# A Framework for the Standardized Description of Handwritten Annotations

Marcel Götze

Department for Simulation and Graphics
University of Magdeburg
Germany

DC2005, September 2005, Madrid

# Agenda

# Introduction

- handwritten annotations well known from paper documents
- readers write short notes on the margin or mark important parts of the document
- this support active reading and personalizes the document (Adler and van Doren, 1972)
- applications already support freehand annotation of electronic documents (Price, Schilit, Golovchinsky, 1998)
- “the intelligent pen” an application for the automatic recognition of annotations
- “magic pages” for the personalized visualization of annotations
- application for the annotation of webpages

# Annotation of Webpages

- annotation of webpages mostly based on typed annotations
- e.g. this supports:
  - CSCW (Cadiz et al., 2000; Davis and Huttenlocher, 1995)
  - organization and classification (Denoue and Vignollet, 2001)
  - semantic web
- but not reader's natural reading behavior
- webpages interesting documents for handwritten annotation
  - information can easy be exchanged
  - many digital documents are webpages
  - handwritten annotations add semantic to (parts of) webpages

# Why use Metadata?

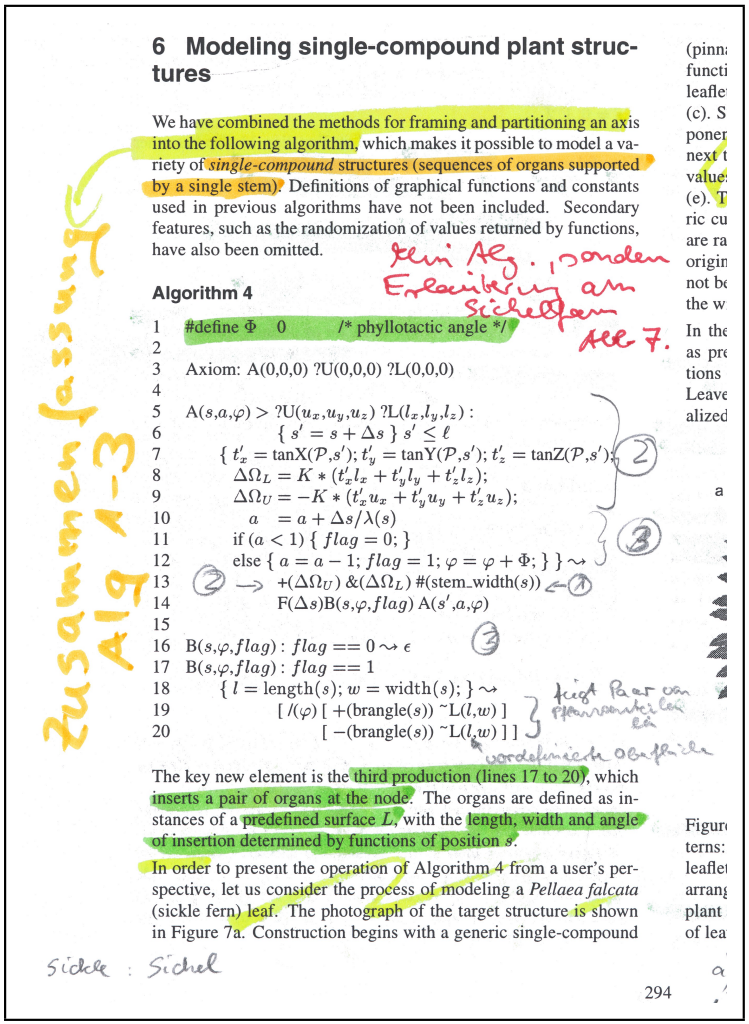
- if you annotate a digital document for yourself – no metadata is needed
- metadata is important if more than one person annotates a document
  - review scenario
  - sharing of knowledge about a document (e.g. notes about further reading, importance of parts of the document)
  - discussion about a document (e.g. ranking)

