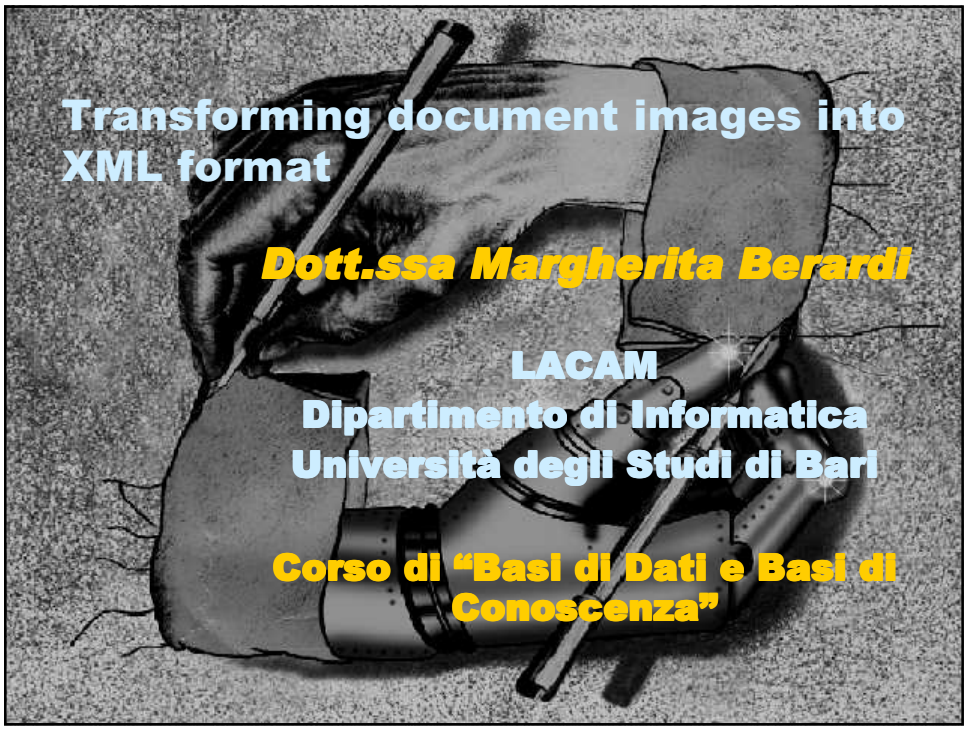


Overview

- The problem of information capture from paper documents
- Document processing steps
- Machine learning techniques for block classification
- Machine learning techniques for document classification and understanding
- Transformation into XML format
- Conclusions

The data acquisition problem

- U.S. National Library of Medicine, Bethesda, Maryland
- Automating the production of bibliographic records for MEDLINE, a database of references in medical journals.
- 12 millions of citations drawn 4,600 journals
- 40,000 records a month
- Creating online bibliographic databases from paper-based journal articles continues to be heavily manual.

The data acquisition problem

- Numerous valuable historic and cultural sources are imperiled and scattered in various national archives
- Full knowledge and usage of this material are severely impeded by access problems due to:
  - difficult-to-use or electronically unavailable sources, i.e. both documents and formal reference systems
  - the lack of appropriate content-based search and retrieval aids that would help users find what they really need.

The data communication problem

- Web-accessible format: HTML, XML, ...
- Why not document images?
  - Slow
  - not editable
  - sequential structure (no hypertext)
- Information retrieval of XML documents is easier
  - XML-QL is a one of the query languages used to express database-style queries in XML documents.
- Commercial OCR systems are still far away from performing satisfactorily the conversion into XML format.

The representation of extracted information: a key issue

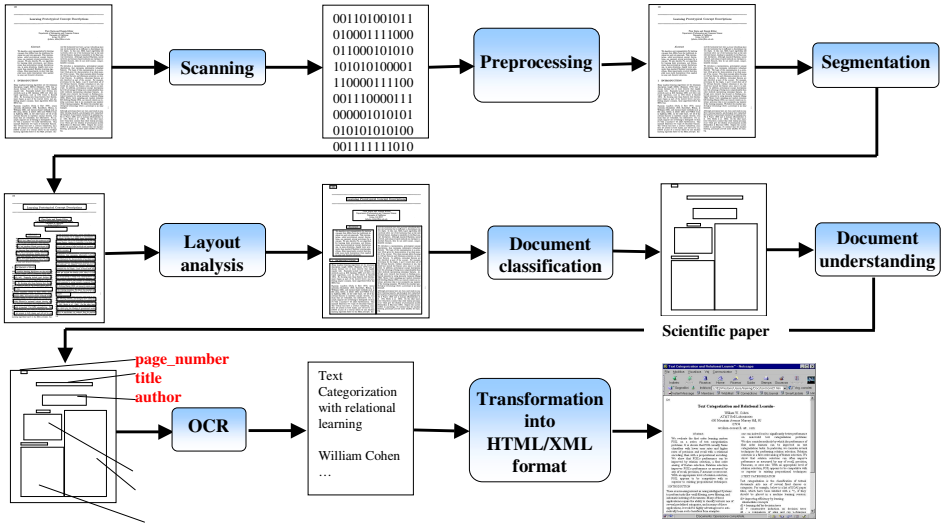
- A suitable representation for semi-structured documents
- Web accessibility and easier retrieval
- Extensibility and generality
- Need to:
  - define hypertext structures
  - store additional information on the semantics of text
  - represent the logical structure of documents

