enumeration value="PPI"></enumeration>
        <enumeration value="PPCM"></enumeration>
        <enumeration value="other"></enumeration>

```

```

        </restriction>
    </simpleType>
</attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="type" type="pc:PageTypeSimpleType">
    <annotation>
        <documentation>Page type</documentation>
    </annotation>
</attribute>
<attribute name="primaryLanguage" type="pc:LanguageSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The primary language used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>
<attribute name="secondaryLanguage" type="pc:LanguageSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The secondary language used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The primary script used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>
<attribute name="secondaryScript" type="pc:ScriptSimpleType"
    use="optional">
    <annotation>
        <documentation>
            The secondary script used in the page (lower-
level definitions override the page-level definition)
        </documentation>
    </annotation>
</attribute>
<attribute name="readingDirection"
    type="pc:ReadingDirectionSimpleType" use="optional">
    <annotation>
        <documentation>
            The direction in which text in a region should
be
            read (within lines) (lower-level definitions
override the page-level definition)
        </documentation>
    </annotation>

```

```

        </attribute>
        <attribute name="textLineOrder"
            type="pc:TextLineOrderSimpleType" use="optional">
            <annotation>
                <documentation>Inner-block order of text lines (in
addition to "readingDirection" which is the inner-text line order of words and
characters) (lower-level definitions override the page-level
definition)</documentation>
            </annotation>
        </attribute>
        <attribute name="conf" type="pc:ConfSimpleType" use="optional">
        <annotation>
            <documentation>Confidence value for whole page (between 0 and
1)</documentation>
        </annotation>
        </attribute>

    </complexType>
    <complexType name="TextRegionType">
        <annotation>
            <documentation>
                Pure text is represented as a text region. This
includes
                drop capitals, but practically ornate text may be
                considered as a graphic.
            </documentation>
        </annotation>
        <complexContent>
            <extension base="pc:RegionType">
                <sequence>
                    <element name="TextLine" type="pc:TextLineType"
                        minOccurs="0" maxOccurs="unbounded">
                    </element>
                    <element name="TextEquiv" type="pc:TextEquivType"
                        minOccurs="0" maxOccurs="unbounded">
                    </element>
                    <element name="TextStyle" type="pc:TextStyleType"
                        minOccurs="0" maxOccurs="1">
                    </element>
                </sequence>
                <attribute name="orientation" type="float" use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
                </attribute>
                <attribute name="type" type="pc:TextTypeSimpleType"
                    use="optional">
                <annotation>
                    <documentation>
                        The nature of the text in the region
                    </documentation>
                </annotation>
                </attribute>
                <attribute name="leading" type="int" use="optional">
                <annotation>
                    <documentation>

```

lines of

The degree of space in points between the text (line spacing)

```

    </documentation>
  </annotation>
</attribute>
<attribute name="readingDirection"
  type="pc:ReadingDirectionSimpleType" use="optional">
  <annotation>
    <documentation>
      The direction in which text in a region
should be
      read (within lines)
    </documentation>
  </annotation>
</attribute>
<attribute name="textLineOrder"
  type="pc:TextLineOrderSimpleType" use="optional">
  <annotation>
    <documentation>Inner-block order of text lines
(in addition to "readingDirection" which is the inner-text line order of words
and characters)</documentation>
  </annotation>
</attribute>
<attribute name="readingOrientation" type="float"
  use="optional">
  <annotation>
    <documentation>The angle the baseline of text
withing a region has to be rotated (relative to the rectangle encapsulating the
region) in clockwise direction in order to correct the present skew (negative
values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
  </annotation>
</attribute>
<attribute name="indented" type="boolean" use="optional">
  <annotation>
    <documentation>
      Defines whether a region of text is
indented or not
    </documentation>
  </annotation>
</attribute>
<attribute name="align" type="pc:AlignSimpleType">
  <annotation>
    <documentation>Text align</documentation>
  </annotation>
</attribute>
<attribute name="primaryLanguage"
type="pc:LanguageSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The primary language used in the region
    </documentation>
  </annotation>
</attribute>
<attribute name="secondaryLanguage"
type="pc:LanguageSimpleType"
  use="optional">

```

```

        <annotation>
            <documentation>
                The secondary language used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="primaryScript" type="pc:ScriptSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The primary script used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType"
        use="optional">
        <annotation>
            <documentation>
                The secondary script used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="production" type="pc:ProductionSimpleType"
use="optional"></attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="CoordsType">
    <attribute name="points" use="required" type="pc:PointsType">
    <annotation>
        <documentation>Point list with format "x1,y1 x2,y2
..."</documentation>
    </annotation>
    </attribute>
    <attribute name="conf" type="pc:ConfSimpleType" use="optional">
    <annotation>
        <documentation>Confidence value (between 0 and 1)</documentation>
    </annotation>
    </attribute>
</complexType>
<complexType name="TextLineType">
    <sequence>
        <element name="AlternativeImage"
            type="pc:AlternativeImageType" minOccurs="0"
            maxOccurs="unbounded">
        <annotation>
            <documentation>
                Alternative text line images (e.g.
                black-and-white)
            </documentation>
        </annotation>
        </element>
        <element name="Coords" type="pc:CoordsType"></element>
        <element name="Baseline" type="pc:BaselineType"
            minOccurs="0">
        <annotation>
            <documentation>
                Multiple connected points that mark the
baseline

```

```

of the glyphs
    </documentation>
  </annotation>
</element>
<element name="Word" type="pc:WordType" minOccurs="0"
  maxOccurs="unbounded">
</element>
<element name="TextEquiv" type="pc:TextEquivType"
  minOccurs="0" maxOccurs="unbounded">
</element>
<element name="TextStyle" type="pc:TextStyleType"
  minOccurs="0">
</element>
<element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
  <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
    <annotation>
      <documentation>Semantic labels /
tags</documentation>
    </annotation>
  </element>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="primaryLanguage"
  type="pc:LanguageSimpleType">
  <annotation>
    <documentation>
      Overrides primaryLanguage attribute of parent
text
    </documentation>
  </annotation>
  </attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The primary script used in the text line
    </documentation>
  </annotation>
  </attribute>
<attribute name="secondaryScript" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The secondary script used in the text line
    </documentation>
  </annotation>
  </attribute>
<attribute name="readingDirection"
  type="pc:ReadingDirectionSimpleType" use="optional">
  <annotation>
    <documentation>
      The direction in which text in a text line
should be read
    </documentation>
  </annotation>
</attribute>

```

```

        <attribute name="production" type="pc:ProductionSimpleType"
use="optional">
            <annotation>
                <documentation>
                    Overrides the production attribute of the
parent
                        text region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="custom" type="string">
            <annotation>
                <documentation>For generic use</documentation>
            </annotation>
        </attribute>
        <attribute name="comments" type="string"></attribute>
        <attribute name="index" type="int" use="optional">
            <annotation>
                <documentation>
                    Position (order number) of this text line
within the
                        parent text region.
                </documentation>
            </annotation>
        </attribute>

    </complexType>
    <complexType name="WordType">
        <sequence>
            <element name="AlternativeImage"
                type="pc:AlternativeImageType" minOccurs="0"
                maxOccurs="unbounded">
                <annotation>
                    <documentation>
                        Alternative word images (e.g.
                        black-and-white)
                    </documentation>
                </annotation>
            </element>
            <element name="Coords" type="pc:CoordsType"></element>
            <element name="Glyph" type="pc:GlyphType" minOccurs="0"
                maxOccurs="unbounded">
            </element>
            <element name="TextEquiv" type="pc:TextEquivType"
                minOccurs="0" maxOccurs="unbounded">
            </element>
            <element name="TextStyle" type="pc:TextStyleType"
                minOccurs="0">
            </element>
            <element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
            <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
                <annotation>
                    <documentation>Semantic labels /
tags</documentation>
                </annotation>
            </element>
        </sequence>

```

```

<attribute name="id" type="ID" use="required"></attribute>
<attribute name="language" type="pc:LanguageSimpleType">
  <annotation>
    <documentation>
      Overrides primaryLanguage attribute of parent
line
      and/or text region
    </documentation>
  </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The primary script used in the word
    </documentation>
  </annotation>
</attribute>
<attribute name="secondaryScript" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The secondary script used in the word
    </documentation>
  </annotation>
</attribute>
<attribute name="readingDirection"
  type="pc:ReadingDirectionSimpleType" use="optional">
  <annotation>
    <documentation>
      The direction in which characters in a word
should be read
    </documentation>
  </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType"
use="optional">
  <annotation>
    <documentation>
      Overrides the production attribute of the
parent
      text line and/or text region.
    </documentation>
  </annotation>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic use</documentation>
  </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
</complexType>
<complexType name="GlyphType">
  <sequence>
    <element name="AlternativeImage"
      type="pc:AlternativeImageType" minOccurs="0"
      maxOccurs="unbounded">
      <annotation>
        <documentation>

```



```

Alternative glyph images (e.g.
black-and-white)
</documentation>
</annotation>
</element>
<element name="Coords" type="pc:CoordsType"></element>
<element name="Graphemes" type="pc:GraphemesType"
  minOccurs="0" maxOccurs="1">
  <annotation>
    <documentation>
      Container for graphemes, grapheme groups
and
      non-printing characters
    </documentation>
  </annotation>
</element>
<element name="TextEquiv" type="pc:TextEquivType"
  minOccurs="0" maxOccurs="unbounded">
</element>
<element name="TextStyle" type="pc:TextStyleType"
  minOccurs="0">
</element>
<element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1"></element>
  <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
    <annotation>
      <documentation>Semantic labels /
tags</documentation>
    </annotation>
  </element>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="Ligature" use="optional" type="boolean">
</attribute>
<attribute name="symbol" use="optional" type="boolean">
</attribute>
<attribute name="script" type="pc:ScriptSimpleType"
  use="optional">
  <annotation>
    <documentation>
      The script used for the glyph
    </documentation>
  </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType"
use="optional">
  <annotation>
    <documentation>
      Overrides the production attribute of the
parent
      word / text line / text region.
    </documentation>
  </annotation>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic use</documentation>
  </annotation>

```

```

        </attribute>
        <attribute name="comments" type="string"></attribute>
    </complexType>
    <complexType name="TextEquivType">
        <sequence>
            <element name="PlainText" type="string" minOccurs="0">
                <annotation>
                    <documentation>
                        Text in a "simple" form (ASCII or
extended ASCII
of
be
                        as mostly used for typing). I.e. no use
                        special characters for ligatures (should
                        stored as two separate characters) etc.
                    </documentation>
                </annotation>
            </element>
            <element name="Unicode" type="string">
                <annotation>
                    <documentation>
                        Correct encoding of the original, always
using
                        the corresponding Unicode code point.
I.e.
                        ligatures have to be represented as one
                        character etc.
                    </documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="index" use="optional">
            <annotation>
                <documentation>Used for sort order in case multiple
TextEquivs are defined. The text content with the lowest index should be
interpreted as the main text content.</documentation>
            </annotation>
        </attribute>
        <simpleType>
            <restriction base="integer">
                <minInclusive value="0"></minInclusive>
            </restriction>
        </simpleType>
    </complexType>
    <attribute name="conf" type="pc:ConfSimpleType" use="optional">
        <annotation>
            <documentation>OCR confidence value (between 0 and
1)</documentation>
        </annotation>
    </attribute>
    <attribute name="dataType" type="pc:TextDataTypeSimpleType"
use="optional">
        <annotation>
            <documentation>Type of text content (is it free
text or a number, for instance)
This is only a descriptive attribute, the text type is not checked during XML
validation</documentation>
        </annotation>
    </attribute>
    <attribute name="dataTypeDetails" type="string" use="optional">

```

```

        <annotation>
            <documentation>Refinement for dataType
attribute. Can be a regular expression, for instance.</documentation>
        </annotation>
    </attribute>
    <!-- <attribute name="mergeWithNextRule"
type="pc:TextMergeRuleSimpleType" use="optional">
        <annotation>
            <documentation>Rule for merging consecutive
text objects. The rule applies to the first object of a pair (i.e. 'remove-last'
removes the last
            character of the first region, can be used to remove hyphen, for
example)</documentation>
        </annotation>
    </attribute>
    <attribute name="mergeWithNextRuleData" type="string"
use="optional">
        <annotation>
            <documentation>Custom data for mergeRule
attribute. Can number of characters to be removed, for example.</documentation>
        </annotation>
    </attribute> -->
    <attribute name="comments" type="string" use="optional"></attribute>
</complexType>
<complexType name="ImageRegionType">
    <annotation>
        <documentation>
            An image is considered to be more intricate and
complex
            than a graphic. These can be photos or drawings.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="colourDepth"
                type="pc:ColourDepthSimpleType" use="optional">
                <annotation>
                    <documentation>
                        The colour bit depth required for
the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>

```

```

        </annotation>
      </attribute>
      <attribute name="embText" type="boolean"
        use="optional">
        <annotation>
          <documentation>
            Specifies whether the region also
contains
            text
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<complexType name="LineDrawingRegionType">
  <annotation>
    <documentation>
      A line drawing is a single colour illustration without
      solid areas.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
      </attribute>
      <attribute name="penColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>
            The pen (foreground) colour of the
region
          </documentation>
        </annotation>
      </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>
            The background colour of the
region
          </documentation>
        </annotation>
      </attribute>
      <attribute name="embText" type="boolean"
        use="optional">
        <annotation>
          <documentation>
            Specifies whether the region also
contains
            text
          </documentation>

```

```

        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<complexType name="GraphicRegionType">
  <annotation>
    <documentation>
      Regions containing simple graphics, such as a company
      logo, should be marked as graphic regions.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
        </attribute>
        <attribute name="type" use="optional"
          type="pc:GraphicsTypeSimpleType">
          <annotation>
            <documentation>
              The type of graphic in the region
            </documentation>
          </annotation>
        </attribute>
        <attribute name="numColours" type="int"
          use="optional">
          <annotation>
            <documentation>
              An approximation of the number of
colours
              used in the region
            </documentation>
          </annotation>
        </attribute>
        <attribute name="embText" type="boolean"
          use="optional">
          <annotation>
            <documentation>
              Specifies whether the region also
contains
              text.
            </documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
<complexType name="TableRegionType">
  <annotation>
    <documentation>
      Tabular data in any form is represented with a table
      region. Rows and columns may or may not have separator

```

```

        lines; these lines are not separator regions.
    </documentation>
</annotation>
<complexContent>
    <extension base="pc:RegionType">
        <sequence>
            <element name="Grid" type="pc:GridType"
minOccurs="0" maxOccurs="1">
                <annotation>
                    <documentation>Table grid (visible
or virtual grid lines)</documentation>
                </annotation></element>
            </sequence>
            <attribute name="orientation" type="float"
use="optional">
                <annotation>
                    <documentation>
                        The angle the rectangle
encapsulating a
                        region has to be rotated in
clockwise
                        direction in order to correct the
present
                        skew (negative values indicate
                        anti-clockwise rotation). Range:
                        -179.999,180
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="rows" type="int" use="optional">
                <annotation>
                    <documentation>
                        The number of rows present in the
table
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="columns" type="int" use="optional">
                <annotation>
                    <documentation>
                        The number of columns present in
the table
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="LineColour"
type="pc:ColourSimpleType"
use="optional">
                <annotation>
                    <documentation>
                        The colour of the lines used in
the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
use="optional">
                <annotation>
                    <documentation>

```

```

region
    The background colour of the
    </documentation>
    </annotation>
  </attribute>
  <attribute name="lineSeparators" type="boolean"
    use="optional">
    <annotation>
      <documentation>
        Specifies the presence of line
separators
      </documentation>
    </annotation>
  </attribute>
  <attribute name="embText" type="boolean"
    use="optional">
    <annotation>
      <documentation>
        Specifies whether the region also
contains
      </documentation>
    </annotation>
  </attribute>
</extension>
</complexContent>
</complexType>
<complexType name="GridType">
  <annotation>
    <documentation>Matrix of grid points defining the table grid on the
page</documentation>
  </annotation>
  <sequence>
    <element name="GridPoints" type="pc:GridPointsType" minOccurs="2"
maxOccurs="unbounded">
      <annotation>
        <documentation>One row in the grid point matrix. Points with
x,y coordinates. (note: for a table with n table rows there should be n+1 grid
rows)</documentation>
      </annotation></element>
    </sequence>
  </complexType>
  <complexType name="GridPointsType">
    <annotation>
      <documentation>Points with x,y coordinates.</documentation>
    </annotation>
    <attribute name="index" type="int" use="required">
      <annotation>
        <documentation>
          The grid row index
        </documentation>
      </annotation>
    </attribute>
    <attribute name="points" type="pc:PointsType"
use="required"></attribute>
  </complexType>

  <complexType name="ChartRegionType">
    <annotation>

```

should

```

    <documentation>
        Regions containing charts or graphs of any type,
    </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </documentation>
            </attribute>
            <attribute name="type" use="optional"
                type="pc:ChartTypeSimpleType">
                <documentation>
                    The type of chart in the region
                </documentation>
            </attribute>
            <attribute name="numColours" type="int"
                use="optional">
                <documentation>
                    An approximation of the number of
colours
                    used in the region
                </documentation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <documentation>
                    The background colour of the
region
                </documentation>
            </attribute>
            <attribute name="embText" type="boolean"
                use="optional">
                <documentation>
                    Specifies whether the region also
contains
                    text
                </documentation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="SeparatorRegionType">
    <annotation>

```



```

        <documentation>
            Separators are lines that lie between columns and
            paragraphs and can be used to logically separate
            different articles from each other.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="colour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The colour of the separator
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="MathsRegionType">
    <annotation>
        <documentation>
            Regions containing equations and mathematical symbols
            should be marked as maths regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float"
                use="optional">
                <annotation>
                    <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType"
                use="optional">
                <annotation>
                    <documentation>
                        The background colour of the
region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="ChemRegionType">

```

```

    <annotation>
      <documentation>
        Regions containing chemical formulas.
      </documentation>
    </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>
            The angle the rectangle
encapsulating a
clockwise
present
            region has to be rotated in
            direction in order to correct the
            skew (negative values indicate
            anti-clockwise rotation). Range:
            -179.999,180
          </documentation>
        </annotation>
      </attribute>

      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>
            The background colour of the
region
          </documentation>
        </annotation>
      </attribute>

    </extension>
  </complexContent>
</complexType>
<complexType name="MapRegionType">
  <annotation>
    <documentation>
      Regions containing maps.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>
            The angle the rectangle
encapsulating a
clockwise
present
            region has to be rotated in
            direction in order to correct the
            skew (negative values indicate
            anti-clockwise rotation). Range:
            -179.999,180
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>

```

```

        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<complexType name="MusicRegionType">
  <annotation>
    <documentation>
      Regions containing musical notations.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180</documentation>
        </annotation>
      </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>
            The background colour of the
region
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<complexType name="AdvertRegionType">
  <annotation>
    <documentation>
      Regions containing advertisements.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float"
        use="optional">
        <annotation>
          <documentation>The angle the rectangle
encapsulating a region has to be rotated in clockwise direction in order to
correct the present skew (negative values indicate anti-clockwise rotation).
Range: -179.999,180
        </documentation>
        </annotation>
      </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType"
        use="optional">
        <annotation>
          <documentation>
            The background colour of the
region
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>

```

```

        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<complexType name="NoiseRegionType">
  <annotation>
    <documentation>
      Noise regions are regions where no real data lies,
only      false data created by artifacts on the document or
      scanner noise.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType"></extension>
  </complexContent>
</complexType>
<complexType name="UnknownRegionType">
  <annotation>
    <documentation>
      To be used if the region type cannot be ascertained.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType"></extension>
  </complexContent>
</complexType>
<complexType name="CustomRegionType">
  <annotation>
    <documentation>
      Regions containing content that is not covered by the
default types (text, graphic, image, line drawing, chart, table, separator,
maths, map, music, chem, advert, noise, unknown)
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="type" use="optional"
        type="string">
        <annotation>
          <documentation>
            Information on the type of content
represented by this region
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>

<complexType name="PrintSpaceType">
  <annotation>
    <documentation>Determines the effective area on the paper of a
printed page. Its size is equal for all pages of a book (exceptions: titlepage,
multipage pictures).
It contains all living elements (except marginals) like body type, footnotes,
headings, running titles.

```

It does not contain pagenumber (if not part of running title), marginals, signature mark, preview words.