# From Typed Annotations to Handwritten Annotations

- starting point for research: typed annotations
- many applications exist that deal with typed and other annotations
- e.g. “multivalent documents” (Phelps, 1997), “Notable” (Baldonado et al., 2001)
- the only one that uses standardized metadata (DC): Annotea (Kahan et al., 2001)
- information about annotations and the annotation itself
  - metadata and annotation description

# Annotations in Paperdocuments

- underlining
- notes
- margin bars
- surroundings
- frames
- textmarker
- numbers
- special signs
- images



6 Modeling single-compound plant structures

We have combined the methods for framing and partitioning an axis into the following algorithm, which makes it possible to model a variety of *single-compound* structures (sequences of organs supported by a single stem). Definitions of graphical functions and constants used in previous algorithms have not been included. Secondary features, such as the randomization of values returned by functions, have also been omitted.

Algorithm 4

```
1  #define Φ  0          /* phyllotactic angle */
2
3  Axiom: A(0,0,0) ?U(0,0,0) ?L(0,0,0)
4
5  A(s,a,φ) > ?U(u_x,u_y,u_z) ?L(l_x,l_y,l_z) :
6          { s' = s + Δs } s' ≤ ℓ
7      { t'_x = tanX(P,s'); t'_y = tanY(P,s'); t'_z = tanZ(P,s');
8        ΔΩ_L = K * (t'_x l_x + t'_y l_y + t'_z l_z);
9        ΔΩ_U = −K * (t'_x u_x + t'_y u_y + t'_z u_z);
10         a   = a + Δs/λ(s)
11       if (a < 1) { flag = 0; }
12       else { a = a − 1; flag = 1; φ = φ + Φ; } } ⇝
13 ⟶ +(ΔΩ_U) &(ΔΩ_L) #(stem_width(s))
14       F(Δs)B(s,φ,flag) A(s',a,φ)
15
16 B(s,φ,flag) : flag == 0 ⇝ ε
17 B(s,φ,flag) : flag == 1
18     { l = length(s); w = width(s); } ⇝
19           [ /(φ) [ +(brangle(s)) ~L(l,w) ]
20                  [ −(brangle(s)) ~L(l,w) ] ]
```

The key new element is the third production (lines 17 to 20), which inserts a pair of organs at the node. The organs are defined as instances of a predefined surface $L$, with the length, width and angle of insertion determined by functions of position $s$.

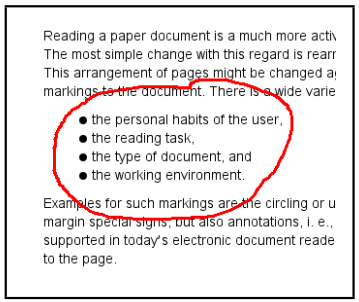
In order to present the operation of Algorithm 4 from a user's perspective, let us consider the process of modeling a *Pellaea falcata* (sickle fern) leaf. The photograph of the target structure is shown in Figure 7a. Construction begins with a generic single-compound

294

# Automatic recognition

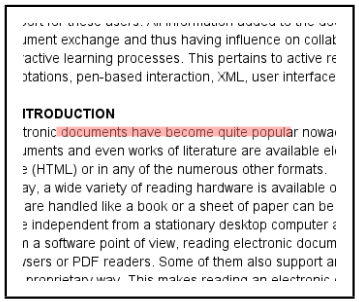
- before automatic recognition:
  - handwritten annotations are pen movements (path of the pen)
- after automatic recognition
  - the shape is known
  - the handwritten annotation is been replaced by the recognized
    - gives a visual feedback
    - saves space (on the harddisk and on the document)
  - geometry changed from one polygonal shape to many geometrical representations
  - Scalable Vector Graphics (SVG) allows the description of all annotation types

