WandaML
a markup language for digital document annotation

Katrin Franke[2], Isabelle Guyon[1], Lambert Schomaker[3], and Louis Vuurpijl[4]

1. ClopiNet, 955 Creston Rd, Berkeley, USA, isabelle@clopinet.com (corresponding author.)

2. Fraunhofer Institute, Berlin, Germany.

3. Rijksuniversiteit Groningen, The Netherlands.

4. University of Nijmegen, The Netherlands.

Abstract

WandaML is an XML-based markup language for the annotation and filter journaling of digital documents. It addresses in particular the needs of forensic handwriting data examination, by allowing experts to enter information about writer, material (pen, paper), script and content, and to record chains of image filtering and feature extraction operations applied to the data. We present the design of this format and some annotation examples, in the more general perspective of digital document annotation. Annotations may be organized in a structure that reflects the document layout via a hierarchy of document regions. WandaML can lend itself to a variety of applications, including the annotation all kinds of handwriting documents (on-line or off-line), images of printed text, medical images, and satellite images.
Keywords: Handwriting, forensic data, XML, annotations, data format, document analysis.

1 Introduction

We present the design of an XML-based markup language to annotate digital documents, called WandaML . This markup language is designed for processing, analyzing and storing handwriting samples in application to forensic handwriting examination and writer identification. In the context of this application, particular specifications were met to ensure objectivity and reproducibility of the processing steps and examination results [6, 5]. Writer identification can never be as accurate as iris or DNA-based identification. However, usually a lot of constraining pieces of information are known (age category, handedness, major script style), which may reduce the size of a reference set to such an extent that automatic writer identification on the basis of script shape within that reduced set becomes viable. To that end, a portable and extensible data format is needed for modelling the knowledge from the forensic application domain. A standard database technology can then be used to apply logical constraints to the search process.

Although our work is very application oriented, it is not particularly application specific. Our format lends itself to promoting research and development of handwriting analysis methods, and establishing common grounds for international exchange of handwritten data samples and annotations. It supports on-line, off-line handwriting and overlays of on-line over off-line data. Its simple and general structure makes it fit for annotating data with other modalities, including printed text, pictures, and sound.

To ensure long-term operation we built upon the e*X*tensible *M*arkup *L*anguage (XML) [4] that follows world-wide standardized syntax rules, recommended by the *W*orld *W*ide *W*eb *C*onsortium (W3C). XML allows users to define their own markup language respecting the basic XML syntax. As an ASCII format, XML is human readable and can be edited with regular text processors, although there is a large body of existing software, which manipulates XML. In particular, XML and Java are very complementary: XML is a widely accepted means of creating portable data, and the Java programming language provides portable code abilities. We wrote Java applications for WandaML as part of the Wanda project [6, 5].

Before undertaking the task of defining a new XML language for forensic applications, we reviewed existing standards that are in use in the handwriting recognition and document analysis community. We distinguish between formats to *encode* data and formats to *annotate* data. WandaML is essentially a data annotation format. It is compatible with a variety of data encoding formats. For image data, typical examples of data encoding formats would be JPEG, GIF, and TIFF. For vectorial data (graphics represented by their coordinate points, not by pixel values) we can cite the Scalable-Vector Graphics (SVG) format.

Several handwritten document annotation formats predate WandaML . Image annotation formats include IAM [9], Xmillum [11] and Trueviz [8]. Formats for virtual ink or "electronic ink" combining both data *encoding* and data *annotation*, include Unipen [7] and InkML [3]. We summarize the features of the existing formats in Table 1. We outline in bold the features that are required for forensic applications. No standard predating WandaML meets all of our requirements.

WandaML is the result of joint efforts of forensic handwriting experts and researchers in computer-based handwriting analysis. Well-established termini and procedures of forensic science inspired many aspects of the design [2, 10, 12]. WandaML also benefitted from the experience of some of the team members who invented Unipen, a data format for on-line handwriting [7].

2 Notations and conventions

A particular XML markup language such as WandaML can be defined in standard ways using the document type definition language (DTD) or XML schemas [4]. We provide DTDs to specify WandaML [1].