```

</documentation>
  </annotation>
  <sequence>
    <element name="Coords" type="pc:CoordsType"></element>
  </sequence>
</complexType>

<complexType name="ReadingOrderType">
  <annotation>
    <documentation>Definition of the reading order within the page. To
express a reading order between elements they have to be included in an
OrderedGroup. Groups may contain further groups.</documentation>
  </annotation>

  <choice minOccurs="1" maxOccurs="1">
    <element name="OrderedGroup" type="pc:OrderedGroupType"></element>
    <element name="UnorderedGroup"
type="pc:UnorderedGroupType"></element>
  </choice>

  <attribute name="conf" type="pc:ConfSimpleType" use="optional">
<annotation>
  <documentation>Confidence value (between 0 and 1)</documentation>
</annotation>
</attribute>

</complexType>

<complexType name="RegionRefIndexedType">
  <annotation>
    <documentation>Numbered region</documentation>
  </annotation>
  <attribute name="index" type="int" use="required">
    <annotation>
      <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
    </annotation>
  </attribute>
  <attribute name="regionRef" type="IDREF" use="required"></attribute>
</complexType>

<complexType name="OrderedGroupIndexedType">
  <annotation>
    <documentation>
      Indexed group containing ordered elements
    </documentation>
  </annotation>
  <sequence>
    <element name="UserDefined" type="pc:UserDefinedType"
minOccurs="0" maxOccurs="1">
    </element>
    <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
      <annotation>
        <documentation>Semantic labels /
tags</documentation>
      </annotation>
    </element>
  </sequence>

```

```

    <choice minOccurs="1" maxOccurs="unbounded">
      <element name="RegionRefIndexed"
        type="pc:RegionRefIndexedType">
      </element>
      <element name="OrderedGroupIndexed"
        type="pc:OrderedGroupIndexedType">
      </element>
      <element name="UnorderedGroupIndexed"
        type="pc:UnorderedGroupIndexedType">
      </element>
    </choice>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="regionRef" type="IDREF" use="optional">
    <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
  </attribute>
  <attribute name="index" type="int" use="required">
    <annotation>
      <documentation>
        Position (order number) of this item within the
        current hierarchy level.
      </documentation>
    </annotation>
  </attribute>
  <attribute name="caption" type="string"></attribute>
  <attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
  <attribute name="continuation" type="boolean">
    <annotation>
      <documentation>
        Is this group a continuation of another group
        (from
          previous column or page, for example)?
      </documentation>
    </annotation>
  </attribute>
  <attribute name="custom" type="string"></attribute>
  <attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="UnorderedGroupIndexedType">
  <annotation>
    <documentation>
      Indexed group containing unordered elements
    </documentation>
  </annotation>
  <sequence>
    <element name="UserDefined" type="pc:UserDefinedType"
      minOccurs="0" maxOccurs="1">
    </element>
    <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
      <annotation>
        <documentation>Semantic labels /
tags</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

```

```

        <element name="RegionRef"
type="pc:RegionRefType"></element>
        <element name="OrderedGroup"
            type="pc:OrderedGroupType">
        </element>
        <element name="UnorderedGroup"
            type="pc:UnorderedGroupType">
        </element>
    </choice>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="regionRef" type="IDREF" use="optional">
    <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
</attribute>
<attribute name="index" type="int" use="required">
    <annotation>
        <documentation>
            Position (order number) of this item within the
            current hierarchy level.
        </documentation>
    </annotation>
</attribute>
<attribute name="caption" type="string"></attribute>
<attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
<attribute name="continuation" type="boolean">
    <annotation>
        <documentation>Is this group a continuation of another group
(from previous column or page, for
example)?</documentation></annotation></attribute>
        <attribute name="custom" type="string"></attribute>
        <attribute name="comments" type="string"></attribute>
    </complexType>

    <complexType name="RegionRefType">
        <attribute name="regionRef" type="IDREF"
use="required"></attribute>
    </complexType>

    <complexType name="OrderedGroupType">
        <annotation>
            <documentation>
                Numbered group (contains ordered elements)
            </documentation>
        </annotation>
        <sequence>
            <element name="UserDefined" type="pc:UserDefinedType"
                minOccurs="0" maxOccurs="1">
            </element>
            <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
                <annotation>
                    <documentation>Semantic labels /
tags</documentation>
                </annotation>
            </element>
            <choice minOccurs="1" maxOccurs="unbounded">
                <element name="RegionRefIndexed"

```

```

        type="pc:RegionRefIndexedType">
    </element>
    <element name="OrderedGroupIndexed"
        type="pc:OrderedGroupIndexedType">
    </element>
    <element name="UnorderedGroupIndexed"
        type="pc:UnorderedGroupIndexedType">
    </element>
</choice>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="regionRef" type="IDREF" use="optional">
    <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
</attribute>
<attribute name="caption" type="string"></attribute>
<attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
<attribute name="continuation" type="boolean">
    <annotation>
        <documentation>Is this group a continuation of another group
(from previous column or page, for example)?</documentation>
    </annotation></attribute>
<attribute name="custom" type="string"></attribute>
<attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="UnorderedGroupType">
    <annotation>
        <documentation>
            Numbered group (contains unordered elements)
        </documentation>
    </annotation>
    <sequence>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef"
type="pc:RegionRefType"></element>
            <element name="OrderedGroup"
                type="pc:OrderedGroupType">
            </element>
            <element name="UnorderedGroup"
                type="pc:UnorderedGroupType">
            </element>
        </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="regionRef" type="IDREF" use="optional">

```



```

        <annotation><documentation>Optional link to a parent region
of nested regions. The parent region doubles as reading order group. Only the
nested regions should be allowed as group members.</documentation></annotation>
    </attribute>
    <attribute name="caption" type="string"></attribute>
    <attribute name="type" type="pc:GroupTypeSimpleType"></attribute>
    <attribute name="continuation" type="boolean">
    <annotation>
        <documentation>Is this group a continuation of another group
(from previous column or page, for
example)?</documentation></annotation></attribute>
    <attribute name="custom" type="string"></attribute>
    <attribute name="comments" type="string"></attribute>
</complexType>

<complexType name="BorderType">
    <annotation>
        <documentation>Border of the actual page (if the scanned image
contains parts not belonging to the page).</documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"></element>
    </sequence>
</complexType>
<simpleType name="ColourSimpleType">
    <restriction base="string">
        <enumeration value="black"></enumeration>
        <enumeration value="blue"></enumeration>
        <enumeration value="brown"></enumeration>
        <enumeration value="cyan"></enumeration>
        <enumeration value="green"></enumeration>
        <enumeration value="grey"></enumeration>
        <enumeration value="indigo"></enumeration>
        <enumeration value="magenta"></enumeration>
        <enumeration value="orange"></enumeration>
        <enumeration value="pink"></enumeration>
        <enumeration value="red"></enumeration>
        <enumeration value="turquoise"></enumeration>
        <enumeration value="violet"></enumeration>
        <enumeration value="white"></enumeration>
        <enumeration value="yellow"></enumeration>
        <enumeration value="other"></enumeration>
    </restriction>
</simpleType>
<simpleType name="ReadingDirectionSimpleType">
    <restriction base="string">
        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
        <enumeration value="top-to-bottom"></enumeration>
        <enumeration value="bottom-to-top"></enumeration>
    </restriction>
</simpleType>
<simpleType name="TextLineOrderSimpleType">
    <restriction base="string">
        <enumeration value="top-to-bottom"></enumeration>
        <enumeration value="bottom-to-top"></enumeration>
        <enumeration value="left-to-right"></enumeration>
        <enumeration value="right-to-left"></enumeration>
    </restriction>

```

```

</simpleType>
<simpleType name="TextTypeSimpleType">
  <restriction base="string">
    <enumeration value="paragraph"></enumeration>
    <enumeration value="heading"></enumeration>
    <enumeration value="caption"></enumeration>
    <enumeration value="header"></enumeration>
    <enumeration value="footer"></enumeration>
    <enumeration value="page-number"></enumeration>
    <enumeration value="drop-capital"></enumeration>
    <enumeration value="credit"></enumeration>
    <enumeration value="floating"></enumeration>
    <enumeration value="signature-mark"></enumeration>
    <enumeration value="catch-word"></enumeration>
    <enumeration value="marginalia"></enumeration>
    <enumeration value="footnote"></enumeration>
    <enumeration value="footnote-continued"></enumeration>
    <enumeration value="endnote"></enumeration>
    <enumeration value="TOC-entry"></enumeration>
    <enumeration value="List-Label"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="PageTypeSimpleType">
  <restriction base="string">
    <enumeration value="front-cover"></enumeration>
    <enumeration value="back-cover"></enumeration>
    <enumeration value="title"></enumeration>
    <enumeration value="table-of-contents"></enumeration>
    <enumeration value="index"></enumeration>
    <enumeration value="content"></enumeration>
    <enumeration value="blank"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="ConfSimpleType">
  <restriction base="float">
    <minInclusive value="0"></minInclusive>
    <maxInclusive value="1"></maxInclusive>
  </restriction>
</simpleType>
<simpleType name="LanguageSimpleType">
  <annotation>
    <documentation> ISO 639.x 2016-07-14</documentation>
  </annotation>
  <restriction base="string">
<enumeration value="Abkhaz"></enumeration>
<enumeration value="Afar"></enumeration>
<enumeration value="Afrikaans"></enumeration>
<enumeration value="Akan"></enumeration>
<enumeration value="Albanian"></enumeration>
<enumeration value="Amharic"></enumeration>
<enumeration value="Arabic"></enumeration>
<enumeration value="Aragonese"></enumeration>
<enumeration value="Armenian"></enumeration>
<enumeration value="Assamese"></enumeration>
<enumeration value="Avaric"></enumeration>
<enumeration value="Avestan"></enumeration>
<enumeration value="Aymara"></enumeration>

```

```

<enumeration value="Azerbaijani"></enumeration>
<enumeration value="Bambara"></enumeration>
<enumeration value="Bashkir"></enumeration>
<enumeration value="Basque"></enumeration>
<enumeration value="Belarusian"></enumeration>
<enumeration value="Bengali"></enumeration>
<enumeration value="Bihari"></enumeration>
<enumeration value="Bislama"></enumeration>
<enumeration value="Bosnian"></enumeration>
<enumeration value="Breton"></enumeration>
<enumeration value="Bulgarian"></enumeration>
<enumeration value="Burmese"></enumeration>
<enumeration value="Cambodian"></enumeration>
<enumeration value="Cantonese"></enumeration>
<enumeration value="Catalan"></enumeration>
<enumeration value="Chamorro"></enumeration>
<enumeration value="Chechen"></enumeration>
<enumeration value="Chichewa"></enumeration>
<enumeration value="Chinese"></enumeration>
<enumeration value="Chuvash"></enumeration>
<enumeration value="Cornish"></enumeration>
<enumeration value="Corsican"></enumeration>
<enumeration value="Cree"></enumeration>
<enumeration value="Croatian"></enumeration>
<enumeration value="Czech"></enumeration>
<enumeration value="Danish"></enumeration>
<enumeration value="Divehi"></enumeration>
<enumeration value="Dutch"></enumeration>
<enumeration value="Dzongkha"></enumeration>
<enumeration value="English"></enumeration>
<enumeration value="Esperanto"></enumeration>
<enumeration value="Estonian"></enumeration>
<enumeration value="Ewe"></enumeration>
<enumeration value="Faroese"></enumeration>
<enumeration value="Fijian"></enumeration>
<enumeration value="Finnish"></enumeration>
<enumeration value="French"></enumeration>
<enumeration value="FuLa"></enumeration>
<enumeration value="Gaelic"></enumeration>
<enumeration value="Galician"></enumeration>
<enumeration value="Ganda"></enumeration>
<enumeration value="Georgian"></enumeration>
<enumeration value="German"></enumeration>
<enumeration value="Greek"></enumeration>
<enumeration value="Guaraní"></enumeration>
<enumeration value="Gujarati"></enumeration>
<enumeration value="Haitian"></enumeration>
<enumeration value="Hausa"></enumeration>
<enumeration value="Hebrew"></enumeration>
<enumeration value="Herero"></enumeration>
<enumeration value="Hindi"></enumeration>
<enumeration value="Hiri Motu"></enumeration>
<enumeration value="Hungarian"></enumeration>
<enumeration value="Icelandic"></enumeration>
<enumeration value="Ido"></enumeration>
<enumeration value="Igbo"></enumeration>
<enumeration value="Indonesian"></enumeration>
<enumeration value="Interlingua"></enumeration>
<enumeration value="Interlingue"></enumeration>

```

```

<enumeration value="Inuktitut"></enumeration>
<enumeration value="Inupiaq"></enumeration>
<enumeration value="Irish"></enumeration>
<enumeration value="Italian"></enumeration>
<enumeration value="Japanese"></enumeration>
<enumeration value="Javanese"></enumeration>
<enumeration value="Kalaallisut"></enumeration>
<enumeration value="Kannada"></enumeration>
<enumeration value="Kanuri"></enumeration>
<enumeration value="Kashmiri"></enumeration>
<enumeration value="Kazakh"></enumeration>
<enumeration value="Khmer"></enumeration>
<enumeration value="Kikuyu"></enumeration>
<enumeration value="Kinyarwanda"></enumeration>
<enumeration value="Kirundi"></enumeration>
<enumeration value="Komi"></enumeration>
<enumeration value="Kongo"></enumeration>
<enumeration value="Korean"></enumeration>
<enumeration value="Kurdish"></enumeration>
<enumeration value="Kwanyama"></enumeration>
<enumeration value="Kyrgyz"></enumeration>
<enumeration value="Lao"></enumeration>
<enumeration value="Latin"></enumeration>
<enumeration value="Latvian"></enumeration>
<enumeration value="Limburgish"></enumeration>
<enumeration value="Lingala"></enumeration>
<enumeration value="Lithuanian"></enumeration>
<enumeration value="Luba-Katanga"></enumeration>
<enumeration value="Luxembourgish"></enumeration>
<enumeration value="Macedonian"></enumeration>
<enumeration value="Malagasy"></enumeration>
<enumeration value="Malay"></enumeration>
<enumeration value="Malayalam"></enumeration>
<enumeration value="Maltese"></enumeration>
<enumeration value="Manx"></enumeration>
<enumeration value="Māori"></enumeration>
<enumeration value="Marathi"></enumeration>
<enumeration value="Marshallese"></enumeration>
<enumeration value="Mongolian"></enumeration>
<enumeration value="Nauru"></enumeration>
<enumeration value="Navajo"></enumeration>
<enumeration value="Ndonga"></enumeration>
<enumeration value="Nepali"></enumeration>
<enumeration value="North Ndebele"></enumeration>
<enumeration value="Northern Sami"></enumeration>
<enumeration value="Norwegian"></enumeration>
<enumeration value="Norwegian Bokmål"></enumeration>
<enumeration value="Norwegian Nynorsk"></enumeration>
<enumeration value="Nuosu"></enumeration>
<enumeration value="Occitan"></enumeration>
<enumeration value="Ojibwe"></enumeration>
<enumeration value="Old Church Slavonic"></enumeration>
<enumeration value="Oriya"></enumeration>
<enumeration value="Oromo"></enumeration>
<enumeration value="Ossetian"></enumeration>
<enumeration value="Pāli"></enumeration>
<enumeration value="Panjabi"></enumeration>
<enumeration value="Pashto"></enumeration>
<enumeration value="Persian"></enumeration>

```