XML is a natural choice

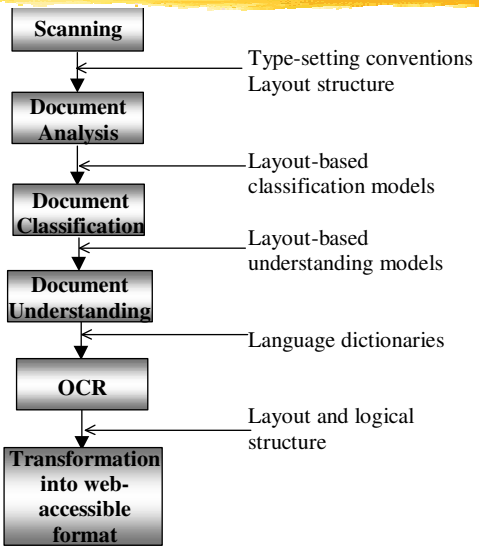
Transforming paper documents into HTML/XML format: a simple task?

- The presentation on the browser is not similar in appearance to the original document (different layout or geometrical structure).
- Rendering problems, such as missing graphical components, wrong reading ordering in two-columned papers, missing indentation, …
- No style sheet is associated to documents saved in HTML format, so that the presentation of textual information cannot be customized for viewing.
- The HTML language cannot represent the logical document structure (title, author, abstract, …)

Document processing steps



Document processing steps: required knowledge.



Acquiring required knowledge: a machine learning approach

- Problem: Knowledge acquisition for "intelligent" document processing systems.
- Solution: Machine learning algorithms, in particular symbolic inductive learning techniques
- Justification: Symbolic learning techniques generate human-comprehensible hypotheses of the underlying patterns (*comprehensibility postulate*).

Step	Technique	Related Issues
Document analysis	Induction of decision trees	▪ Incrementality ▪ Space efficiency
Document classification and understanding	Induction of first-order rules	▪ Repr. language ▪ Dependencies among concepts

WISDOM++

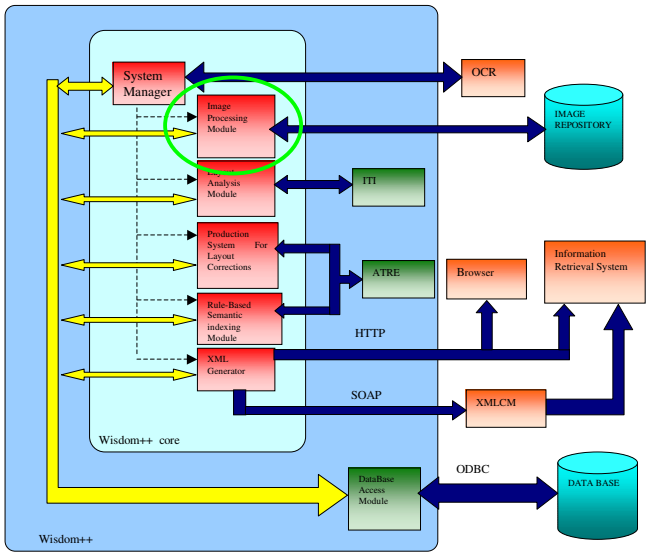
- Document analysis system
  - Document analysis
  - Document classification
  - Document understanding
  - Text recognition with an OCR
  - Transformation of the document into HTML/XML format
- Distinguishing features
  - Adaptivity ⇐ machine learning tools & techniques
  - Interactivity ⇐ glass-box model

www.di.uniba.it/~malerba/wisdom++/

Acquiring required knowledge

Step	Knowledge technologies
classification of basic-blocks	the decision tree learning system ITI
layout analysis	a production-system which operates with a forward-chaining control structure
automatic global layout analysis correction	the inductive logic programming system ATRE
semantic indexing (document image classification and understanding)	a knowledge-based system which contains explicitly represented rules and supports inference by means of theorem proving mechanisms

System architecture (1)