Following the general XML nomenclature, in XML annotation we have:

```
<my_tag my_attribute="my_value">
    <my_element/>
</my_tag>
```

	InkXML	Unipen	SVG	IAM	XMillum	TrueViz	WANDA
XML-based	√		√	√	√	√	√
raster images			√	√	√	√	√
vector images	√	√	√				√
virtual ink	√	√					√
v-ink segments	√	√					
image regions				√	√	√	√
ink/image overlay	√						√
device annot.	√	√					√
writer annot.		√		√			√
script annot.		√					√
material annot.							√
content annot.		√		√			√
filters/plugins			√		√		√
interactivity			√		√		
layers					√	√	
styles			√		√		
external link		√	√				√

Table 1: **Existing format comparison.**

where, `<my_tag/>` and `<my_element/>` are tags[1] or "elements", and `my_attribute` is an attribute of `my_tag`, having value `my_value`. In DTDs, "entities" are defined, which may be used as lists of admissible attribute values.

We adopt a minimum set of conventions carried throughout WANDAML :

- Entities, attributes and elements contain only lowercase and underscore characters.
- Certain attributes have special meanings: `id` a unique identifier, `type` a type from a pre-defined list, `label` an optional user-defined string that can be used for search purpose.
- To simplify parsing, we have defined a number of "containers" (`<pages/>`, `<filters/>`, `<annotations/>`, `<inputs/>`, and `<outputs/>`), which contain elements of the same name (`<page/>`, `<filter/>`, `<annotation/>`, `<input/>`, and `<output/>`) and have the optional attribute `number_of`.

A defined markup language based on XML is extensible via the use of name spaces. A body of XML text may be enclosed between tags with an opening tag containing the attribute `xmlns` (XML name space.) The value of the attribute `xmlns` is a unique name reserved to identify the definition of a particular XML subset. The URI (Uniform Resource Identifier) of a DTD file or a URL (Uniform Resource Locators) are frequently used for that purpose. The use of name spaces in WANDAML allows us to separate the definition of the basic language skeleton from XML subsets that are application specific.

[1]We often use the shorthand `<my_tag/>` for empty tags `<my_tag></my_tag>` or tags whose content is not expanded.

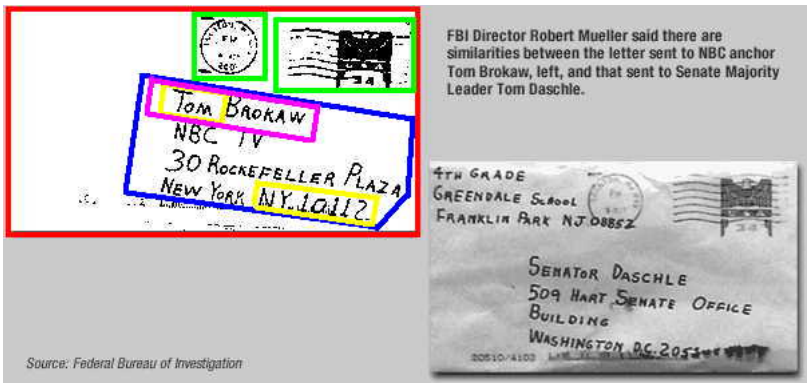

Figure 1: **Wanda document annotation using regions .** These envelopes from actual suspects in the recent anthrax criminal case provide us with an example of possible use of our WANDA regions: A region of interest is delineated, isolating one envelope. A filter is applied to the region to clean the data. A hierarchy of regions (e.g. in address block, lines, words, and characters) is defined and annotated.

3 A simple scenario

To understand the WANDAML annotation mechanisms and concepts, it is useful to first understand what it was originally intended for by analyzing an example. The concepts introduced in this Section will be later defined formally in Section **??**. In Figure 1, we show an example of forensic evidence. The tasks of a computer-assisted human expert analyzing this document may include:

1. outlining regions of interest,
2. enhancing image quality,
3. measuring handwriting characteristics (size of characters, slant, etc.),
4. translating handwriting into typed text,
5. retracing characters with electronic ink,
6. annotating defined regions with information relative to content (threat, envelope, form, etc.), writer (age, gender, etc.), script (handprinted, cursive, etc.), and material (paper, pen, etc.)