```

<enumeration value="Polish"></enumeration>
<enumeration value="Portuguese"></enumeration>
<enumeration value="Punjabi"></enumeration>
<enumeration value="Quechua"></enumeration>
<enumeration value="Romanian"></enumeration>
<enumeration value="Romansh"></enumeration>
<enumeration value="Russian"></enumeration>
<enumeration value="Samoan"></enumeration>
<enumeration value="Sango"></enumeration>
<enumeration value="Sanskrit"></enumeration>
<enumeration value="Sardinian"></enumeration>
<enumeration value="Serbian"></enumeration>
<enumeration value="Shona"></enumeration>
<enumeration value="Sindhi"></enumeration>
<enumeration value="Sinhala"></enumeration>
<enumeration value="Slovak"></enumeration>
<enumeration value="Slovene"></enumeration>
<enumeration value="Somali"></enumeration>
<enumeration value="South Ndebele"></enumeration>
<enumeration value="Southern Sotho"></enumeration>
<enumeration value="Spanish"></enumeration>
<enumeration value="Sundanese"></enumeration>
<enumeration value="Swahili"></enumeration>
<enumeration value="Swati"></enumeration>
<enumeration value="Swedish"></enumeration>
<enumeration value="Tagalog"></enumeration>
<enumeration value="Tahitian"></enumeration>
<enumeration value="Tajik"></enumeration>
<enumeration value="Tamil"></enumeration>
<enumeration value="Tatar"></enumeration>
<enumeration value="Telugu"></enumeration>
<enumeration value="Thai"></enumeration>
<enumeration value="Tibetan"></enumeration>
<enumeration value="Tigrinya"></enumeration>
<enumeration value="Tonga"></enumeration>
<enumeration value="Tsonga"></enumeration>
<enumeration value="Tswana"></enumeration>
<enumeration value="Turkish"></enumeration>
<enumeration value="Turkmen"></enumeration>
<enumeration value="Twi"></enumeration>
<enumeration value="Uighur"></enumeration>
<enumeration value="Ukrainian"></enumeration>
<enumeration value="Urdu"></enumeration>
<enumeration value="Uzbek"></enumeration>
<enumeration value="Venda"></enumeration>
<enumeration value="Vietnamese"></enumeration>
<enumeration value="Volapük"></enumeration>
<enumeration value="Walloon"></enumeration>
<enumeration value="Welsh"></enumeration>
<enumeration value="Western Frisian"></enumeration>
<enumeration value="Wolof"></enumeration>
<enumeration value="Xhosa"></enumeration>
<enumeration value="Yiddish"></enumeration>
<enumeration value="Yoruba"></enumeration>
<enumeration value="Zhuang"></enumeration>
<enumeration value="Zulu"></enumeration>
<enumeration value="other"></enumeration>
  </restriction>
</simpleType>

```

```

<simpleType name="ScriptSimpleType">
  <annotation>
    <documentation> iso15924 2016-07-14</documentation>
  </annotation>

  <restriction base="string">
<enumeration value="AdLm - AdLam"></enumeration>
<enumeration value="Afak - Afaka"></enumeration>
<enumeration value="Aghb - Caucasian Albanian"></enumeration>
<enumeration value="Ahom - Ahom, Tai Ahom"></enumeration>
<enumeration value="Arab - Arabic"></enumeration>
<enumeration value="Aran - Arabic (Nastaliq variant)"></enumeration>
<enumeration value="Armi - Imperial Aramaic"></enumeration>
<enumeration value="Armn - Armenian"></enumeration>
<enumeration value="Avst - Avestan"></enumeration>
<enumeration value="Bali - Balinese"></enumeration>
<enumeration value="Bamu - Bamum"></enumeration>
<enumeration value="Bass - Bassa Vah"></enumeration>
<enumeration value="Batk - Batak"></enumeration>
<enumeration value="Beng - Bengali"></enumeration>
<enumeration value="Bhks - Bhaiksuki"></enumeration>
<enumeration value="Blis - Blissymbols"></enumeration>
<enumeration value="Bopo - Bopomofo"></enumeration>
<enumeration value="Brah - Brahmi"></enumeration>
<enumeration value="Brai - Braille"></enumeration>
<enumeration value="Bugi - Buginese"></enumeration>
<enumeration value="Buhd - Buhid"></enumeration>
<enumeration value="Cakm - Chakma"></enumeration>
<enumeration value="Cans - Unified Canadian Aboriginal Syllabics"></enumeration>
<enumeration value="Cari - Carian"></enumeration>
<enumeration value="Cham - Cham"></enumeration>
<enumeration value="Cher - Cherokee"></enumeration>
<enumeration value="Cirt - Cirth"></enumeration>
<enumeration value="Copt - Coptic"></enumeration>
<enumeration value="Cprt - Cypriot"></enumeration>
<enumeration value="Cyrl - Cyrillic"></enumeration>
<enumeration value="Cyrs - Cyrillic (Old Church Slavonic
variant)"></enumeration>
<enumeration value="Deva - Devanagari (Nagari)"></enumeration>
<enumeration value="Dsrt - Deseret (Mormon)"></enumeration>
<enumeration value="Dupl - Duployan shorthand, Duployan
stenography"></enumeration>
<enumeration value="Egyd - Egyptian demotic"></enumeration>
<enumeration value="Egyh - Egyptian hieratic"></enumeration>
<enumeration value="Egyp - Egyptian hieroglyphs"></enumeration>
<enumeration value="Elba - Elbasan"></enumeration>
<enumeration value="Ethi - Ethiopic"></enumeration>
<enumeration value="Geok - Khutsuri (Asomtavruli and Nuskhuri)"></enumeration>
<enumeration value="Geor - Georgian (Mkhedruli)"></enumeration>
<enumeration value="Glag - Glagolitic"></enumeration>
<enumeration value="Goth - Gothic"></enumeration>
<enumeration value="Gran - Grantha"></enumeration>
<enumeration value="Grek - Greek"></enumeration>
<enumeration value="Gujr - Gujarati"></enumeration>
<enumeration value="Guru - Gurmukhi"></enumeration>
<enumeration value="Hanb - Han with Bopomofo"></enumeration>
<enumeration value="Hang - Hangul"></enumeration>
<enumeration value="Hani - Han (Hanzi, Kanji, Hanja)"></enumeration>

```



```

<enumeration value="Hano - Hanunoo (Hanunóo)"></enumeration>
<enumeration value="Hans - Han (Simplified variant)"></enumeration>
<enumeration value="Hant - Han (Traditional variant)"></enumeration>
<enumeration value="Hatr - Hatran"></enumeration>
<enumeration value="Hebr - Hebrew"></enumeration>
<enumeration value="Hira - Hiragana"></enumeration>
<enumeration value="Hluw - Anatolian Hieroglyphs"></enumeration>
<enumeration value="Hmng - Pahawh Hmong"></enumeration>
<enumeration value="Hrkt - Japanese syllabaries"></enumeration>
<enumeration value="Hung - Old Hungarian (Hungarian Runic)"></enumeration>
<enumeration value="Inds - Indus (Harappan)"></enumeration>
<enumeration value="Ital - Old Italic (Etruscan, Oscan etc.)"></enumeration>
<enumeration value="Jamo - Jamo"></enumeration>
<enumeration value="Java - Javanese"></enumeration>
<enumeration value="Jpan - Japanese"></enumeration>
<enumeration value="Jurc - Jurchen"></enumeration>
<enumeration value="Kali - Kayah Li"></enumeration>
<enumeration value="Kana - Katakana"></enumeration>
<enumeration value="Khar - Kharoshthi"></enumeration>
<enumeration value="Khmr - Khmer"></enumeration>
<enumeration value="Khoj - Khojki"></enumeration>
<enumeration value="Kitl - Khitan large script"></enumeration>
<enumeration value="Kits - Khitan small script"></enumeration>
<enumeration value="Knda - Kannada"></enumeration>
<enumeration value="Kore - Korean (alias for Hangul + Han)"></enumeration>
<enumeration value="Kpel - Kpelle"></enumeration>
<enumeration value="Kthi - Kaithi"></enumeration>
<enumeration value="Lana - Tai Tham (Lanna)"></enumeration>
<enumeration value="Lao - Lao"></enumeration>
<enumeration value="Latf - Latin (Fraktur variant)"></enumeration>
<enumeration value="Latg - Latin (Gaelic variant)"></enumeration>
<enumeration value="Latn - Latin"></enumeration>
<enumeration value="Leke - Leke"></enumeration>
<enumeration value="Lepc - Lepcha (Róng)"></enumeration>
<enumeration value="Limb - Limbu"></enumeration>
<enumeration value="Lina - Linear A"></enumeration>
<enumeration value="Linb - Linear B"></enumeration>
<enumeration value="Lisu - Lisu (Fraser)"></enumeration>
<enumeration value="Loma - Loma"></enumeration>
<enumeration value="Lyci - Lycian"></enumeration>
<enumeration value="Lydi - Lydian"></enumeration>
<enumeration value="Mahj - Mahajani"></enumeration>
<enumeration value="Mand - Mandaic, Mandaean"></enumeration>
<enumeration value="Mani - Manichaean"></enumeration>
<enumeration value="Marc - Marchen"></enumeration>
<enumeration value="Maya - Mayan hieroglyphs"></enumeration>
<enumeration value="Mend - Mende Kikakui"></enumeration>
<enumeration value="Merc - Meroitic Cursive"></enumeration>
<enumeration value="Mero - Meroitic Hieroglyphs"></enumeration>
<enumeration value="Mlym - Malayalam"></enumeration>
<enumeration value="Modi - Modi, Modī"></enumeration>
<enumeration value="Mong - Mongolian"></enumeration>
<enumeration value="Moon - Moon (Moon code, Moon script, Moon
type)"></enumeration>
<enumeration value="Mroo - Mro, Mru"></enumeration>
<enumeration value="Mtei - Meitei Mayek (Meithei, Meetei)"></enumeration>
<enumeration value="Mult - Multani"></enumeration>
<enumeration value="Mymr - Myanmar (Burmese)"></enumeration>

```

<enumeration value="Narb - Old North Arabian (Ancient North Arabian)"></enumeration>
 <enumeration value="Nbat - Nabataean"></enumeration>
 <enumeration value="Newa - Newa, Newar, Newari"></enumeration>
 <enumeration value="Nkgb - Nakhi Geba"></enumeration>
 <enumeration value="Nkoo - N'Ko"></enumeration>
 <enumeration value="Nshu - Nüshu"></enumeration>
 <enumeration value="Ogam - Ogham"></enumeration>
 <enumeration value="Olck - OL Chiki (OL Cemet', OL, Santali)"></enumeration>
 <enumeration value="Orkh - Old Turkic, Orkhon Runic"></enumeration>
 <enumeration value="Orya - Oriya"></enumeration>
 <enumeration value="Osge - Osage"></enumeration>
 <enumeration value="Osma - Osmanya"></enumeration>
 <enumeration value="Palm - Palmyrene"></enumeration>
 <enumeration value="Pauc - Pau Cin Hau"></enumeration>
 <enumeration value="Perm - Old Permic"></enumeration>
 <enumeration value="Phag - Phags-pa"></enumeration>
 <enumeration value="Phli - Inscriptional Pahlavi"></enumeration>
 <enumeration value="Phlp - Psalter Pahlavi"></enumeration>
 <enumeration value="Phlv - Book Pahlavi"></enumeration>
 <enumeration value="Phnx - Phoenician"></enumeration>
 <enumeration value="Piqd - Klingon (KLI pIqaD)"></enumeration>
 <enumeration value="PLrd - Miao (Pollard)"></enumeration>
 <enumeration value="Prti - Inscriptional Parthian"></enumeration>
 <enumeration value="Rjng - Rejang (Redjang, Kaganga)"></enumeration>
 <enumeration value="Roro - Rongorongo"></enumeration>
 <enumeration value="Runr - Runic"></enumeration>
 <enumeration value="Samr - Samaritan"></enumeration>
 <enumeration value="Sara - Sarati"></enumeration>
 <enumeration value="Sarb - Old South Arabian"></enumeration>
 <enumeration value="Saur - Saurashtra"></enumeration>
 <enumeration value="Sgnw - SignWriting"></enumeration>
 <enumeration value="Shaw - Shavian (Shaw)"></enumeration>
 <enumeration value="Shrd - Sharada, Śārādā"></enumeration>
 <enumeration value="Sidd - Siddham"></enumeration>
 <enumeration value="Sind - Khudawadi, Sindhi"></enumeration>
 <enumeration value="Sinh - Sinhala"></enumeration>
 <enumeration value="Sora - Sora Sompeng"></enumeration>
 <enumeration value="Sund - Sundanese"></enumeration>
 <enumeration value="Sylo - Syloti Nagri"></enumeration>
 <enumeration value="Sycr - Syriac"></enumeration>
 <enumeration value="Syre - Syriac (Estrangelo variant)"></enumeration>
 <enumeration value="Syrj - Syriac (Western variant)"></enumeration>
 <enumeration value="Syrn - Syriac (Eastern variant)"></enumeration>
 <enumeration value="Tagb - Tagbanwa"></enumeration>
 <enumeration value="Tacr - Takri"></enumeration>
 <enumeration value="Tale - Tai Le"></enumeration>
 <enumeration value="Talu - New Tai Lue"></enumeration>
 <enumeration value="Taml - Tamil"></enumeration>
 <enumeration value="Tang - Tangut"></enumeration>
 <enumeration value="Tavt - Tai Viet"></enumeration>
 <enumeration value="Telu - Telugu"></enumeration>
 <enumeration value="Teng - Tengwar"></enumeration>
 <enumeration value="Tfng - Tifinagh (Berber)"></enumeration>
 <enumeration value="Tglg - Tagalog (Baybayin, Alibata)"></enumeration>
 <enumeration value="Thaa - Thaana"></enumeration>
 <enumeration value="Thai - Thai"></enumeration>
 <enumeration value="Tibt - Tibetan"></enumeration>
 <enumeration value="Tirh - Tirhuta"></enumeration>


```

<enumeration value="Ugar - Ugaritic"></enumeration>
<enumeration value="Vaii - Vai"></enumeration>
<enumeration value="Visp - Visible Speech"></enumeration>
<enumeration value="Wara - Warang Citi (Varang Kshiti)"></enumeration>
<enumeration value="Wole - Woleai"></enumeration>
<enumeration value="Xpeo - Old Persian"></enumeration>
<enumeration value="Xsux - Cuneiform, Sumero-Akkadian"></enumeration>
<enumeration value="Yiii - Yi"></enumeration>
<enumeration value="Zinh - Code for inherited script"></enumeration>
<enumeration value="Zmth - Mathematical notation"></enumeration>
<enumeration value="Zsye - Symbols (Emoji variant)"></enumeration>
<enumeration value="Zsym - Symbols"></enumeration>
<enumeration value="Zxxx - Code for unwritten documents"></enumeration>
<enumeration value="Zyyy - Code for undetermined script"></enumeration>
<enumeration value="Zzzz - Code for uncoded script"></enumeration>
<enumeration value="other"></enumeration>
</restriction>
</simpleType>
<simpleType name="ColourDepthSimpleType">
  <restriction base="string">
    <enumeration value="bilevel"></enumeration>
    <enumeration value="greyscale"></enumeration>
    <enumeration value="colour"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="GraphicsTypeSimpleType">
  <restriction base="string">
    <enumeration value="Logo"></enumeration>
    <enumeration value="Letterhead"></enumeration>
    <enumeration value="decoration"></enumeration>
    <enumeration value="frame"></enumeration>
    <enumeration value="handwritten-annotation"></enumeration>
    <enumeration value="stamp"></enumeration>
    <enumeration value="signature"></enumeration>
    <enumeration value="barcode"></enumeration>
    <enumeration value="paper-grow"></enumeration>
    <enumeration value="punch-hole"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<simpleType name="ChartTypeSimpleType">
  <restriction base="string">
    <enumeration value="bar"></enumeration>
    <enumeration value="line"></enumeration>
    <enumeration value="pie"></enumeration>
    <enumeration value="scatter"></enumeration>
    <enumeration value="surface"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>
<complexType name="LayersType">
  <annotation>
    <documentation>
      Can be used to express the z-index of overlapping
      regions. An element with a greater z-index is always in
      front of another element with lower z-index.
    </documentation>
  </annotation>

```

```

    </annotation>
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="Layer" type="pc:LayerType"></element>
    </sequence>
</complexType>

<complexType name="LayerType">
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"></element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="zIndex" type="int" use="required"></attribute>
    <attribute name="caption" type="string"></attribute>
</complexType>

<complexType name="BaselineType">
    <attribute name="points" type="pc:PointsType" use="required"></attribute>
    <attribute name="conf" type="pc:ConfSimpleType" use="optional">
        <annotation>
            <documentation>Confidence value (between 0 and 1)</documentation>
        </annotation>
    </attribute>
</complexType>

<simpleType name="PointsType">
    <annotation>
        <documentation>Point list with format "x1,y1 x2,y2
..."</documentation>
    </annotation>
    <restriction base="string">
        <pattern value="([0-9]+,[0-9]+ )+([0-9]+,[0-9]+)"></pattern>
    </restriction>
</simpleType>

<complexType name="RelationsType">
    <annotation>
        <documentation>
            Container for one-to-one relations between layout
            objects (for example: DropCap - paragraph, caption -
            image)
        </documentation>
    </annotation>
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="Relation" type="pc:RelationType"></element>
    </sequence>
</complexType>

<complexType name="RelationType">
    <annotation>
        <documentation>
            One-to-one relation between to layout object. Use 'link'
            for loose relations and 'join' for strong relations
            (where something is fragmented for instance).