# SVG for Handwritten Annotations




surrounding

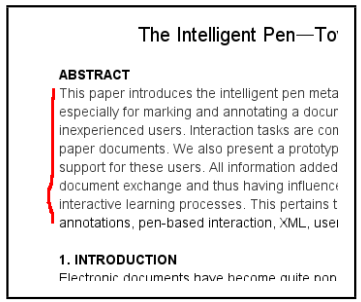
<ellipse cx cy rx ry fill stroke stroke-width>




textmarker

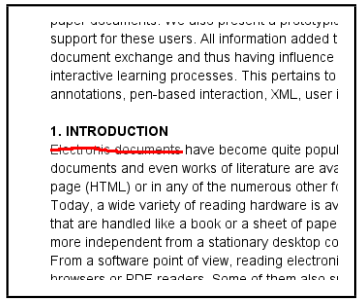
<line x1 y1 x2 y2 stroke stroke-width opacity>

<rectangle x y width height fill stroke opacity>




margin bar

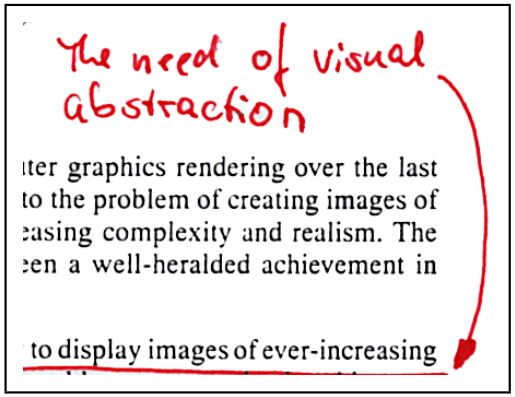
<line x1 y1 x2 y2 stroke stroke-width>




underline

<line x1 y1 x2 y2 stroke stroke-width>

# SVG for handwritten Annotations

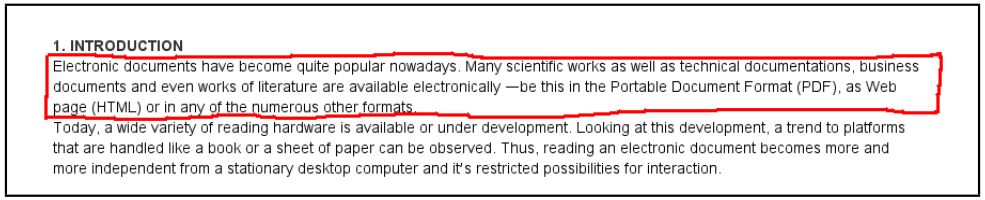



note

<path d cx cy rx ry stroke stroke-width>

images

<image x y width height xlink:href="myimage.gif">




rectangular frame

<rect x y width height rx ry fill stroke stroke-width>

# Dublin Core for Handwritten Annotations

- difference between typed and handwritten annotations
- Annotea defines classes such as: “advice”, “comment”, “example”, etc. interactively.
- handwritten annotations define classes inherently using pencolor, pentype
- this causes a difference in the use of DC

# Metadata for Handwritten Annotations

- *type of annotation:* e.g. underlining, surrounding, margin bar, notes
- *type of the pen:* e.g. ballpen, pencil, textmarker
- *color:* of the pen.
- *author:* different authors use the same pentype and color
- *date and time:* to save the chronological order of annotations
- *part of the text:* scope of the annotation within the text
- *reference to the document:* annotation can be stored apart from the original document (URL, ISBN, etc.)

# The use of Dublin Core Elements

- restriction to the original 15 elements
- no refinements

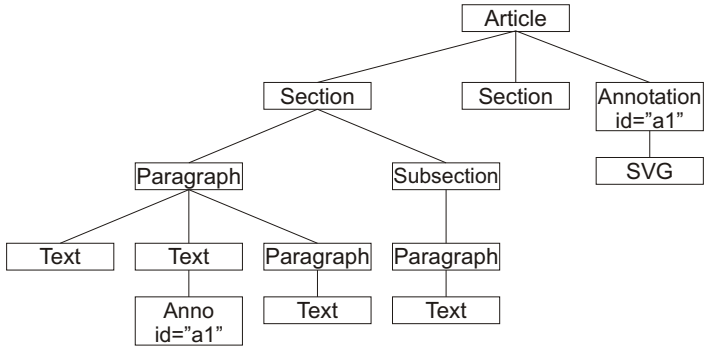
| | |
|---|---|
| author | given name, surname |
| date | date and time of the creation |
| relation | reference to the digital document |
| coverage | scope of the annotation within the text |
| type | type of the annotation |
| format | used pentype and color |