A document annotation file always start with a root element `<wandoc/>`:

```
<wandoc
    id="20032004"
    label="Anthrax example case"
    xmlns="http://clopinet.com/isabelle/Projects/WANDA/wandoc/DTD-HOME.html"
/>
```

The `id` is a machine generated unique document annotation identifier, while the `label` is provided by the user. The name space `xmlns` points to the `wandoc` language definition.

In our example (Figure 1), the `<wandoc/>` element contains a single page:

```
<wandoc ...>
    <pages numberOf="1">
        <page id="20032004_copy5" label="frontpage" next=""/>
    </pages>
</wandoc>
```

By convention, we replace tag attributes by "..." to shorten the description. The page contains one call to a filter, two annotations, and one region:

```
<page ...>
     <filters number_of="1"/>
     <annotations number_of="2"/>
     <regions number_of="1"/>
</page>
```

In the next paragraph, we expand the `<filters ...>` tag. In this example, a filter is importing an image from a scanner using the a scan software called "IBIS" and returning a link to the resulting image. In the `wandoc` framework, a document consists of one or several pages, each of which may be represented by an image. Notice that there is no tag `<image/>`. Images are imported through especially defined filters, e.g. the `scan` filter.

```
<filters number_of="1">
    <filter type="import" label="ibisScan">
        <inputs>
            <input type="stream" number="1" xmlns="../scan.dtd">
                <scan/>
            </input>
        </inputs>
        <module type="extern" exec="ibis.exe">
            <meta version="3.51" />
        </module>
        <outputs>
            <output type="file">
                <wanda\_link href="copy5.tif" />
            </output>
        </outputs>
    </filter>
</filters>
```

region	- attributes: id, label, next - delineated by a set of points (rectangle or polygon) - content: filters, annotations, other regions, wanda link, meta

Table 2: **Wanda region summary.**

A set of annotations are entered by an expert (e.g. with a GUI):

```
<annotations number_of="2">
    <annotation type="content" xmlns="../content.dtd">
        <whole_document type="envelope" intent="personal"/>
    </annotation>
    <annotation type="writer" xmlns="../writer.dtd">
        <writer id="2015">
            <properties handedness="left" skill="ok"/>
        </writer>
    </annotation>
</annotations>
```

The region content is the following:

```
<regions number_of="1">
    <region id="20032004_0001" label="Letter to Tom Brokaw" next="2">
        <points>
            <point x="0" y="0" />
            <point x="0" y="124" />
            <point x="76" y="124" />
            <point x="76" y="124" />
        </points>
        <annotations number_of="3"/>
        <filters number_of="1"/>
    </region>
</regions>
```

A region is defined by a unique id (presumably machine generated). Ids allow users and programs (filters) to refer to regions. Otherwise, by default, filters apply to their parent region, page or document. Additionally, regions may possess a user-defined label, the intend of which is to facilitate searching through regions. The attribute "next" is used to indicate a logical ordering of the regions, which are at the same hierarchical level. Such ordering is used, for instance, to indicate reading order. Regions are delineated by a polygon defined by a set of points. The origin is at the upper left corner. The unit, if not specified, is the pixel. The region of our example possesses three annotations and one filter. The filter corresponds to some measurements and returns features (not shown).

4 Basic concepts: regions, annotations, and filters

The example of the previous section introduced a number of concepts that are essential to WandaML : regions, annotations, and filters that we explain in more details in this section. WandaML follows a general skeleton:[2]

```
<wandoc>
    <filters/> [?]
    <annotations/> [?]
    <wanda_link/> [*]
    <meta/> [?]
    <pages> [?]
        <page> [+]
            <filters/> [?]
            <annotations/> [?]
            <regions/> [?]
            <wanda_link/> [*]
            <meta/> [?]
        </page>
    </pages>
</wandoc>
```