            Examples for 'link': caption - image floating -
            paragraph paragraph - paragraph (when a paragraph is
            split across columns and the last word of the first

```

paragraph DOES NOT continue in the second paragraph)
 drop-cap - paragraph (when the drop-cap is a whole word)

Examples for 'join': word - word (separated word at the end of a line) drop-cap - paragraph (when the drop-cap is not a whole word) paragraph - paragraph (when a paragraph is split across columns and the last word of the first paragraph DOES continue in the second paragraph)

```

    </documentation>
  </annotation>

  <sequence>
    <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
      <annotation>
        <documentation>Semantic labels /
tags</documentation>
      </annotation>
    </element>
    <element name="SourceRegionRef" type="pc:RegionRefType"
minOccurs="1" maxOccurs="1"></element>
    <element name="TargetRegionRef" type="pc:RegionRefType"
minOccurs="1" maxOccurs="1"></element>
  </sequence>
  <attribute name="id" type="ID" use="required"></attribute>
  <attribute name="type" use="optional">
    <simpleType>
      <restriction base="string">
        <enumeration value="link"></enumeration>
        <enumeration value="join"></enumeration>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="custom" type="string">
    <annotation>
      <documentation>For generic
use</documentation></annotation></attribute>
  <attribute name="comments" type="string"></attribute>
</complexType>

<simpleType name="ProductionSimpleType">
  <annotation>
    <documentation>Text production type</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="printed"></enumeration>
    <enumeration value="typewritten"></enumeration>
    <enumeration value="handwritten-cursive"></enumeration>
    <enumeration value="handwritten-printschrift"></enumeration>
    <enumeration value="medieval-manuscript"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<complexType name="TextStyleType">
  <annotation>
    <documentation>
      Monospace (fixed-pitch, non-proportional) or

```

```

        proportional font
    </documentation>
</annotation>
<attribute name="fontFamily" type="string">
    <annotation>
        <documentation>
            For instance: Arial, Times New Roman. Add more
            information if necessary (e.g. blackletter,
            antiqua).
        </documentation>
    </annotation>
</attribute>
<attribute name="serif" type="boolean">
    <annotation>
        <documentation>
            Serif or sans-serif typeface
        </documentation>
    </annotation>
</attribute>
<attribute name="monospace" type="boolean"></attribute>
<attribute name="fontSize" type="float">
    <annotation>
        <documentation>
            The size of the characters in points
        </documentation>
    </annotation>
</attribute>
<attribute name="xHeight" type="integer" use="optional">
    <annotation>
        <documentation>The x-height or corpus size refers to the
distance between the baseline and the mean line of lower-case letters in a
typeface. The unit is assumed to be pixels.</documentation>
    </annotation>
</attribute>
<attribute name="kerning" type="int">
    <annotation>
        <documentation>
            The degree of space (in points) between the
            characters in a string of text
        </documentation>
    </annotation>
</attribute>
<attribute name="textColour" type="pc:ColourSimpleType"></attribute>
<attribute name="textColourRgb" type="integer">
    <annotation>
        <documentation>Text colour in RGB encoded format (red value)
+ (256 x green value) + (65536 x blue value)</documentation>
    </annotation>
</attribute>
<attribute name="bgColour" type="pc:ColourSimpleType">
    <annotation>
        <documentation>Background colour</documentation>
    </annotation>
</attribute>
<attribute name="bgColourRgb" type="integer">
    <annotation>
        <documentation>Background colour in RGB encoded format (red
value) + (256 x green value) + (65536 x blue value)</documentation>
    </annotation>

```

```

</attribute>
<attribute name="reverseVideo" type="boolean">
  <annotation>
    <documentation>
      Specifies whether the colour of the text appears
      reversed against a background colour
    </documentation>
  </annotation>
</attribute>
<attribute name="bold" type="boolean"></attribute>
<attribute name="italic" type="boolean"></attribute>
<attribute name="underlined" type="boolean"></attribute>
<attribute name="subscript" type="boolean"></attribute>
<attribute name="superscript" type="boolean"></attribute>
<attribute name="strikethrough" type="boolean"></attribute>
<attribute name="smallCaps" type="boolean"></attribute>
<attribute name="LetterSpaced" type="boolean"></attribute>
</complexType>

<complexType name="RegionType" abstract="true">
  <sequence>
    <element name="AlternativeImage"
      type="pc:AlternativeImageType" minOccurs="0"
      maxOccurs="unbounded">
      <annotation>
        <documentation>
          Alternative region images (e.g.
          black-and-white)
        </documentation>
      </annotation>
    </element>
    <element name="Coords" type="pc:CoordsType"></element>
    <element name="UserDefined" type="pc:UserDefinedType"
      minOccurs="0" maxOccurs="1">
    </element>
    <element name="Labels" type="pc:LabelsType" minOccurs="0"
      maxOccurs="unbounded">
      <annotation>
        <documentation>Semantic labels /
tags</documentation>
      </annotation>
    </element>
    <element name="Roles" type="pc:RolesType" minOccurs="0"
      maxOccurs="1">
      <annotation>
        <documentation>
          Roles the region takes (e.g. in context of a
          parent region)
        </documentation>
      </annotation>
    </element>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="TextRegion"
type="pc:TextRegionType"></element>
      <element name="ImageRegion"
type="pc:ImageRegionType"></element>
      <element name="LineDrawingRegion"
type="pc:LineDrawingRegionType">
    </element>
  </choice>

```

```

        <element name="GraphicRegion"
            type="pc:GraphicRegionType">
        </element>
        <element name="TableRegion"
type="pc:TableRegionType"></element>
        <element name="ChartRegion"
type="pc:ChartRegionType"></element>
        <element name="SeparatorRegion"
            type="pc:SeparatorRegionType">
        </element>
        <element name="MathsRegion"
type="pc:MathsRegionType"></element>
        <element name="ChemRegion"
type="pc:ChemRegionType"></element>
        <element name="MusicRegion"
type="pc:MusicRegionType"></element>
        <element name="AdvertRegion"
            type="pc:AdvertRegionType">
        </element>
        <element name="NoiseRegion"
type="pc:NoiseRegionType"></element>
        <element name="UnknownRegion"
            type="pc:UnknownRegionType">
        </element>
        <element name="CustomRegion"
            type="pc:CustomRegionType">
        </element>
    </choice>
</sequence>
<attribute name="id" type="ID" use="required"></attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string"></attribute>
<attribute name="continuation" type="boolean">
    <annotation>
        <documentation>Is this region a continuation of another
region (in previous column or page, for example)?</documentation>
    </annotation></attribute>
</complexType>

<complexType name="AlternativeImageType">
    <attribute name="filename" type="string" use="required"></attribute>
    <attribute name="comments" type="string"></attribute>
    <attribute name="conf" type="pc:ConfSimpleType" use="optional">
        <annotation>
            <documentation>Confidence value (between 0 and 1)</documentation>
        </annotation>
    </attribute>
</complexType>

<simpleType name="AlignSimpleType">
    <restriction base="string">
        <enumeration value="left"></enumeration>
        <enumeration value="centre"></enumeration>
        <enumeration value="right"></enumeration>
        <enumeration value="justify"></enumeration>
    </restriction>
</simpleType>

```

```

    </restriction>
</simpleType>
<simpleType name="GroupTypeSimpleType">
  <restriction base="string">
    <enumeration value="paragraph"></enumeration>
    <enumeration value="list"></enumeration>
    <enumeration value="list-item"></enumeration>
    <enumeration value="figure"></enumeration>
    <enumeration value="article"></enumeration>
    <enumeration value="div"></enumeration>
    <enumeration value="other"></enumeration>
  </restriction>
</simpleType>

<simpleType name="TextDataTypeSimpleType">
  <restriction base="string">
    <enumeration value="xsd:decimal">
      <annotation>
        <documentation>Examples: "123.456", "+1234.456", "-
1234.456", "-.456", "-456"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:float">
      <annotation>
        <documentation>Examples: "123.456", "+1234.456", "-
1.2344e56", "-.45E-6", "INF", "-INF", "NaN"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:integer">
      <annotation>
        <documentation>Examples: "123456", "+00000012", "-1",
"-456"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:boolean">
      <annotation>
        <documentation>Examples: "true", "false", "1",
"0"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:date">
      <annotation>
        <documentation>Examples: "2001-10-26", "2001-10-
26+02:00", "2001-10-26Z", "2001-10-26+00:00", "-2001-10-26", "-20000-04-
01"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:time">
      <annotation>
        <documentation>Examples: "21:32:52", "21:32:52+02:00",
"19:32:52Z", "19:32:52+00:00", "21:32:52.12679"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:dateTime">
      <annotation>
        <documentation>Examples: "2001-10-26T21:32:52", "2001-
10-26T21:32:52+02:00", "2001-10-26T19:32:52Z", "2001-10-26T19:32:52+00:00", "-
2001-10-26T21:32:52", "2001-10-26T21:32:52.12679"</documentation>
      </annotation></enumeration>
    <enumeration value="xsd:string">
      <annotation>
        <documentation>Generic text string</documentation>
      </annotation></enumeration>
    <enumeration value="other">
      <annotation>

```

```

        <documentation>An XSD type that is not listed or a
custom type (use dataTypeDetails attribute)</documentation>
    </annotation></enumeration>
</restriction>
</simpleType>

    <!-- <simpleType name="TextMergeRuleSimpleType">
<restriction base="string">
    <enumeration value="no-whitespace">
        <annotation>
            <documentation>Concatenate text without adding
anything</documentation>
        </annotation></enumeration>
    <enumeration value="add-space">
        <annotation>
            <documentation>Adds a single space character between
the text two parts</documentation>
        </annotation></enumeration>
    <enumeration value="add-tab">
        <annotation>
            <documentation>Adds a single tab character between the
text two parts</documentation>
        </annotation></enumeration>
    <enumeration value="add-line-break">
        <annotation>
            <documentation>Adds a single line break character
between the text two parts</documentation>
        </annotation></enumeration>
    <enumeration value="add-custom">
        <annotation>
            <documentation>Adds a user-defined string between the
two text parts (specify in merge rule data)</documentation>
        </annotation></enumeration>
    <enumeration value="remove-last">
        <annotation>
            <documentation>Removes the last character of the first
text part (e.g. remove hyphen)</documentation>
        </annotation></enumeration>
    <enumeration value="remove-multiple">
        <annotation>
            <documentation>Removes the last x characters from the
first text part (specify number in merge rule data)</documentation>
        </annotation></enumeration>
    <enumeration value="remove-all">
        <annotation>
            <documentation>Ignores the first text part completely
(e.g. for drop caps that are repeated in the following
paragraph)</documentation>
        </annotation></enumeration>
    </restriction>
</simpleType> -->

<complexType name="GraphemesType">
    <annotation>
        <documentation>
            Container for graphemes, grapheme groups and
            non-printing characters
        </documentation>
    </annotation>

```



```

    <choice minOccurs="1" maxOccurs="unbounded">
      <element name="Grapheme" type="pc:GraphemeType"></element>
      <element name="NonPrintingChar"
type="pc:NonPrintingCharType"></element>
      <element name="GraphemeGroup"
type="pc:GraphemeGroupType"></element>
    </choice>
  </complexType>

  <complexType name="GraphemeBaseType">
    <annotation>
      <documentation>Base type for graphemes, grapheme groups and non-
printing characters</documentation>
    </annotation>
    <sequence>
      <element name="TextEquiv" type="pc:TextEquivType"
minOccurs="0" maxOccurs="unbounded">
      </element>
    </sequence>
    <attribute name="id" type="ID" use="required"></attribute>
    <attribute name="index" use="required">
      <annotation>
        <documentation>Order index of grapheme, group, or non-printing
character within the parent container (graphemes or glyph or grapheme
group)</documentation>
      </annotation>
      <simpleType>
        <restriction base="int">
          <minInclusive value="0"></minInclusive>
        </restriction>
      </simpleType>
    </attribute>
    <attribute name="Ligature" type="boolean"></attribute>
    <attribute name="charType">
      <annotation>
        <documentation>Type of character represented by the
grapheme/group/non-printing character element</documentation>
      </annotation>
      <simpleType>
        <restriction base="string">
          <enumeration value="base"></enumeration>
          <enumeration value="combining"></enumeration>
        </restriction>
      </simpleType>
    </attribute>
    <attribute name="custom" type="string">
      <annotation>
        <documentation>For generic
use</documentation></annotation></attribute>
      <attribute name="comments" type="string">
        <annotation>
          <documentation>For generic
use</documentation></annotation></attribute>
    </complexType>

    <complexType name="GraphemeType">
      <annotation>
        <documentation>Represents a sub-element of a glyph. Smallest
graphical unit that can be assigned a Unicode code point</documentation>

```

```

    </annotation>
    <complexContent>
      <extension base="pc:GraphemeBaseType">
        <sequence>
          <element name="Coords" type="pc:CoordsType"></element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="NonPrintingCharType">
    <annotation>
      <documentation>A glyph component without visual representation but
with Unicode code point. Non-visual / non-printing / control character. Part of
grapheme container (of glyph) or grapheme sub group.</documentation>
    </annotation>
    <complexContent>
      <extension base="pc:GraphemeBaseType"></extension>
    </complexContent>
  </complexType>

  <complexType name="GraphemeGroupType">
    <complexContent>
      <extension base="pc:GraphemeBaseType">
        <choice minOccurs="0" maxOccurs="unbounded">
          <element name="Grapheme"
type="pc:GraphemeType"></element>
          <element name="NonPrintingChar"
type="pc:NonPrintingCharType">
            </element>
        </choice>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="UserDefinedType">
    <annotation>
      <documentation>Container for user-defined
attributes</documentation>
    </annotation>
    <sequence>
      <element name="UserAttribute" type="pc:UserAttributeType"
minOccurs="1" maxOccurs="unbounded"></element>
    </sequence>
  </complexType>

  <complexType name="UserAttributeType">
    <annotation>
      <documentation>Structured custom data defined by name, type and
value.</documentation>
    </annotation>
    <attribute name="name" type="string"></attribute>
    <attribute name="description" type="string"></attribute>
    <attribute name="type">
      <simpleType>
        <restriction base="string">
          <enumeration value="xsd:string"></enumeration>
          <enumeration value="xsd:integer"></enumeration>
          <enumeration value="xsd:boolean"></enumeration>

```

```

        <enumeration value="xsd:float"></enumeration>
    </restriction>
</simpleType>
</attribute>
<attribute name="value" type="string"></attribute>
</complexType>

<complexType name="TableCellRoleType">
    <attribute name="rowIndex" type="int" use="required">
        <annotation>
            <documentation>Cell position in table starting with row
0</documentation>
        </annotation></attribute>
        <attribute name="columnIndex" type="int" use="required">
            <annotation>
                <documentation>Cell position in table starting with column
0</documentation>
            </annotation></attribute>
            <attribute name="rowSpan" type="int" use="optional">
                <annotation>
                    <documentation>Number of rows the cell spans (optional;
default is 1)</documentation>
                </annotation></attribute>
                <attribute name="colSpan" type="int" use="optional">
                    <annotation>
                        <documentation>Number of columns the cell spans (optional;
default is 1)</documentation>
                    </annotation></attribute>
                    <attribute name="header" type="boolean" use="optional">
                        <annotation>
                            <documentation>
                                Is the cell a column or row header?
                            </documentation>
                        </annotation>
                    </attribute>
                </complexType>

<complexType name="RoLesType">
    <sequence>
        <element name="TableCellRole" type="pc:TableCellRoleType"
minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation>Data for a region that takes on the
role of a table cell within a parent table region</documentation>
            </annotation></element>
        </sequence>
    </complexType>
</schema>

```

Pagecontent Format Specification – Eighth Release

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- defaults in XMLSchema: attribute use="optional" element minOccurs="1"
maxOccurs="1" abstract="false" nillable="false" -->
<schema
targetNamespace="http://schema.primaresearch.org/PAGE/gts/pagecontent/2019-07-
15"
xmlns:pc="http://schema.primaresearch.org/PAGE/gts/pagecontent/2019-07-
15"
elementFormDefault="qualified"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <element name="PcGts" type="pc:PcGtsType">
    <annotation>
      <documentation>Page Content - Ground Truth and
Storage</documentation>
    </annotation>
  </element>

  <complexType name="PcGtsType">
    <sequence>
      <element name="Metadata" type="pc:MetadataType"></element>
      <element name="Page" type="pc:PageType"></element>
    </sequence>
    <attribute name="pcGtsId" type="ID"/>
  </complexType>

  <complexType name="MetadataType">
    <sequence>
      <element name="Creator" type="string"/>
      <element name="Created" type="dateTime">
        <annotation>
          <documentation>
            The timestamp has to be in UTC (Coordinated
            Universal Time) and not local time.
          </documentation>
        </annotation>
      </element>
      <element name="LastChange" type="dateTime">
        <annotation>
          <documentation>
            The timestamp has to be in UTC
            (Coordinated Universal Time)
            and not local time.
          </documentation>
        </annotation>
      </element>
      <element name="Comments" type="string"
        minOccurs="0" maxOccurs="1">
      </element>
      <element name="UserDefined" type="pc:UserDefinedType"
        minOccurs="0" maxOccurs="1">
      </element>
      <element name="MetadataItem" type="pc:MetadataItemType"
        minOccurs="0" maxOccurs="unbounded">
      </element>
```

```

        </sequence>
        <attribute name="externalRef" type="string">
            <annotation>
                <documentation>External reference of any
kind</documentation>
            </annotation>
        </attribute>
    </complexType>

    <complexType name="MetadataItemType">
        <sequence>
            <element name="Labels" type="pc:LabelsType"
                minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <documentation>Semantic labels /
tags</documentation>
                </annotation>
            </element>
        </sequence>
        <attribute name="type">
            <annotation>
                <documentation>
                    Type of metadata (e.g. author)
                </documentation>
            </annotation>
            <simpleType>
                <restriction base="string">
                    <enumeration value="author"/>
                    <enumeration value="imageProperties"/>
                    <enumeration value="processingStep"/>
                    <enumeration value="other"/>
                </restriction>
            </simpleType>
        </attribute>
        <attribute name="name" type="string">
            <annotation>
                <documentation>
                    E.g. imagePhotometricInterpretation
                </documentation>
            </annotation>
        </attribute>
        <attribute name="value" type="string" use="required">
            <annotation>
                <documentation>E.g. RGB</documentation>
            </annotation>
        </attribute>
        <attribute name="date" type="dateTime"/>
    </complexType>
    <complexType name="LabelsType">
        <sequence>
            <element name="Label" type="pc:LabelType"
                minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <documentation>
                        A semantic label / tag
                    </documentation>
                </annotation>
            </element>
        </sequence>

```

```

    <attribute name="externalModel" type="string">
      <annotation>
        <documentation>
          Reference to external model / ontology / schema
        </documentation>
      </annotation>
    </attribute>
    <attribute name="externalId" type="string">
      <annotation>
        <documentation>
          E.g. an RDF resource identifier
          (to be used as subject or object of an RDF triple)
        </documentation>
      </annotation>
    </attribute>
    <attribute name="prefix" type="string">
      <annotation>
        <documentation>
          Prefix for all labels (e.g. first part of an URI)
        </documentation>
      </annotation>
    </attribute>
    <attribute name="comments" type="string"/>
  </complexType>
  <complexType name="LabelType">
    <annotation>
      <documentation>Semantic label</documentation>
    </annotation>
    <attribute name="value" type="string" use="required">
      <annotation>
        <documentation>
          The label / tag (e.g. 'person').
          Can be an RDF resource identifier
          (e.g. object of an RDF triple).
        </documentation>
      </annotation>
    </attribute>
    <attribute name="type" type="string">
      <annotation>
        <documentation>
          Additional information on the label
          (e.g. 'YYYY-mm-dd' for a date label).
          Can be used as predicate of an RDF triple.
        </documentation>
      </annotation>
    </attribute>
    <attribute name="comments" type="string">
  </attribute>
</complexType>
<complexType name="PageType">
  <sequence>
    <element name="AlternativeImage"
type="pc:AlternativeImageType"
      minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>
          Alternative document page images
          (e.g. black-and-white).
        </documentation>