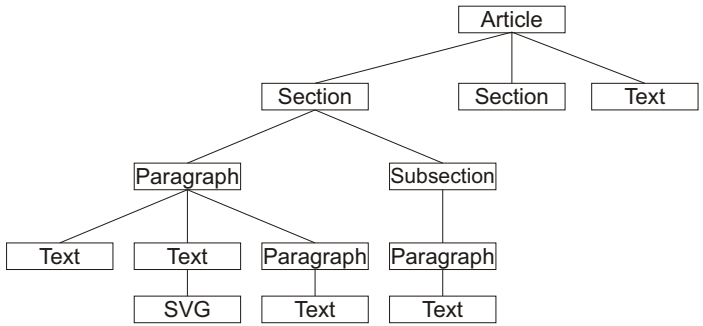
# Creation of the Metadata

- metadata will be automatically created
  - every information about the annotation can be automatically collected by the system
  - based on the interaction with the document: coverage, relation, date and time
  - based on the interaction with the system: format of the pen, type of the annotation
  - based on the login name: author

# SVG and DOM

- the common internal document format for webpages is the document object model (DOM)
- XML based tree structure
- SVG fits perfectly into this tree structure
- SVG allows the attachment of a metadata element (RDF) for the description of a SVG element
- annotations can be grouped into on SVG node



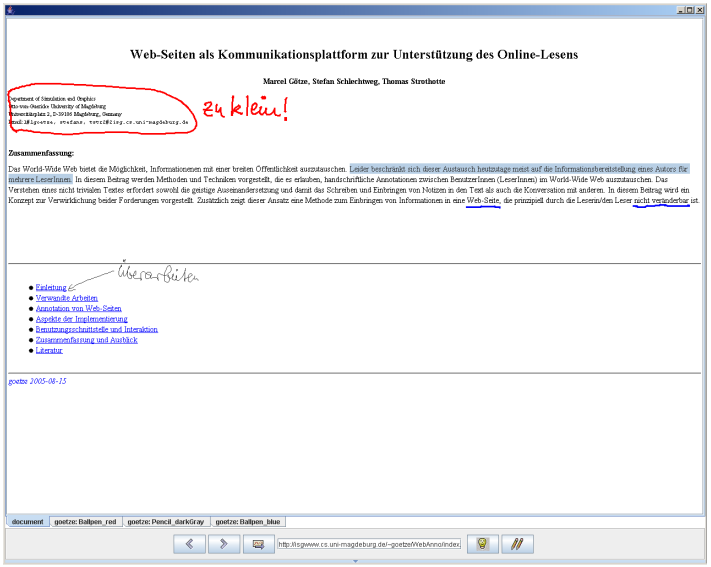
# Conclusion

- standardized description of handwritten annotations
- supports description of annotation itself and
- use of metadata (e.g. author, pentype or pencolor) specifically for handwritten annotations
- use for webpages and other digital documents
- supports:
  - the automatic processing of annotations (e.g. search, execution)
  - the separate storage of annotations

# Future Work - WebAnno

Handwritten Annotations on Webpages

- problem: webpages can not be edited
- annotations can only be stored separately from the document
- Client-Server Architecture
- user interface based on pen and paper metaphor
- annotation data and metadata will be analyzed and categorized



# Future Work

- combination of handwritten and typed annotations
- automatic recognition of handwritten notes
- description of correction marks
- description of the semantic of the annotation
  - every annotation is meaningful
  - automatic recognition is not possible (except correction marks)
  - annotations itself are metadata
- semantic web

Thank you for your attention.

Questions?, Comments?, Ideas?

Marcel Götze
goetze@isg.cs.uni-magdeburg.de