There is a thin line between a markup language and a programming language. We make use of such possibilities of XML by introducing "filters", which allow users to record and eventually play back operations on the data. A filter can be understood as a computer program that processes the document and returns either a new transformed image or a set of features. We adopt the following skeleton for filters:

```
<filter> [+]
    <inputs> [+]
        <input/> [+]
    </inputs>
    <module/> [+]
    <outputs> [+]
        <output/> [+]
    </outputs>
</filter>
```

The tags `<input/>` and `<output/>` wrap around application specific tags defining inputs and outputs. Via the use of name spaces, new types of inputs and outputs can be defined to extend WandaML , without changing the core of the language. We include in the documentation examples of application specific filters [1].

Filters are general enough to encode any kind of computer-assisted document processing, including the definition of regions and annotations. Still, we introduce specialized tags for

[2]We use the following convention to describe tag requirements: [+] at least one; [?] zero or one; [*] any number including zero.

writer	- person (first and last name, gender, year of birth) - properties (handedness, skill) - education (country, level) - language (native)
material	- paper (type, size, material, weight, product, absorbency) - pen (type, product, tip features, ink features) - pad (type, surface, hardness)
script	- type (e.g. Latin, Greek, Arabic, etc.) - language (language in which the script is written) - style (major style, connection, capitalization, consistency, stroke quality embellishment, connectivity, speed)
content	- document information (type, intent) - text block attributes (type, length) - text block content analysis (tone, grammar, spelling, verbatim transcription)

Table 3: **Wanda annotation categories.** We summarize in this table the attributes collected in the various annotation categories.

the "region" and "annotation" concepts because such concepts are central in the applications envisioned. The use of the specialized tags `<region/>` and `<annotation/>` enhances legibility and facilitates computer parsing.

Wanda regions provide the possibility of multiple levels of annotations and local filtering. Regions may either be manually extracted or be the result of an automatic document segmentation. We summarize in Table 2 the essential region properties.

The `<annotation/>` tag is generic, it is merely a wrapper tag specifying a name space. This name space references a set of application specific tags, e.g. `<writer/>`, `<script/>`, `<content/>`, or `<material/>`. We summarize the content of these annotations in Table 3.

We introduce a `<meta/>` tag that regroups information about how the particular annotation was generated (author, date created, contact email, author's affiliation, etc.) and pertains to all WandaML subsets. We also introduce a generic `<wanda_link/>` tag: it refers to a file locator and plays a role similar to an HTML hyperlink or an Xlink.[3]

Finally, WandaML allows experts to superimpose virtual ink (electronic ink) as entered, for instance, using a digitizing tablet, on top of a document image. This is achieved with the `wink` XML subset, which we do not describe here for space constraint reasons. See our techreport for details [1].

Overall, the `wandoc` DTD contains only 21 elements (or tags). But we have defined 10 specialized languages for our application, totalling 146 elements [1].

[3]The definition of a `wanda_link` in terms of Xlink has not been firmed up yet but there are provisions for it in the DTD.

5 Application to forensic handwriting examination

WandaML might become an important information exchange vehicle in daily forensic casework and forensic analysis research. After being generated by expert rating and/or by technical equipment, WandaML data may be used to generate reports documenting handwriting characteristics and be transferred to a database for large-scale writer identification.

In its current version WandaML elaborates on the Fish [10] system, and introduces further categories and admissible values. The annotation categories implemented in WandaML (Table 3) are well established in forensic science. Other categories such as the one supplied for digitalization, cleaning and interactive measurements are of great importance for computer-based examination. In the Wanda system, we implemented standard measurements such as slant, character and word spacing, ascender and descender length [13]. Using the open concepts of `<filters/>` and `<annotations/>` WandaML is also capable to fulfill upcoming demands.

The representation of handwritings features with a standardized XML format supports the exchange of examination result between different laboratories and/or governmental entities. The data can be rapidly imported into any computer platform. So, WandaML promotes interoperability, and, with the anchors for further extensions, long-term usability.