```

```

        </annotation>
    </element>
    <element name="Border" type="pc:BorderType" minOccurs="0"
        maxOccurs="1">
    </element>
    <element name="PrintSpace" type="pc:PrintSpaceType"
        minOccurs="0" maxOccurs="1">
    </element>
    <element name="ReadingOrder" type="pc:ReadingOrderType"
        minOccurs="0" maxOccurs="1">
    <annotation>
        <documentation>Order of blocks within the
page.</documentation>
    </annotation>
    </element>
    <element name="Layers" type="pc:LayersType" minOccurs="0"
        maxOccurs="1">
    <annotation>
        <documentation>
            Unassigned regions are considered to be in the
            (virtual) default layer which is to be treated
            as below any other layers.
        </documentation>
    </annotation>
    </element>
    <element name="Relations" type="pc:RelationsType"
        minOccurs="0">
    </element>
    <element name="TextStyle" type="pc:TextStyleType"
        minOccurs="0" maxOccurs="1">
    <annotation>
        <documentation>Default text
style</documentation>
    </annotation>
    </element>
    <element name="UserDefined" type="pc:UserDefinedType"
        minOccurs="0" maxOccurs="1">
    </element>
    <element name="Labels" type="pc:LabelsType"
        minOccurs="0" maxOccurs="unbounded">
    <annotation>
        <documentation>Semantic labels /
tags</documentation>
    </annotation>
    </element>
    <choice minOccurs="0" maxOccurs="unbounded">
        <element name="TextRegion" type="pc:TextRegionType"/>
        <element name="ImageRegion"
type="pc:ImageRegionType"/>
        <element name="LineDrawingRegion"
type="pc:LineDrawingRegionType"/>
        <element name="GraphicRegion"
type="pc:GraphicRegionType"/>
        <element name="TableRegion"
type="pc:TableRegionType"/>
        <element name="ChartRegion"
type="pc:ChartRegionType"/>
        <element name="MapRegion" type="pc:MapRegionType"/>
    </choice>

```

```

        <element name="SeparatorRegion"
type="pc:SeparatorRegionType"/>
        <element name="MathsRegion"
type="pc:MathsRegionType"/>
        <element name="ChemRegion" type="pc:ChemRegionType"/>
        <element name="MusicRegion"
type="pc:MusicRegionType"/>
        <element name="AdvertRegion"
type="pc:AdvertRegionType"/>
        <element name="NoiseRegion"
type="pc:NoiseRegionType"/>
        <element name="UnknownRegion"
type="pc:UnknownRegionType"/>
        <element name="CustomRegion"
type="pc:CustomRegionType"/>
    </choice>
</sequence>
<attribute name="imageFilename" type="string" use="required">
    <annotation>
        <documentation>
extension.
            Contains the image file name including the file
            extension.
        </documentation>
    </annotation>
</attribute>
<attribute name="imageWidth" type="int" use="required">
    <annotation>
        <documentation>Specifies the width of the
image.</documentation>
    </annotation>
</attribute>
<attribute name="imageHeight" type="int" use="required">
    <annotation>
        <documentation>Specifies the height of the
image.</documentation>
    </annotation>
</attribute>
<attribute name="imageXResolution" type="float">
    <annotation>
        <documentation>Specifies the image resolution in
width.</documentation>
    </annotation>
</attribute>
<attribute name="imageYResolution" type="float">
    <annotation>
        <documentation>Specifies the image resolution in
height.</documentation>
    </annotation>
</attribute>
<attribute name="imageResolutionUnit">
    <annotation>
        <documentation>
            Specifies the unit of the resolution information
            referring to a standardised unit of measurement
            (pixels per inch, pixels per centimeter or other).
        </documentation>
    </annotation>
</simpleType>
<restriction base="string">

```



```

        <enumeration value="PPI"/>
        <enumeration value="PPCM"/>
        <enumeration value="other"/>
    </restriction>
</simpleType>
</attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="orientation" type="float">
    <annotation>
        <documentation>
            The angle the rectangle encapsulating the page
            (or its Border) has to be rotated in clockwise
direction
            in order to correct the present skew
            (negative values indicate anti-clockwise rotation).
            (The rotated image can be further referenced
            via "AlternativeImage".)
            Range: -179.999,180
        </documentation>
    </annotation>
</attribute>
<attribute name="type" type="pc:PageTypeSimpleType">
    <annotation>
        <documentation>
            The type of the page within the document
            (e.g. cover page).
        </documentation>
    </annotation>
</attribute>
<attribute name="primaryLanguage" type="pc:LanguageSimpleType">
    <annotation>
        <documentation>
            The primary language used in the page
            (lower-level definitions override the page-level
definition).
        </documentation>
    </annotation>
</attribute>
<attribute name="secondaryLanguage" type="pc:LanguageSimpleType">
    <annotation>
        <documentation>
            The secondary language used in the page
            (lower-level definitions override the page-level
definition).
        </documentation>
    </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType">
    <annotation>
        <documentation>
            The primary script used in the page
            (lower-level definitions override the page-level
definition).
        </documentation>
    </annotation>

```

```

    </attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType">
      <annotation>
        <documentation>
          The secondary script used in the page
          (lower-level definitions override the page-level
definition).
        </documentation>
      </annotation>
    </attribute>
    <attribute name="readingDirection"
type="pc:ReadingDirectionSimpleType">
      <annotation>
        <documentation>
          The direction in which text within lines
          should be read (order of words and characters),
          in addition to "textLineOrder"
          (lower-level definitions override the page-level
definition).
        </documentation>
      </annotation>
    </attribute>
    <attribute name="textLineOrder" type="pc:TextLineOrderSimpleType">
      <annotation>
        <documentation>
          The order of text lines within a block,
          in addition to "readingDirection"
          (lower-level definitions override the page-level
definition).
        </documentation>
      </annotation>
    </attribute>
    <attribute name="conf" type="pc:ConfSimpleType">
      <annotation>
        <documentation>Confidence value for whole page
(between 0 and 1)</documentation>
      </annotation>
    </attribute>
  </complexType>
  <complexType name="TextRegionType">
    <annotation>
      <documentation>
        Pure text is represented as a text region. This includes
        drop capitals, but practically ornate text may be
        considered as a graphic.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="pc:RegionType">
        <sequence>
          <element name="TextLine" type="pc:TextLineType"
            minOccurs="0" maxOccurs="unbounded"/>
          <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded"/>
          <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0" maxOccurs="1"/>
        </sequence>
        <attribute name="orientation" type="float">
          <annotation>

```

region rotation).	<pre> <documentation> The angle the rectangle encapsulating the has to be rotated in clockwise direction in order to correct the present skew (negative values indicate anti-clockwise (The rotated image can be further referenced via "AlternativeImage".) Range: -179.999,180 </documentation> </annotation> </attribute> <attribute name="type" type="pc:TextTypeSimpleType"> <annotation> <documentation> The nature of the text in the region </documentation> </annotation> </attribute> <attribute name="Leading" type="int"> <annotation> <documentation> The degree of space in points between the lines text (line spacing) </documentation> </annotation> </attribute> <attribute name="readingDirection" type="pc:ReadingDirectionSimpleType"> <annotation> <documentation> The direction in which text within lines should be read (order of words and characters), in addition to "textLineOrder". </documentation> </annotation> </attribute> <attribute name="textLineOrder" type="pc:TextLineOrderSimpleType"> <annotation> <documentation> The order of text lines within the block, in addition to "readingDirection". </documentation> </annotation> </attribute> <attribute name="readingOrientation" type="float"> <annotation> <documentation> The angle the baseline of text within the has to be rotated (relative to the rectangle encapsulating the region) in clockwise in order to correct the present skew, in addition to "orientation" </pre>
of	
region direction	

```

                                (negative values indicate anti-clockwise
rotation).
                                Range: -179.999,180
                                </documentation>
                                </annotation>
                                </attribute>
                                <attribute name="indented" type="boolean">
                                <annotation>
                                <documentation>
                                Defines whether a region of text is indented or
not
                                </documentation>
                                </annotation>
                                </attribute>
                                <attribute name="align" type="pc:AlignSimpleType">
                                <annotation>
                                <documentation>Text align</documentation>
                                </annotation>
                                </attribute>
                                <attribute name="primaryLanguage"
type="pc:LanguageSimpleType">
                                <annotation>
                                <documentation>
                                The primary language used in the region
                                </documentation>
                                </annotation>
                                </attribute>
                                <attribute name="secondaryLanguage"
type="pc:LanguageSimpleType">
                                <annotation>
                                <documentation>
                                The secondary language used in the region
                                </documentation>
                                </annotation>
                                </attribute>
                                <attribute name="primaryScript" type="pc:ScriptSimpleType">
                                <annotation>
                                <documentation>
                                The primary script used in the region
                                </documentation>
                                </annotation>
                                </attribute>
                                <attribute name="secondaryScript"
type="pc:ScriptSimpleType">
                                <annotation>
                                <documentation>
                                The secondary script used in the region
                                </documentation>
                                </annotation>
                                </attribute>
                                <attribute name="production"
type="pc:ProductionSimpleType"/>
                                </extension>
                                </complexContent>
                                </complexType>
                                <complexType name="CoordsType">
                                <attribute name="points" type="pc:PointsType" use="required">
                                <annotation>
                                <documentation>

```

```

        Polygon outline of the element as a path of points.
        No points may lie outside the outline of its parent,
        which in the case of Border is the bounding rectangle
        of the root image. Paths are closed by convention,
        i.e. the last point logically connects with the first
        (and at least 3 points are required to span an area).
        Paths must be planar (i.e. must not self-intersect).
        </documentation>
    </annotation>
</attribute>
<attribute name="conf" type="pc:ConfSimpleType">
    <annotation>
        <documentation>Confidence value (between 0 and
1)</documentation>
    </annotation>
</attribute>
</complexType>
<complexType name="TextLineType">
    <sequence>
        <element name="AlternativeImage"
type="pc:AlternativeImageType"
        minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>
                    Alternative text line images (e.g.
                    black-and-white)
                </documentation>
            </annotation>
        </element>
        <element name="Coords" type="pc:CoordsType"/>
        <element name="Baseline" type="pc:BaselineType"
            minOccurs="0">
            <annotation>
                <documentation>
                    Multiple connected points that mark the
baseline
                    of the glyphs
                </documentation>
            </annotation>
        </element>
        <element name="Word" type="pc:WordType" minOccurs="0"
            maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0">
        </element>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>

```

```

</sequence>
<attribute name="id" type="ID" use="required"/>
<attribute name="primaryLanguage" type="pc:LanguageSimpleType">
  <annotation>
    <documentation>
      Overrides primaryLanguage attribute of parent text
      region
    </documentation>
  </annotation>
</attribute>
<attribute name="primaryScript" type="pc:ScriptSimpleType">
  <annotation>
    <documentation>
      The primary script used in the text line
    </documentation>
  </annotation>
</attribute>
<attribute name="secondaryScript" type="pc:ScriptSimpleType">
  <annotation>
    <documentation>
      The secondary script used in the text line
    </documentation>
  </annotation>
</attribute>
<attribute name="readingDirection"
type="pc:ReadingDirectionSimpleType">
  <annotation>
    <documentation>
      The direction in which text within the line
      should be read (order of words and characters).
    </documentation>
  </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType">
  <annotation>
    <documentation>
      Overrides the production attribute of the parent
      text region
    </documentation>
  </annotation>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic use</documentation>
  </annotation>
</attribute>
<attribute name="comments" type="string"/>
<attribute name="index" type="int">
  <annotation>
    <documentation>
      Position (order number) of this text line within the
      parent text region.
    </documentation>
  </annotation>
</attribute>
</complexType>
<complexType name="WordType">
  <sequence>

```

```

        <element name="AlternativeImage"
type="pc:AlternativeImageType"
        minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>
                    Alternative word images (e.g.
                    black-and-white)
                </documentation>
            </annotation>
        </element>
        <element name="Coords" type="pc:CoordsType"/>
        <element name="Glyph" type="pc:GlyphType" minOccurs="0"
        maxOccurs="unbounded">
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
        minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
        minOccurs="0">
        </element>
        <element name="UserDefined" type="pc:UserDefinedType"
        minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType" minOccurs="0"
maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
    <attribute name="Language" type="pc:LanguageSimpleType">
        <annotation>
            <documentation>
                Overrides primaryLanguage attribute of parent line
                and/or text region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="primaryScript" type="pc:ScriptSimpleType">
        <annotation>
            <documentation>
                The primary script used in the word
            </documentation>
        </annotation>
    </attribute>
    <attribute name="secondaryScript" type="pc:ScriptSimpleType">
        <annotation>
            <documentation>
                The secondary script used in the word
            </documentation>
        </annotation>
    </attribute>
    <attribute name="readingDirection"
type="pc:ReadingDirectionSimpleType">
        <annotation>
            <documentation>
                The direction in which text within the word

```

```

        should be read (order of characters).
    </documentation>
</annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType">
    <annotation>
        <documentation>
            Overrides the production attribute of the parent
            text line and/or text region.
        </documentation>
    </annotation>
</attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string"/>
</complexType>
<complexType name="GlyphType">
    <sequence>
        <element name="AlternativeImage"
type="pc:AlternativeImageType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>
                    Alternative glyph images (e.g.
                    black-and-white)
                </documentation>
            </annotation>
        </element>
        <element name="Coords" type="pc:CoordsType"/>
        <element name="Graphemes" type="pc:GraphemesType"
            minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation>
                    Container for graphemes, grapheme groups and
                    non-printing characters
                </documentation>
            </annotation>
        </element>
        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
        <element name="TextStyle" type="pc:TextStyleType"
            minOccurs="0">
        </element>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>
    </sequence>
<attribute name="id" type="ID" use="required"/>

```



```

<attribute name="ligature" use="optional" type="boolean"/>
<attribute name="symbol" use="optional" type="boolean"/>
<attribute name="script" type="pc:ScriptSimpleType">
  <annotation>
    <documentation>
      The script used for the glyph
    </documentation>
  </annotation>
</attribute>
<attribute name="production" type="pc:ProductionSimpleType">
  <annotation>
    <documentation>
      Overrides the production attribute of the parent
      word / text line / text region.
    </documentation>
  </annotation>
</attribute>
<attribute name="custom" type="string">
  <annotation>
    <documentation>For generic use</documentation>
  </annotation>
</attribute>
<attribute name="comments" type="string"/>
</complexType>
<complexType name="TextEquivType">
  <sequence>
    <element name="PlainText" type="string" minOccurs="0">
      <annotation>
        <documentation>
          Text in a "simple" form (ASCII or extended
          as mostly used for typing). I.e. no use of
          special characters for ligatures (should be
          stored as two separate characters) etc.
        </documentation>
      </annotation>
    </element>
    <element name="Unicode" type="string">
      <annotation>
        <documentation>
          Correct encoding of the original, always using
          the corresponding Unicode code point. I.e.
          ligatures have to be represented as one
          character etc.
        </documentation>
      </annotation>
    </element>
  </sequence>
  <attribute name="index" use="optional">
    <annotation>
      <documentation>
        Used for sort order in case multiple TextEquivs are
        defined.
        The text content with the lowest index should be
        interpreted
        as the main text content.
      </documentation>
    </annotation>
  </simpleType>

```

```

        <restriction base="integer">
            <minInclusive value="0"></minInclusive>
        </restriction>
    </simpleType>
</attribute>
<attribute name="conf" type="pc:ConfSimpleType">
    <annotation>
        <documentation>OCR confidence value (between 0 and
1)</documentation>
    </annotation>
</attribute>
<attribute name="dataType" type="pc:TextDataTypeSimpleType">
    <annotation>
        <documentation>
            Type of text content (is it free text or a number, for
instance).

            This is only a descriptive attribute, the text type
            is not checked during XML validation.
        </documentation>
    </annotation>
</attribute>
<attribute name="dataTypeDetails" type="string">
    <annotation>
        <documentation>
            Refinement for dataType attribute. Can be a regular
expression, for instance.
        </documentation>
    </annotation>
</attribute>
<!-- <attribute name="mergeWithNextRule"
type="pc:TextMergeRuleSimpleType">
    <annotation>
        <documentation>Rule for merging consecutive
text objects. The rule applies to the first object of a pair (i.e. 'remove-last'
removes the last
character of the first region, can be used to remove hyphen, for
example)</documentation>
    </annotation>
</attribute>
<attribute name="mergeWithNextRuleData" type="string">
    <annotation>
        <documentation>Custom data for mergeRule
attribute. Can number of characters to be removed, for example.</documentation>
    </annotation>
</attribute> -->
<attribute name="comments" type="string"/>
</complexType>
<complexType name="ImageRegionType">
    <annotation>
        <documentation>
            An image is considered to be more intricate and complex
            than a graphic. These can be photos or drawings.
        </documentation>
    </annotation>
<complexContent>
    <extension base="pc:RegionType">
        <attribute name="orientation" type="float">
            <annotation>
                <documentation>

```

region

rotation).

The angle the rectangle encapsulating a
has to be rotated in clockwise direction
in order to correct the present skew
(negative values indicate anti-clockwise
rotation).

Range: -179.999,180
</documentation>
</annotation>
</attribute>
<attribute name="colourDepth"
type="pc:ColourDepthSimpleType">
<annotation>
<documentation>
The colour bit depth required for the
region
</documentation>
</annotation>
</attribute>
<attribute name="bgColour" type="pc:ColourSimpleType">
<annotation>
<documentation>
The background colour of the region
</documentation>
</annotation>
</attribute>
<attribute name="embText" type="boolean">
<annotation>
<documentation>
Specifies whether the region also
contains
text
</documentation>
</annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="LineDrawingRegionType">
<annotation>
<documentation>
A line drawing is a single colour illustration without
solid areas.
</documentation>
</annotation>
<complexContent>
<extension base="pc:RegionType">
<attribute name="orientation" type="float">
<annotation>
<documentation>
The angle the rectangle encapsulating a
region
has to be rotated in clockwise direction
in order to correct the present skew
(negative values indicate anti-clockwise
rotation).

Range: -179.999,180
</documentation>
</annotation>

```

        </attribute>
        <attribute name="penColour"
type="pc:ColourSimpleType">
            <annotation>
                <documentation>
                    The pen (foreground) colour of the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType">
            <annotation>
                <documentation>
                    The background colour of the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="embText" type="boolean">
            <annotation>
                <documentation>
                    Specifies whether the region also
contains
                    text
                </documentation>
            </annotation>
        </attribute>
    </extension>
</complexContent>
</complexType>
<complexType name="GraphicRegionType">
    <annotation>
        <documentation>
            Regions containing simple graphics, such as a company
            logo, should be marked as graphic regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float">
                <annotation>
                    <documentation>
                        The angle the rectangle encapsulating a
region
                        has to be rotated in clockwise direction
                        in order to correct the present skew
                        (negative values indicate anti-clockwise
rotation).
                        Range: -179.999,180
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="type"
type="pc:GraphicsTypeSimpleType">
                <annotation>
                    <documentation>
                        The type of graphic in the region
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="numColours" type="int">

```

```

        <annotation>
            <documentation>
                An approximation of the number of colours
                used in the region
            </documentation>
        </annotation>
    </attribute>
    <attribute name="embText" type="boolean">
        <annotation>
            <documentation>
                Specifies whether the region also
                text.
            </documentation>
        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
<complexType name="TableRegionType">
    <annotation>
        <documentation>
            Tabular data in any form is represented with a table
            region. Rows and columns may or may not have separator
            lines; these lines are not separator regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <sequence>
                <element name="Grid" type="pc:GridType"
                    minOccurs="0" maxOccurs="1">
                    <annotation>
                        <documentation>Table grid (visible
or virtual grid lines)</documentation>
                    </annotation>
                </element>
            </sequence>
            <attribute name="orientation" type="float">
                <annotation>
                    <documentation>
                        The angle the rectangle encapsulating a
                        region
                        has to be rotated in clockwise direction
                        in order to correct the present skew
                        (negative values indicate anti-clockwise
rotation).
                        Range: -179.999,180
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="rows" type="int">
                <annotation>
                    <documentation>
                        The number of rows present in the table
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="columns" type="int">

```

	<pre> <annotation> <documentation> The number of columns present in the </documentation> </annotation> </attribute> <attribute name="LineColour" type="pc:ColourSimpleType"> </pre>
region	<pre> <annotation> <documentation> The colour of the lines used in the </documentation> </annotation> </attribute> <attribute name="bgColour" type="pc:ColourSimpleType"> <annotation> <documentation> The background colour of the region </documentation> </annotation> </attribute> <attribute name="lineSeparators" type="boolean"> <annotation> <documentation> Specifies the presence of line separators </documentation> </annotation> </attribute> <attribute name="embText" type="boolean"> <annotation> <documentation> Specifies whether the region also </documentation> </annotation> </pre>
contains	<pre> text </documentation> </annotation> </attribute> </extension> </complexContent> </complexType> <complexType name="GridType"> <annotation> <documentation> Matrix of grid points defining the table grid on the page. </documentation> </annotation> <sequence> <element name="GridPoints" type="pc:GridPointsType" minOccurs="2" maxOccurs="unbounded"> <annotation> <documentation> One row in the grid point matrix. Points with x,y coordinates. (note: for a table with n table rows there </pre>
should be n+1 grid rows)	<pre> </documentation> </annotation> </sequence> </complexType> </pre>

```

        </element>
    </sequence>
</complexType>
<complexType name="GridPointsType">
    <annotation>
        <documentation>Points with x,y coordinates.</documentation>
    </annotation>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>
                The grid row index
            </documentation>
        </annotation>
    </attribute>
    <attribute name="points" type="pc:PointsType"
        use="required"/>
</complexType>
<complexType name="ChartRegionType">
    <annotation>
        <documentation>
            Regions containing charts or graphs of any type, should
            be marked as chart regions.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float">
                <annotation>
                    <documentation>
                        The angle the rectangle encapsulating a
region
                        has to be rotated in clockwise direction
                        in order to correct the present skew
rotation).
                        (negative values indicate anti-clockwise
                        rotation).
                    </documentation>
                    Range: -179.999,180
                </documentation>
            </annotation>
        </attribute>
        <attribute name="type" type="pc:ChartTypeSimpleType">
            <annotation>
                <documentation>
                    The type of chart in the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="numColours" type="int">
            <annotation>
                <documentation>
                    An approximation of the number of colours
                    used in the region
                </documentation>
            </annotation>
        </attribute>
        <attribute name="bgColour" type="pc:ColourSimpleType">
            <annotation>
                <documentation>
                    The background colour of the region
                </documentation>
            </annotation>
        </attribute>
    </complexContent>
</extension>
</complexType>

```

	<pre> </annotation> </attribute> <attribute name="embText" type="boolean"> <annotation> <documentation> Specifies whether the region also </pre>
contains	<pre> text </documentation> </annotation> </attribute> </extension> </complexContent> </complexType> <complexType name="SeparatorRegionType"> <annotation> <documentation> Separators are lines that lie between columns and paragraphs and can be used to logically separate different articles from each other. </documentation> </annotation> <complexContent> <extension base="pc:RegionType"> <attribute name="orientation" type="float"> <annotation> <documentation> The angle the rectangle encapsulating a </pre>
region	<pre> has to be rotated in clockwise direction in order to correct the present skew (negative values indicate anti-clockwise </pre>
rotation).	<pre> Range: -179.999,180 </documentation> </annotation> </attribute> <attribute name="colour" type="pc:ColourSimpleType"> <annotation> <documentation> The colour of the separator </documentation> </annotation> </attribute> </extension> </complexContent> </complexType> <complexType name="MathsRegionType"> <annotation> <documentation> Regions containing equations and mathematical symbols should be marked as maths regions. </documentation> </annotation> <complexContent> <extension base="pc:RegionType"> <attribute name="orientation" type="float"> <annotation> <documentation> </pre>

region

The angle the rectangle encapsulating a

has to be rotated in clockwise direction

in order to correct the present skew

(negative values indicate anti-clockwise

rotation).

Range: -179.999,180

```

</documentation>
</annotation>
</attribute>
<attribute name="bgColour" type="pc:ColourSimpleType">
  <annotation>
    <documentation>
      The background colour of the region
    </documentation>
  </annotation>
</attribute>
</extension>
</complexContent>
</complexType>
<complexType name="ChemRegionType">
  <annotation>
    <documentation>
      Regions containing chemical formulas.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float">
        <annotation>
          <documentation>
            The angle the rectangle encapsulating a
            region has to be rotated in clockwise
            direction in order to correct the present
            skew (negative values indicate
            anti-clockwise rotation). Range:
            -179.999,180
          </documentation>
        </annotation>
      </attribute>
      <attribute name="bgColour" type="pc:ColourSimpleType">
        <annotation>
          <documentation>
            The background colour of the region
          </documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<complexType name="MapRegionType">
  <annotation>
    <documentation>
      Regions containing maps.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="pc:RegionType">
      <attribute name="orientation" type="float">

```

```

        <annotation>
            <documentation>
                The angle the rectangle encapsulating a
                region has to be rotated in clockwise
                direction in order to correct the present
                skew (negative values indicate
                anti-clockwise rotation). Range:
                -179.999,180
            </documentation>
        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
<complexType name="MusicRegionType">
    <annotation>
        <documentation>
            Regions containing musical notations.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float">
                <annotation>
                    <documentation>
                        The angle the rectangle encapsulating a
region
                        has to be rotated in clockwise direction
rotation).
                        in order to correct the present skew
                        (negative values indicate anti-clockwise
                        Range: -179.999,180
                    </documentation>
                </annotation>
            </attribute>
            <attribute name="bgColour" type="pc:ColourSimpleType">
                <annotation>
                    <documentation>
                        The background colour of the region
                    </documentation>
                </annotation>
            </attribute>
        </extension>
    </complexContent>
</complexType>
<complexType name="AdvertRegionType">
    <annotation>
        <documentation>
            Regions containing advertisements.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:RegionType">
            <attribute name="orientation" type="float">
                <annotation>
                    <documentation>
region
                        The angle the rectangle encapsulating a
                        has to be rotated in clockwise direction

```

rotation).

```

                                in order to correct the present skew
                                (negative values indicate anti-clockwise
                                rotation).

                                Range: -179.999,180
                                </documentation>
                            </annotation>
                        </attribute>
                        <attribute name="bgColour" type="pc:ColourSimpleType">
                            <annotation>
                                <documentation>
                                    The background colour of the region
                                </documentation>
                            </annotation>
                        </attribute>
                    </extension>
                </complexContent>
            </complexType>
            <complexType name="NoiseRegionType">
                <annotation>
                    <documentation>
                        Noise regions are regions where no real data lies, only
                        false data created by artifacts on the document or
                        scanner noise.
                    </documentation>
                </annotation>
                <complexContent>
                    <extension base="pc:RegionType"></extension>
                </complexContent>
            </complexType>
            <complexType name="UnknownRegionType">
                <annotation>
                    <documentation>
                        To be used if the region type cannot be ascertained.
                    </documentation>
                </annotation>
                <complexContent>
                    <extension base="pc:RegionType"></extension>
                </complexContent>
            </complexType>
            <complexType name="CustomRegionType">
                <annotation>
                    <documentation>
                        Regions containing content that is not covered
                        by the default types (text, graphic, image,
                        line drawing, chart, table, separator, maths,
                        map, music, chem, advert, noise, unknown).
                    </documentation>
                </annotation>
                <complexContent>
                    <extension base="pc:RegionType">
                        <attribute name="type" type="string">
                            <annotation>
                                <documentation>
                                    Information on the type of content
                                    represented by this region
                                </documentation>
                            </annotation>
                        </attribute>
                    </extension>
                </complexContent>
            </complexType>
        </schema>
    </pre>
```

```

        </complexContent>
    </complexType>
    <complexType name="PrintSpaceType">
        <annotation>
            <documentation>
                Determines the effective area on the paper of a printed
page.

                Its size is equal for all pages of a book
                (exceptions: titlepage, multipage pictures).
                It contains all living elements (except marginals)
                like body type, footnotes, headings, running titles.
                It does not contain pagenumber (if not part of running
title),
                marginals, signature mark, preview words.
            </documentation>
        </annotation>
        <sequence>
            <element name="Coords" type="pc:CoordsType"/>
        </sequence>
    </complexType>
    <complexType name="ReadingOrderType">
        <annotation>
            <documentation>
                Definition of the reading order within the page.
                To express a reading order between elements
                they have to be included in an OrderedGroup.
                Groups may contain further groups.
            </documentation>
        </annotation>
        <choice minOccurs="1" maxOccurs="1">
            <element name="OrderedGroup" type="pc:OrderedGroupType"/>
            <element name="UnorderedGroup"
type="pc:UnorderedGroupType"/>
        </choice>
        <attribute name="conf" type="pc:ConfSimpleType">
            <annotation>
                <documentation>Confidence value (between 0 and
1)</documentation>
            </annotation>
        </attribute>
    </complexType>
    <complexType name="RegionRefIndexedType">
        <annotation>
            <documentation>Numbered region</documentation>
        </annotation>
        <attribute name="index" type="int" use="required">
            <annotation>
                <documentation>Position (order number) of this item
within the current hierarchy level.</documentation>
            </annotation>
        </attribute>
        <attribute name="regionRef" type="IDREF" use="required"/>
    </complexType>
    <complexType name="OrderedGroupIndexedType">
        <annotation>
            <documentation>
                Indexed group containing ordered elements
            </documentation>
        </annotation>

```

```

    <sequence>
      <element name="UserDefined" type="pc:UserDefinedType"
        minOccurs="0" maxOccurs="1"/>
      <element name="Labels" type="pc:LabelsType"
        minOccurs="0" maxOccurs="unbounded">
        <annotation>
          <documentation>Semantic labels /
tags</documentation>
          </annotation>
        </element>
        <choice minOccurs="1" maxOccurs="unbounded">
          <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"/>
          <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"/>
          <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"/>
        </choice>
      </sequence>
      <attribute name="id" type="ID" use="required"/>
      <attribute name="regionRef" type="IDREF">
        <annotation>
          <documentation>
Optional link to a parent region of nested regions.
The parent region doubles as reading order group.
Only the nested regions should be allowed as group
members.
          </documentation>
        </annotation>
      </attribute>
      <attribute name="index" type="int" use="required">
        <annotation>
          <documentation>
Position (order number) of this item within the
current hierarchy level.
          </documentation>
        </annotation>
      </attribute>
      <attribute name="caption" type="string"/>
      <attribute name="type" type="pc:GroupTypeSimpleType"/>
      <attribute name="continuation" type="boolean">
        <annotation>
          <documentation>
Is this group a continuation of another group (from
previous column or page, for example)?
          </documentation>
        </annotation>
      </attribute>
      <attribute name="custom" type="string">
        <annotation>
          <documentation>For generic use</documentation>
        </annotation>
      </attribute>
      <attribute name="comments" type="string"/>
    </complexType>
    <complexType name="UnorderedGroupIndexedType">
      <annotation>
        <documentation>
Indexed group containing unordered elements

```

```

        </documentation>
    </annotation>
    <sequence>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef" type="pc:RegionRefType"/>
            <element name="OrderedGroup"
type="pc:OrderedGroupType"/>
            <element name="UnorderedGroup"
type="pc:UnorderedGroupType"/>
        </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
    <attribute name="regionRef" type="IDREF">
        <annotation>
            <documentation>
                Optional link to a parent region of nested regions.
                The parent region doubles as reading order group.
                Only the nested regions should be allowed as group
members.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="index" type="int" use="required">
        <annotation>
            <documentation>
                Position (order number) of this item within the
                current hierarchy level.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="caption" type="string"/>
    <attribute name="type" type="pc:GroupTypeSimpleType"/>
    <attribute name="continuation" type="boolean">
        <annotation>
            <documentation>
                Is this group a continuation of another group
                (from previous column or page, for example)?
            </documentation>
        </annotation>
    </attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic use</documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string"/>
</complexType>
<complexType name="RegionRefType">
    <attribute name="regionRef" type="IDREF" use="required"/>

```

```

</complexType>
<complexType name="OrderedGroupType">
  <annotation>
    <documentation>
      Numbered group (contains ordered elements)
    </documentation>
  </annotation>
  <sequence>
    <element name="UserDefined" type="pc:UserDefinedType"
      minOccurs="0" maxOccurs="1">
    </element>
    <element name="Labels" type="pc:LabelsType"
      minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>Semantic labels /
tags</documentation>
      </annotation>
    </element>
    <choice minOccurs="1" maxOccurs="unbounded">
      <element name="RegionRefIndexed"
type="pc:RegionRefIndexedType"/>
      <element name="OrderedGroupIndexed"
type="pc:OrderedGroupIndexedType"/>
      <element name="UnorderedGroupIndexed"
type="pc:UnorderedGroupIndexedType"/>
    </choice>
  </sequence>
  <attribute name="id" type="ID" use="required"/>
  <attribute name="regionRef" type="IDREF">
    <annotation>
      <documentation>
        Optional link to a parent region of nested regions.
        The parent region doubles as reading order group.
        Only the nested regions should be allowed as group
members.
      </documentation>
    </annotation>
  </attribute>
  <attribute name="caption" type="string"/>
  <attribute name="type" type="pc:GroupTypeSimpleType"/>
  <attribute name="continuation" type="boolean">
    <annotation>
      <documentation>
        Is this group a continuation of another group
        (from previous column or page, for example)?
      </documentation>
    </annotation>
  </attribute>
  <attribute name="custom" type="string">
    <annotation>
      <documentation>For generic use</documentation>
    </annotation>
  </attribute>
  <attribute name="comments" type="string"/>
</complexType>
<complexType name="UnorderedGroupType">
  <annotation>
    <documentation>
      Numbered group (contains unordered elements)

```

```

        </documentation>
    </annotation>
    <sequence>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>
        <choice minOccurs="1" maxOccurs="unbounded">
            <element name="RegionRef" type="pc:RegionRefType"/>
            <element name="OrderedGroup"
type="pc:OrderedGroupType"/>
            <element name="UnorderedGroup"
type="pc:UnorderedGroupType"/>
        </choice>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
    <attribute name="regionRef" type="IDREF">
        <annotation>
            <documentation>
                Optional link to a parent region of nested regions.
                The parent region doubles as reading order group.
                Only the nested regions should be allowed as group
members.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="caption" type="string"/>
    <attribute name="type" type="pc:GroupTypeSimpleType"/>
    <attribute name="continuation" type="boolean">
        <annotation>
            <documentation>
                Is this group a continuation of another group
                (from previous column or page, for example)?
            </documentation>
        </annotation>
    </attribute>
    <attribute name="custom" type="string">
        <annotation>
            <documentation>For generic use</documentation>
        </annotation>
    </attribute>
    <attribute name="comments" type="string"/>
</complexType>
<complexType name="BorderType">
    <annotation>
        <documentation>
            Border of the actual page (if the scanned image
            contains parts not belonging to the page).
        </documentation>
    </annotation>
    <sequence>
        <element name="Coords" type="pc:CoordsType"/>
    </sequence>

```



```

</complexType>
<simpleType name="ColourSimpleType">
  <restriction base="string">
    <enumeration value="black"/>
    <enumeration value="blue"/>
    <enumeration value="brown"/>
    <enumeration value="cyan"/>
    <enumeration value="green"/>
    <enumeration value="grey"/>
    <enumeration value="indigo"/>
    <enumeration value="magenta"/>
    <enumeration value="orange"/>
    <enumeration value="pink"/>
    <enumeration value="red"/>
    <enumeration value="turquoise"/>
    <enumeration value="violet"/>
    <enumeration value="white"/>
    <enumeration value="yellow"/>
    <enumeration value="other"/>
  </restriction>
</simpleType>
<simpleType name="ReadingDirectionSimpleType">
  <restriction base="string">
    <enumeration value="left-to-right"/>
    <enumeration value="right-to-left"/>
    <enumeration value="top-to-bottom"/>
    <enumeration value="bottom-to-top"/>
  </restriction>
</simpleType>
<simpleType name="TextLineOrderSimpleType">
  <restriction base="string">
    <enumeration value="top-to-bottom"/>
    <enumeration value="bottom-to-top"/>
    <enumeration value="left-to-right"/>
    <enumeration value="right-to-left"/>
  </restriction>
</simpleType>
<simpleType name="TextTypeSimpleType">
  <restriction base="string">
    <enumeration value="paragraph"/>
    <enumeration value="heading"/>
    <enumeration value="caption"/>
    <enumeration value="header"/>
    <enumeration value="footer"/>
    <enumeration value="page-number"/>
    <enumeration value="drop-capital"/>
    <enumeration value="credit"/>
    <enumeration value="floating"/>
    <enumeration value="signature-mark"/>
    <enumeration value="catch-word"/>
    <enumeration value="marginalia"/>
    <enumeration value="footnote"/>
    <enumeration value="footnote-continued"/>
    <enumeration value="endnote"/>
    <enumeration value="TOC-entry"/>
    <enumeration value="List-Label"/>
    <enumeration value="other"/>
  </restriction>
</simpleType>