6 Conclusion

This paper described a new data format WandaML to annotate forensic handwriting data. The choice of XML makes it intrinsically extensible. Our design is centered around a small number of concepts and conventions. Regions, annotations, and filters are the three essential elements of all WandaML annotated documents. Regions are parts of documents delineated with a rectangular of polygonal shape. Regions are naturally organized by the XML hierarchy. This reflects well the hierarchical nature of documents' organization (e.g., page, paragraph, line, word, character). In addition, within a given hierarchical level, regions are organized in linked lists to encode logical relations such as reading order. Region annotations are encapsulated within a generic annotation tag. Via the mechanism of XML name spaces, this provides users with the flexibility of defining their own types of annotations to supplement the types already defined: writer, material, script, and content. WandaML has a simple syntax to format filters. New types of filter inputs and outputs may be defined via the use of name spaces. WandaML also defines some application specific filter formats, and a virtual ink encoding format. We foresee that the simplicity and extensibility of the framework will encourage its wide spreading within the handwriting recognition community and in the document analysis community at large.

Acknowledgements

This work was sponsored by the Bundeskriminalamt in Wiesbaden, Germany (BKA). We gratefully acknowledge the many persons who contributed, and particularly: Axel Kerkhoff, Werner Kuckuck, Gerhard Grube, Altug Metin, Tomas Kühn, Martin Penk, Steffen Rose, Johan Everts, Geertje Zwarts, Merijn van Erp, Bernhard Boser, and Stefan Giesler.

References

[1] A data standard for the annotation and storage of handwriting samples in the context of (computer-based) forensic handwriting analysis and writer identification. Technical report, http://www.unipen.org/wandaML/, 2003.

[2] de Jong, et al. Computer aided analysis of handwriting, the NIFO-TNO approach. In *4th European Handwriting Conference for Police and Gov. Handwriting Experts*, 1994.

[3] Ink Markup Language W3C Working Draft. http://www.w3.org/tr/inkml/.

[4] XML eXtensible Markup Language. http://www.w3.org/xml/.

[5] Franke, et al. Wanda: A common ground for forensic handwriting examination and writer identification. *ENFHEX news - Bulletin of the European Network of Forensic Handwriting Experts*, (1/04, ISSN-1456-1469).

[6] K. Franke, et al. Wanda: A generic framework applied in forensic handwriting analysis and writer identification. In A. Abraham, et al, editor, *Proc. 3rd International Conference on Hybrid Intelligent Systems*, pages 927–938, Amsterdam, 2003.

[7] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN project of on-line data exchange and benchmarks. In *Proceedings of the 12th International Conference on Pattern Recognition*. IAPR-IEEE, http://www.unipen.org, 1994.

[8] T. Kanungo, C. H. Lee, J. Czorapinski, and I. Bella. Trueviz: a groundtruth/metadata editing and visualizing toolkit for ocr. In *SPIE Conference on Document Recognition and Retrieval*. http://www.cfar.umd.edu/∼kanungo/software/trueviz-$1_0$2.tar.gz, 2001.

[9] U. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. In *5th Int. Conf. on Document Analysis and Recognition, ICDAR'99*, pages 705–708, Bangalore, India, 1999. http://www.iam.unibe.ch/ zimmerma/iamdb/iamdb.html.

[10] M. Philipp. Expected future developments in the Forensic Information System Handwriting. In *4th Euro. Conf. for Police and Gov. Handwg. Experts*, London, UK, 1994.

[11] Roussel, et al. Web-based cooperative document understanding. In *6th IEEE Int. Conf. on Doc. Ana. and Rec.*, pages 368–373. http://xmillum.sourceforge.net/, 2001.

[12] L. R. B. Schomaker and L. G. Vuurpijl. Forensic writer identification: A benchmark data set and a comparison of two systems. Technical report, Nijmegen Institute for Cognition and Information, University of Nijmegen, The Netherlands, 2000.

[13] M. van Erp, L.G. Vuurpijl, K. Franke, and L. R. B. Schomaker. The wanda measurement tool for forensic document examination. In *11th Conf. of the Int. Graphonomics Sty.*, pages 282–285, Scottsdale, Arizona, USA, 2003. http://pentel.ipk.fhg.de.