```

```

<simpleType name="PageTypeSimpleType">
  <restriction base="string">
    <enumeration value="front-cover"/>
    <enumeration value="back-cover"/>
    <enumeration value="title"/>
    <enumeration value="table-of-contents"/>
    <enumeration value="index"/>
    <enumeration value="content"/>
    <enumeration value="blank"/>
    <enumeration value="other"/>
  </restriction>
</simpleType>
<simpleType name="ConfSimpleType">
  <restriction base="float">
    <minInclusive value="0"/></minInclusive>
    <maxInclusive value="1"/></maxInclusive>
  </restriction>
</simpleType>
<simpleType name="LanguageSimpleType">
  <annotation>
    <documentation>ISO 639.x 2016-07-14</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="Abkhaz"/>
    <enumeration value="Afar"/>
    <enumeration value="Afrikaans"/>
    <enumeration value="Akan"/>
    <enumeration value="Albanian"/>
    <enumeration value="Amharic"/>
    <enumeration value="Arabic"/>
    <enumeration value="Aragonese"/>
    <enumeration value="Armenian"/>
    <enumeration value="Assamese"/>
    <enumeration value="Avaric"/>
    <enumeration value="Avestan"/>
    <enumeration value="Aymara"/>
    <enumeration value="Azerbaijani"/>
    <enumeration value="Bambara"/>
    <enumeration value="Bashkir"/>
    <enumeration value="Basque"/>
    <enumeration value="Belarusian"/>
    <enumeration value="Bengali"/>
    <enumeration value="Bihari"/>
    <enumeration value="Bislama"/>
    <enumeration value="Bosnian"/>
    <enumeration value="Breton"/>
    <enumeration value="Bulgarian"/>
    <enumeration value="Burmese"/>
    <enumeration value="Cambodian"/>
    <enumeration value="Cantonese"/>
    <enumeration value="Catalan"/>
    <enumeration value="Chamorro"/>
    <enumeration value="Chechen"/>
    <enumeration value="Chichewa"/>
    <enumeration value="Chinese"/>
    <enumeration value="Chuvash"/>
    <enumeration value="Cornish"/>
    <enumeration value="Corsican"/>
    <enumeration value="Cree"/>
  </restriction>
</simpleType>

```

```

<enumeration value="Croatian"/>
<enumeration value="Czech"/>
<enumeration value="Danish"/>
<enumeration value="Divehi"/>
<enumeration value="Dutch"/>
<enumeration value="Dzongkha"/>
<enumeration value="English"/>
<enumeration value="Esperanto"/>
<enumeration value="Estonian"/>
<enumeration value="Ewe"/>
<enumeration value="Faroese"/>
<enumeration value="Fijian"/>
<enumeration value="Finnish"/>
<enumeration value="French"/>
<enumeration value="Fula"/>
<enumeration value="Gaelic"/>
<enumeration value="Galician"/>
<enumeration value="Ganda"/>
<enumeration value="Georgian"/>
<enumeration value="German"/>
<enumeration value="Greek"/>
<enumeration value="Guarani"/>
<enumeration value="Gujarati"/>
<enumeration value="Haitian"/>
<enumeration value="Hausa"/>
<enumeration value="Hebrew"/>
<enumeration value="Herero"/>
<enumeration value="Hindi"/>
<enumeration value="Hiri Motu"/>
<enumeration value="Hungarian"/>
<enumeration value="Icelandic"/>
<enumeration value="Ido"/>
<enumeration value="Igbo"/>
<enumeration value="Indonesian"/>
<enumeration value="InterLingua"/>
<enumeration value="Interlingue"/>
<enumeration value="Inuktitut"/>
<enumeration value="Inupiaq"/>
<enumeration value="Irish"/>
<enumeration value="Italian"/>
<enumeration value="Japanese"/>
<enumeration value="Javanese"/>
<enumeration value="Kalaallisut"/>
<enumeration value="Kannada"/>
<enumeration value="Kanuri"/>
<enumeration value="Kashmiri"/>
<enumeration value="Kazakh"/>
<enumeration value="Khmer"/>
<enumeration value="Kikuyu"/>
<enumeration value="Kinyarwanda"/>
<enumeration value="Kirundi"/>
<enumeration value="Komi"/>
<enumeration value="Kongo"/>
<enumeration value="Korean"/>
<enumeration value="Kurdish"/>
<enumeration value="Kwanyama"/>
<enumeration value="Kyrgyz"/>
<enumeration value="Lao"/>
<enumeration value="Latin"/>

```

```

<enumeration value="Latvian"/>
<enumeration value="Limburgish"/>
<enumeration value="Lingala"/>
<enumeration value="Lithuanian"/>
<enumeration value="Luba-Katanga"/>
<enumeration value="Luxembourgish"/>
<enumeration value="Macedonian"/>
<enumeration value="Malagasy"/>
<enumeration value="Malay"/>
<enumeration value="Malayalam"/>
<enumeration value="Maltese"/>
<enumeration value="Manx"/>
<enumeration value="Māori"/>
<enumeration value="Marathi"/>
<enumeration value="Marshallese"/>
<enumeration value="Mongolian"/>
<enumeration value="Nauru"/>
<enumeration value="Navajo"/>
<enumeration value="Ndonga"/>
<enumeration value="Nepali"/>
<enumeration value="North Ndebele"/>
<enumeration value="Northern Sami"/>
<enumeration value="Norwegian"/>
<enumeration value="Norwegian Bokmål"/>
<enumeration value="Norwegian Nynorsk"/>
<enumeration value="Nuosu"/>
<enumeration value="Occitan"/>
<enumeration value="Ojibwe"/>
<enumeration value="Old Church Slavonic"/>
<enumeration value="Oriya"/>
<enumeration value="Oromo"/>
<enumeration value="Ossetian"/>
<enumeration value="Pāli"/>
<enumeration value="Panjabi"/>
<enumeration value="Pashto"/>
<enumeration value="Persian"/>
<enumeration value="Polish"/>
<enumeration value="Portuguese"/>
<enumeration value="Punjabi"/>
<enumeration value="Quechua"/>
<enumeration value="Romanian"/>
<enumeration value="Romansh"/>
<enumeration value="Russian"/>
<enumeration value="Samoan"/>
<enumeration value="Sango"/>
<enumeration value="Sanskrit"/>
<enumeration value="Sardinian"/>
<enumeration value="Serbian"/>
<enumeration value="Shona"/>
<enumeration value="Sindhi"/>
<enumeration value="Sinhala"/>
<enumeration value="Slovak"/>
<enumeration value="Slovene"/>
<enumeration value="Somali"/>
<enumeration value="South Ndebele"/>
<enumeration value="Southern Sotho"/>
<enumeration value="Spanish"/>
<enumeration value="Sundanese"/>
<enumeration value="Swahili"/>

```

```

<enumeration value="Swati"/>
<enumeration value="Swedish"/>
<enumeration value="Tagalog"/>
<enumeration value="Tahitian"/>
<enumeration value="Tajik"/>
<enumeration value="Tamil"/>
<enumeration value="Tatar"/>
<enumeration value="Telugu"/>
<enumeration value="Thai"/>
<enumeration value="Tibetan"/>
<enumeration value="Tigrinya"/>
<enumeration value="Tonga"/>
<enumeration value="Tsonga"/>
<enumeration value="Tswana"/>
<enumeration value="Turkish"/>
<enumeration value="Turkmen"/>
<enumeration value="Twi"/>
<enumeration value="Uighur"/>
<enumeration value="Ukrainian"/>
<enumeration value="Urdu"/>
<enumeration value="Uzbek"/>
<enumeration value="Venda"/>
<enumeration value="Vietnamese"/>
<enumeration value="Volapük"/>
<enumeration value="Walloon"/>
<enumeration value="Welsh"/>
<enumeration value="Western Frisian"/>
<enumeration value="Wolof"/>
<enumeration value="Xhosa"/>
<enumeration value="Yiddish"/>
<enumeration value="Yoruba"/>
<enumeration value="Zhuang"/>
<enumeration value="Zulu"/>
<enumeration value="other"/>
</restriction>
</simpleType>
<simpleType name="ScriptSimpleType">
  <annotation>
    <documentation>iso15924 2016-07-14</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="Adlm - Adlam"/>
    <enumeration value="Afak - Afaka"/>
    <enumeration value="Aghb - Caucasian Albanian"/>
    <enumeration value="Ahom - Ahom, Tai Ahom"/>
    <enumeration value="Arab - Arabic"/>
    <enumeration value="Aran - Arabic (Nastaliq variant)"/>
    <enumeration value="Armi - Imperial Aramaic"/>
    <enumeration value="Armn - Armenian"/>
    <enumeration value="Avst - Avestan"/>
    <enumeration value="Bali - Balinese"/>
    <enumeration value="Bamu - Bamum"/>
    <enumeration value="Bass - Bassa Vah"/>
    <enumeration value="Batk - Batak"/>
    <enumeration value="Beng - Bengali"/>
    <enumeration value="Bhks - Bhaiksuki"/>
    <enumeration value="Blis - Blissymbols"/>
    <enumeration value="Bopo - Bopomofo"/>
    <enumeration value="Brah - Brahmi"/>
  </restriction>
</simpleType>

```

```

<enumeration value="Brai - Braille"/>
<enumeration value="Bugi - Buginese"/>
<enumeration value="Buhd - Buhid"/>
<enumeration value="Cakm - Chakma"/>
<enumeration value="Cans - Unified Canadian Aboriginal
Syllabics"/>
<enumeration value="Cari - Carian"/>
<enumeration value="Cham - Cham"/>
<enumeration value="Cher - Cherokee"/>
<enumeration value="Cirt - Cirth"/>
<enumeration value="Copt - Coptic"/>
<enumeration value="Cprt - Cypriot"/>
<enumeration value="Cyrl - Cyrillic"/>
<enumeration value="Cyrs - Cyrillic (Old Church Slavonic
variant)"/>
<enumeration value="Deva - Devanagari (Nagari)"/>
<enumeration value="Dsrt - Deseret (Mormon)"/>
<enumeration value="Dupl - Duployan shorthand, Duployan
stenography"/>
<enumeration value="Egyd - Egyptian demotic"/>
<enumeration value="Egyh - Egyptian hieratic"/>
<enumeration value="Egyp - Egyptian hieroglyphs"/>
<enumeration value="Elba - Elbasan"/>
<enumeration value="Ethi - Ethiopic"/>
<enumeration value="Geok - Khutsuri (Asomtavruli and
Nuskhuri)"/>
<enumeration value="Geor - Georgian (Mkhedruli)"/>
<enumeration value="Glag - Glagolitic"/>
<enumeration value="Goth - Gothic"/>
<enumeration value="Gran - Grantha"/>
<enumeration value="Grek - Greek"/>
<enumeration value="Gujr - Gujarati"/>
<enumeration value="Guru - Gurmukhi"/>
<enumeration value="Hanb - Han with Bopomofo"/>
<enumeration value="Hang - Hangul"/>
<enumeration value="Hani - Han (Hanzi, Kanji, Hanja)"/>
<enumeration value="Hano - Hanunoo (Hanunóo)"/>
<enumeration value="Hans - Han (Simplified variant)"/>
<enumeration value="Hant - Han (Traditional variant)"/>
<enumeration value="Hatr - Hatran"/>
<enumeration value="Hebr - Hebrew"/>
<enumeration value="Hira - Hiragana"/>
<enumeration value="Hluw - Anatolian Hieroglyphs"/>
<enumeration value="Hmng - Pahawh Hmong"/>
<enumeration value="Hrkt - Japanese syllabaries"/>
<enumeration value="Hung - Old Hungarian (Hungarian
Runic)"/>
<enumeration value="Inds - Indus (Harappan)"/>
<enumeration value="Ital - Old Italic (Etruscan, Oscan
etc.)"/>
<enumeration value="Jamo - Jamo"/>
<enumeration value="Java - Javanese"/>
<enumeration value="Jpan - Japanese"/>
<enumeration value="Jurc - Jurchen"/>
<enumeration value="Kali - Kayah Li"/>
<enumeration value="Kana - Katakana"/>
<enumeration value="Khar - Kharoshthi"/>
<enumeration value="Khmr - Khmer"/>
<enumeration value="Khoj - Khojki"/>

```

```

<enumeration value="Kitl - Khitan large script"/>
<enumeration value="Kits - Khitan small script"/>
<enumeration value="Knda - Kannada"/>
<enumeration value="Kore - Korean (alias for Hangul +
Han)"/>

<enumeration value="Kpel - Kpelle"/>
<enumeration value="Kthi - Kaithi"/>
<enumeration value="Lana - Tai Tham (Lanna)"/>
<enumeration value="Laoo - Lao"/>
<enumeration value="Latf - Latin (Fraktur variant)"/>
<enumeration value="Latg - Latin (Gaelic variant)"/>
<enumeration value="Latn - Latin"/>
<enumeration value="Leke - Leke"/>
<enumeration value="Lepc - Lepcha (Róng)"/>
<enumeration value="Limb - Limbu"/>
<enumeration value="Lina - Linear A"/>
<enumeration value="Linb - Linear B"/>
<enumeration value="Lisu - Lisu (Fraser)"/>
<enumeration value="Loma - Loma"/>
<enumeration value="Lyci - Lycian"/>
<enumeration value="Lydi - Lydian"/>
<enumeration value="Mahj - Mahajani"/>
<enumeration value="Mand - Mandaic, Mandaean"/>
<enumeration value="Mani - Manichaean"/>
<enumeration value="Marc - Marchen"/>
<enumeration value="Maya - Mayan hieroglyphs"/>
<enumeration value="Mend - Mende Kikakui"/>
<enumeration value="Merc - Meroitic Cursive"/>
<enumeration value="Mero - Meroitic Hieroglyphs"/>
<enumeration value="Mlym - Malayalam"/>
<enumeration value="Modi - Modi, Modī"/>
<enumeration value="Mong - Mongolian"/>
<enumeration value="Moon - Moon (Moon code, Moon script,
Moon type)"/>

<enumeration value="Mroo - Mro, Mru"/>
<enumeration value="Mtei - Meitei Mayek (Meithei, Meetei)"/>
<enumeration value="Mult - Multani"/>
<enumeration value="Mymr - Myanmar (Burmese)"/>
<enumeration value="Narb - Old North Arabian (Ancient North
Arabian)"/>

<enumeration value="Nbat - Nabataean"/>
<enumeration value="Newa - Newa, Newar, Newari"/>
<enumeration value="Nkgb - Nakhi Geba"/>
<enumeration value="Nkoo - N'Ko"/>
<enumeration value="Nshu - Nüshu"/>
<enumeration value="Ogam - Ogham"/>
<enumeration value="Olck - OL Chiki (OL Cemet', OL,
Santali)"/>

<enumeration value="Orkh - Old Turkic, Orkhon Runic"/>
<enumeration value="Orya - Oriya"/>
<enumeration value="Osge - Osage"/>
<enumeration value="Osma - Osmanya"/>
<enumeration value="PalM - Palmyrene"/>
<enumeration value="Pauc - Pau Cin Hau"/>
<enumeration value="Perm - Old Permic"/>
<enumeration value="Phag - Phags-pa"/>
<enumeration value="Phli - Inscriptional Pahlavi"/>
<enumeration value="Phlp - Psalter Pahlavi"/>
<enumeration value="Phlv - Book Pahlavi"/>

```



```

<enumeration value="Phnx - Phoenician"/>
<enumeration value="Piqd - Klingon (KLI pIqaD)"/>
<enumeration value="PLrd - Miao (Pollard)"/>
<enumeration value="Prti - Inscriptional Parthian"/>
<enumeration value="Rjng - Rejang (Redjang, Kaganga)"/>
<enumeration value="Roro - Rongorongo"/>
<enumeration value="Runr - Runic"/>
<enumeration value="Samr - Samaritan"/>
<enumeration value="Sara - Sarati"/>
<enumeration value="Sarb - Old South Arabian"/>
<enumeration value="Saur - Saurashtra"/>
<enumeration value="Sgnw - SignWriting"/>
<enumeration value="Shaw - Shavian (Shaw)"/>
<enumeration value="Shrd - Sharada, Śārādā"/>
<enumeration value="Sidd - Siddham"/>
<enumeration value="Sind - Khudawadi, Sindhi"/>
<enumeration value="Sinh - Sinhala"/>
<enumeration value="Sora - Sora Sompeng"/>
<enumeration value="Sund - Sundanese"/>
<enumeration value="Sylo - Syloti Nagri"/>
<enumeration value="Sycr - Syriac"/>
<enumeration value="Syre - Syriac (Estrangelo variant)"/>
<enumeration value="Syrj - Syriac (Western variant)"/>
<enumeration value="Syrn - Syriac (Eastern variant)"/>
<enumeration value="Tagb - Tagbanwa"/>
<enumeration value="Tacr - Takri"/>
<enumeration value="Tale - Tai Le"/>
<enumeration value="Talu - New Tai Lue"/>
<enumeration value="Taml - Tamil"/>
<enumeration value="Tang - Tangut"/>
<enumeration value="Tavt - Tai Viet"/>
<enumeration value="Telu - Telugu"/>
<enumeration value="Teng - Tengwar"/>
<enumeration value="Tfng - Tifinagh (Berber)"/>
<enumeration value="Tglg - Tagalog (Baybayin, Alibata)"/>
<enumeration value="Thaa - Thaana"/>
<enumeration value="Thai - Thai"/>
<enumeration value="Tibt - Tibetan"/>
<enumeration value="Tirh - Tirhuta"/>
<enumeration value="Ugar - Ugaritic"/>
<enumeration value="Vaii - Vai"/>
<enumeration value="Visp - Visible Speech"/>
<enumeration value="Wara - Warang Citi (Varang Kshiti)"/>
<enumeration value="Wole - Woleai"/>
<enumeration value="Xpeo - Old Persian"/>
<enumeration value="Xsux - Cuneiform, Sumero-Akkadian"/>
<enumeration value="Yiii - Yi"/>
<enumeration value="Zinh - Code for inherited script"/>
<enumeration value="Zmth - Mathematical notation"/>
<enumeration value="Zsye - Symbols (Emoji variant)"/>
<enumeration value="Zsym - Symbols"/>
<enumeration value="Zxxx - Code for unwritten documents"/>
<enumeration value="Zyyy - Code for undetermined script"/>
<enumeration value="Zzzz - Code for uncoded script"/>
<enumeration value="other"/>
</restriction>
</simpleType>
<simpleType name="ColourDepthSimpleType">
  <restriction base="string">

```



```

        <enumeration value="bilevel"/>
        <enumeration value="greyscale"/>
        <enumeration value="colour"/>
        <enumeration value="other"/>
    </restriction>
</simpleType>
<simpleType name="GraphicsTypeSimpleType">
    <restriction base="string">
        <enumeration value="Logo"/>
        <enumeration value="Letterhead"/>
        <enumeration value="decoration"/>
        <enumeration value="frame"/>
        <enumeration value="handwritten-annotation"/>
        <enumeration value="stamp"/>
        <enumeration value="signature"/>
        <enumeration value="barcode"/>
        <enumeration value="paper-grow"/>
        <enumeration value="punch-hole"/>
        <enumeration value="other"/>
    </restriction>
</simpleType>
<simpleType name="ChartTypeSimpleType">
    <restriction base="string">
        <enumeration value="bar"/>
        <enumeration value="line"/>
        <enumeration value="pie"/>
        <enumeration value="scatter"/>
        <enumeration value="surface"/>
        <enumeration value="other"/>
    </restriction>
</simpleType>
<complexType name="LayersType">
    <annotation>
        <documentation>
            Can be used to express the z-index of overlapping
            regions. An element with a greater z-index is always in
            front of another element with lower z-index.
        </documentation>
    </annotation>
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="Layer" type="pc:LayerType"/></element>
    </sequence>
</complexType>

<complexType name="LayerType">
    <sequence minOccurs="1" maxOccurs="unbounded">
        <element name="RegionRef" type="pc:RegionRefType"/>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
    <attribute name="zIndex" type="int" use="required"/>
    <attribute name="caption" type="string"/>
</complexType>

<complexType name="BaselineType">
    <attribute name="points" type="pc:PointsType"
        use="required"/>
    </attribute>
    <attribute name="conf" type="pc:ConfSimpleType">
        <annotation>

```

```

1) </documentation>
    </annotation>
  </attribute>
</complexType>

<simpleType name="PointsType">
  <annotation>
    <documentation>
      Point list with format "x1,y1 x2,y2 ...", where
      "x" / "y" refer to the horizontal / vertical
      pixel positions in a coordinate system which always
      references the root PageType/@imageFilename, with
      "0,0" in the upper left corner of the root image and
      "imageWidth,imageHeight" in the lower right.</documentation>
    </annotation>
    <restriction base="string">
      <pattern value="([0-9]+,[0-9]+ )+([0-9]+,[0-9]+)"></pattern>
    </restriction>
  </simpleType>

<complexType name="RelationsType">
  <annotation>
    <documentation>
      Container for one-to-one relations between layout
      objects (for example: DropCap - paragraph, caption -
      image).
    </documentation>
  </annotation>
  <sequence minOccurs="1" maxOccurs="unbounded">
    <element name="Relation" type="pc:RelationType"/>
  </sequence>
</complexType>

<complexType name="RelationType">
  <annotation>
    <documentation>
      One-to-one relation between to layout object. Use 'link'
      for loose relations and 'join' for strong relations
      (where something is fragmented for instance).

      Examples for 'link': caption - image floating -
      paragraph paragraph - paragraph (when a paragraph is
      split across columns and the last word of the first
      paragraph DOES NOT continue in the second paragraph)
      drop-cap - paragraph (when the drop-cap is a whole word)

      Examples for 'join': word - word (separated word at the
      end of a line) drop-cap - paragraph (when the drop-cap
      is not a whole word) paragraph - paragraph (when a
      paragraph is split across columns and the last word of
      the first paragraph DOES continue in the second
      paragraph)
    </documentation>
  </annotation>
  <sequence>
    <element name="Labels" type="pc:LabelsType"
      minOccurs="0" maxOccurs="unbounded">
      <annotation>

```

```

                                <documentation>Semantic labels /
tags</documentation>
                                </annotation>
                                </element>
                                <element name="SourceRegionRef" type="pc:RegionRefType"
                                    minOccurs="1" maxOccurs="1">
                                </element>
                                <element name="TargetRegionRef" type="pc:RegionRefType"
                                    minOccurs="1" maxOccurs="1">
                                </element>
                            </sequence>
                            <attribute name="id" type="ID" use="required"/>
                            <attribute name="type">
                                <simpleType>
                                    <restriction base="string">
                                        <enumeration value="link"/>
                                        <enumeration value="join"/>
                                    </restriction>
                                </simpleType>
                            </attribute>
                            <attribute name="custom" type="string">
                                <annotation>
                                    <documentation>For generic use</documentation>
                                </annotation>
                            </attribute>
                            <attribute name="comments" type="string"/>
                        </complexType>

                        <simpleType name="ProductionSimpleType">
                            <annotation>
                                <documentation>Text production type</documentation>
                            </annotation>
                            <restriction base="string">
                                <enumeration value="printed"/>
                                <enumeration value="typewritten"/>
                                <enumeration value="handwritten-cursive"/>
                                <enumeration value="handwritten-printschrift"/>
                                <enumeration value="medieval-manuscript"/>
                                <enumeration value="other"/>
                            </restriction>
                        </simpleType>

                        <complexType name="TextStyleType">
                            <annotation>
                                <documentation>
                                    Monospace (fixed-pitch, non-proportional) or
                                    proportional font.
                                </documentation>
                            </annotation>
                            <attribute name="fontFamily" type="string">
                                <annotation>
                                    <documentation>
                                        For instance: Arial, Times New Roman.
                                        Add more information if necessary
                                        (e.g. blackletter, antiqua).
                                    </documentation>
                                </annotation>
                            </attribute>
                            <attribute name="serif" type="boolean">

```

```

        <annotation>
            <documentation>
                Serif or sans-serif typeface.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="monospace" type="boolean"/>
    <attribute name="fontSize" type="float">
        <annotation>
            <documentation>
                The size of the characters in points.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="xHeight" type="integer">
        <annotation>
            <documentation>
                The x-height or corpus size refers to the distance
                between the baseline and the mean line of
                lower-case letters in a typeface.
                The unit is assumed to be pixels.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="kerning" type="int">
        <annotation>
            <documentation>
                The degree of space (in points) between
                the characters in a string of text.
            </documentation>
        </annotation>
    </attribute>
    <attribute name="textColour" type="pc:ColourSimpleType"/>
    <attribute name="textColourRgb" type="integer">
        <annotation>
            <documentation>
                Text colour in RGB encoded format
                (red value) + (256 x green value) + (65536 x blue
value).
            </documentation>
        </annotation>
    </attribute>
    <attribute name="bgColour" type="pc:ColourSimpleType">
        <annotation>
            <documentation>Background colour</documentation>
        </annotation>
    </attribute>
    <attribute name="bgColourRgb" type="integer">
        <annotation>
            <documentation>
                Background colour in RGB encoded format
                (red value) + (256 x green value) + (65536 x blue
value).
            </documentation>
        </annotation>
    </attribute>
    <attribute name="reverseVideo" type="boolean">
        <annotation>
            <documentation>

```

```

        Specifies whether the colour of the text appears
        reversed against a background colour.
        </documentation>
    </annotation>
</attribute>
<attribute name="bold" type="boolean"/>
<attribute name="italic" type="boolean"/>
<attribute name="underlined" type="boolean"/>
<attribute name="underlineStyle"
    type="pc:UnderlineStyleSimpleType" use="optional">
    <documentation>Line style details if "underlined" is
TRUE
        </documentation>
    </annotation>
</attribute>
<attribute name="subscript" type="boolean"/>
<attribute name="superscript" type="boolean"/>
<attribute name="strikethrough" type="boolean"/>
<attribute name="smallCaps" type="boolean"/>
<attribute name="letterSpaced" type="boolean"/>
</complexType>

<complexType name="RegionType" abstract="true">
    <sequence>
        <element name="AlternativeImage"
type="pc:AlternativeImageType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>
                    Alternative region images
                    (e.g. black-and-white).
                </documentation>
            </annotation>
        </element>
        <element name="Coords" type="pc:CoordsType"/>
        <element name="UserDefined" type="pc:UserDefinedType"
            minOccurs="0" maxOccurs="1">
        </element>
        <element name="Labels" type="pc:LabelsType"
            minOccurs="0" maxOccurs="unbounded">
            <annotation>
                <documentation>Semantic labels /
tags</documentation>
            </annotation>
        </element>
        <element name="Roles" type="pc:RolesType"
            minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation>
                    Roles the region takes
                    (e.g. in context of a parent region).
                </documentation>
            </annotation>
        </element>
        <choice minOccurs="0" maxOccurs="unbounded">
            <element name="TextRegion" type="pc:TextRegionType"/>
            <element name="ImageRegion"
type="pc:ImageRegionType"/>

```

```

        <element name="LineDrawingRegion"
type="pc:LineDrawingRegionType"/>
        <element name="GraphicRegion"
type="pc:GraphicRegionType"/>
        <element name="TableRegion"
type="pc:TableRegionType"/>
        <element name="ChartRegion"
type="pc:ChartRegionType"/>
        <element name="SeparatorRegion"
type="pc:SeparatorRegionType"/>
        <element name="MathsRegion"
type="pc:MathsRegionType"/>
        <element name="ChemRegion" type="pc:ChemRegionType"/>
        <element name="MusicRegion"
type="pc:MusicRegionType"/>
        <element name="AdvertRegion"
type="pc:AdvertRegionType"/>
        <element name="NoiseRegion"
type="pc:NoiseRegionType"/>
        <element name="UnknownRegion"
type="pc:UnknownRegionType"/>
        <element name="CustomRegion"
type="pc:CustomRegionType"/>
    </choice>
</sequence>
<attribute name="id" type="ID" use="required"/>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string"/>
<attribute name="continuation" type="boolean">
    <annotation>
        <documentation>
            Is this region a continuation of another region
            (in previous column or page, for example)?
        </documentation>
    </annotation>
</attribute>
</complexType>

<complexType name="AlternativeImageType">
    <attribute name="filename" type="string" use="required"/>
    <attribute name="comments" type="string"/>
    <attribute name="conf" type="pc:ConfSimpleType">
        <annotation>
            <documentation>Confidence value (between 0 and
1)</documentation>
        </annotation>
    </attribute>
</complexType>

<simpleType name="AlignSimpleType">
    <restriction base="string">
        <enumeration value="left"/>
        <enumeration value="centre"/>
        <enumeration value="right"/>
        <enumeration value="justify"/>
    </restriction>
</simpleType>

```

```

        </restriction>
    </simpleType>
    <simpleType name="GroupTypeSimpleType">
        <restriction base="string">
            <enumeration value="paragraph"/>
            <enumeration value="list"/>
            <enumeration value="list-item"/>
            <enumeration value="figure"/>
            <enumeration value="article"/>
            <enumeration value="div"/>
            <enumeration value="other"/>
        </restriction>
    </simpleType>
    <simpleType name="TextDataTypeSimpleType">
        <restriction base="string">
            <enumeration value="xsd:decimal">
                <annotation>
                    <documentation>
                        Examples:
                        "123.456", "+1234.456",
                        "-1234.456", "-.456", "-456"
                    </documentation>
                </annotation>
            </enumeration>
            <enumeration value="xsd:float">
                <annotation>
                    <documentation>
                        Examples:
                        "123.456", "+1234.456", "-1.2344e56",
                        "-.45E-6", "INF", "-INF", "NaN"
                    </documentation>
                </annotation>
            </enumeration>
            <enumeration value="xsd:integer">
                <annotation>
                    <documentation>
                        Examples:
                        "123456", "+00000012", "-1", "-456"
                    </documentation>
                </annotation>
            </enumeration>
            <enumeration value="xsd:boolean">
                <annotation>
                    <documentation>
                        Examples: "true", "false", "1", "0"
                    </documentation>
                </annotation>
            </enumeration>
            <enumeration value="xsd:date">
                <annotation>
                    <documentation>
                        Examples:
                        "2001-10-26", "2001-10-26+02:00",
                        "2001-10-26Z", "2001-10-26+00:00",
                        "-2001-10-26", "-20000-04-01"
                    </documentation>
                </annotation>
            </enumeration>
            <enumeration value="xsd:time">

```

```

        <annotation>
            <documentation>
                Examples:
                "21:32:52", "21:32:52+02:00", "19:32:52Z",
                "19:32:52+00:00", "21:32:52.12679"
            </documentation>
        </annotation>
    </enumeration>
    <enumeration value="xsd:dateTime">
        <annotation>
            <documentation>
                Examples:
                "2001-10-26T21:32:52", "2001-10-
26T21:32:52+02:00",
                "2001-10-26T19:32:52Z", "2001-10-
26T19:32:52+00:00",
                "-2001-10-26T21:32:52", "2001-10-
26T21:32:52.12679"
            </documentation>
        </annotation>
    </enumeration>
    <enumeration value="xsd:string">
        <annotation>
            <documentation>Generic text
string</documentation>
        </annotation>
    </enumeration>
    <enumeration value="other">
        <annotation>
            <documentation>
                An XSD type that is not listed or a custom type
                (use dataTypeDetails attribute).
            </documentation>
        </annotation>
    </enumeration>
</restriction>
</simpleType>
<complexType name="GraphemesType">
    <annotation>
        <documentation>
            Container for graphemes, grapheme groups and
            non-printing characters.
        </documentation>
    </annotation>
    <choice minOccurs="1" maxOccurs="unbounded">
        <element name="Grapheme" type="pc:GraphemeType"/>
        <element name="NonPrintingChar"
type="pc:NonPrintingCharType"/>
        <element name="GraphemeGroup" type="pc:GraphemeGroupType"/>
    </choice>
</complexType>
<complexType name="GraphemeBaseType" abstract="true">
    <annotation>
        <documentation>
            Base type for graphemes, grapheme groups and non-printing
characters.
        </documentation>
    </annotation>
</sequence>

```



```

        <element name="TextEquiv" type="pc:TextEquivType"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="id" type="ID" use="required"/>
    <attribute name="index" use="required">
        <annotation>
            <documentation>
                Order index of grapheme, group, or non-printing
                character
                within the parent container (graphemes or glyph or
                grapheme group).
            </documentation>
        </annotation>
    </simpleType>
    <restriction base="int">
        <minInclusive value="0"/></minInclusive>
    </restriction>
</simpleType>
</attribute>
<attribute name="Ligature" type="boolean"/>
<attribute name="charType">
    <annotation>
        <documentation>
            Type of character represented by the
            grapheme, group, or non-printing character element.
        </documentation>
    </annotation>
    <simpleType>
        <restriction base="string">
            <enumeration value="base"/>
            <enumeration value="combining"/>
        </restriction>
    </simpleType>
</attribute>
<attribute name="custom" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
<attribute name="comments" type="string">
    <annotation>
        <documentation>For generic use</documentation>
    </annotation>
</attribute>
</complexType>
<complexType name="GraphemeType">
    <annotation>
        <documentation>
            Represents a sub-element of a glyph.
            Smallest graphical unit that can be
            assigned a Unicode code point.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="pc:GraphemeBaseType">
            <sequence>
                <element name="Coords" type="pc:CoordsType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </complexContent>
    </complexType>
    <complexType name="NonPrintingCharType">
        <annotation>
            <documentation>
                A glyph component without visual representation
                but with Unicode code point.
                Non-visual / non-printing / control character.
                Part of grapheme container (of glyph) or grapheme sub
group.
            </documentation>
        </annotation>
        <complexContent>
            <extension base="pc:GraphemeBaseType">
            </extension>
        </complexContent>
    </complexType>
    <complexType name="GraphemeGroupType">
        <complexContent>
            <extension base="pc:GraphemeBaseType">
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="Grapheme"
type="pc:GraphemeType"/>
                    <element name="NonPrintingChar"
type="pc:NonPrintingCharType"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>

    <complexType name="UserDefinedType">
        <annotation>
            <documentation>Container for user-defined
attributes</documentation>
        </annotation>
        <sequence>
            <element name="UserAttribute" type="pc:UserAttributeType"
minOccurs="1" maxOccurs="unbounded">
            </element>
        </sequence>
    </complexType>
    <complexType name="UserAttributeType">
        <annotation>
            <documentation>Structured custom data defined by name, type
and value.</documentation>
        </annotation>
        <attribute name="name" type="string"/>
        <attribute name="description" type="string"/>
        <attribute name="type">
            <simpleType>
                <restriction base="string">
                    <enumeration value="xsd:string"/>
                    <enumeration value="xsd:integer"/>
                    <enumeration value="xsd:boolean"/>
                    <enumeration value="xsd:float"/>
                </restriction>
            </simpleType>
        </attribute>
        <attribute name="value" type="string"/>

```

```

</complexType>

<complexType name="TableCellRoleType">
  <attribute name="rowIndex" type="int" use="required">
    <annotation>
      <documentation>Cell position in table starting with
row 0</documentation>
    </annotation>
  </attribute>
  <attribute name="columnIndex" type="int" use="required">
    <annotation>
      <documentation>Cell position in table starting with
column 0</documentation>
    </annotation>
  </attribute>
  <attribute name="rowSpan" type="int">
    <annotation>
      <documentation>Number of rows the cell spans
(optional; default is 1)</documentation>
    </annotation>
  </attribute>
  <attribute name="colSpan" type="int">
    <annotation>
      <documentation>Number of columns the cell spans
(optional; default is 1)</documentation>
    </annotation>
  </attribute>
  <attribute name="header" type="boolean">
    <annotation>
      <documentation>
        Is the cell a column or row header?
      </documentation>
    </annotation>
  </attribute>
</complexType>
<complexType name="RolesType">
  <sequence>
    <element name="TableCellRole" type="pc:TableCellRoleType"
      minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation>
          Data for a region that takes on the role
          of a table cell within a parent table region.
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<simpleType name="UnderLineStyleSimpleType">
  <restriction base="string">
    <enumeration value="singleLine" />
    <enumeration value="doubleLine" />
    <enumeration value="other" />
  </restriction>
</simpleType>
</schema>

```