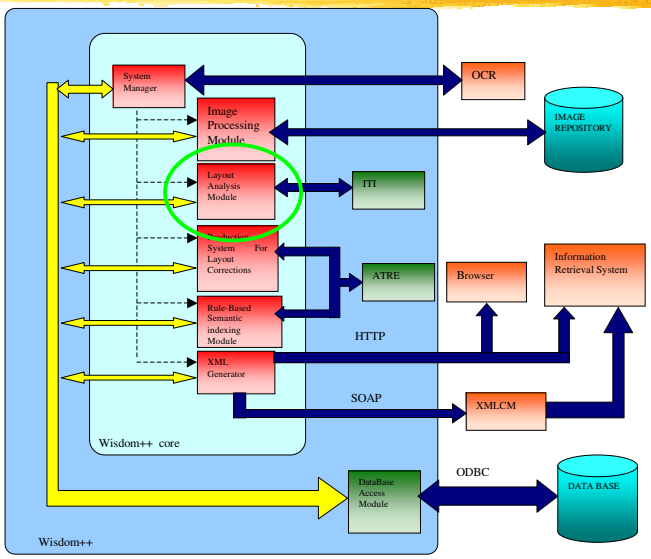
Pre-processing

Input: page image (TIFF format, 300 dpi, ~1 Mb)

Problem	Solution
Evaluation of the skew angle	Alignment measure based on the horizontal projection profile
Rotation	Rotation based on the skew angle
Computation of the spread factor	Ratio of the mean distance between peaks and peak width

Output: pre-processed page image

System architecture (2)



Segmentation

Input: pre-processed page image

Problem

Identification of rectangular blocks enclosing content portions

Solution

Variant of the *Run Length Smoothing Algorithm* where:

- the image is scanned only twice (instead of 4 times) with no additional cost
- the smoothing parameters are defined on the ground of the spread factor

Output: segmented page image

Block classification

Input: segmented page image (unclassified blocks)

Problem	Solution
Labeling blocks according to the type of content ▮ text block ▮ horizontal line ▮ vertical line ▮ picture ▮ graphics	Decision tree classifier

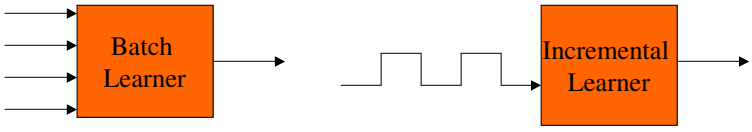
Output: segmented page image (classified blocks)

Learning decision trees for blocks classification

- The classifier is a decision tree automatically built from a set of training examples (blocks) of the five classes.
- Two approaches to decision-tree induction:

Batch learning: examples are considered all together to build the decision tree

Incremental: the current decision tree is revised in response to each newly training example presented to the system

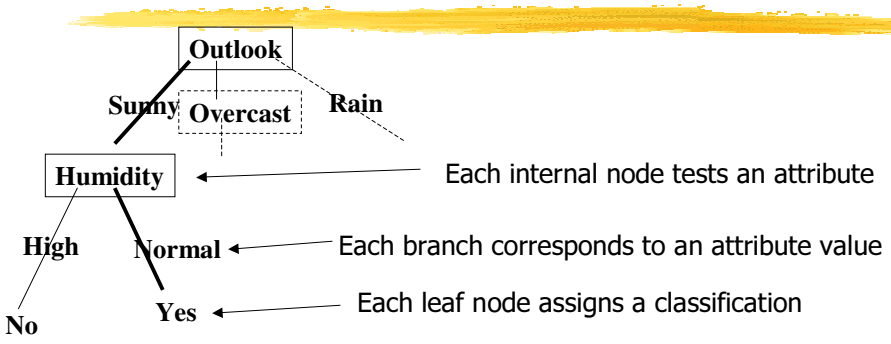


ITI (Utgoff, 1994) is the only incremental decision tree learning system that handles numerical data.

Block classification

- Features to describe blocks:
  - ✓ height,length and area of the image block;
  - ✓ *eccentricity*: *length/height*;
  - ✓ total number of black pixels in the image block;
  - ✓ total number of black-white transitions in all rows of the image block;
  - ✓ percentage of black pixels per area;
  - ✓ *mean_tr*: *blackpix/bw_trans*;
  - ✓ *F1*: short run emphasis;
  - ✓ *F2*: long run emphasis;
  - ✓ *F3*: extra long run emphasis.
- Labels of blocks (result of the classification):
  - ✓ text block
  - ✓ horizontal line
  - ✓ vertical line
  - ✓ picture
  - ✓ graphics

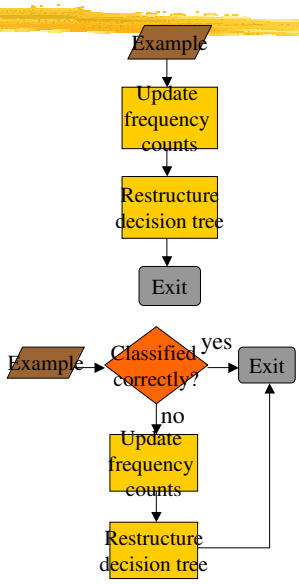
Decision trees



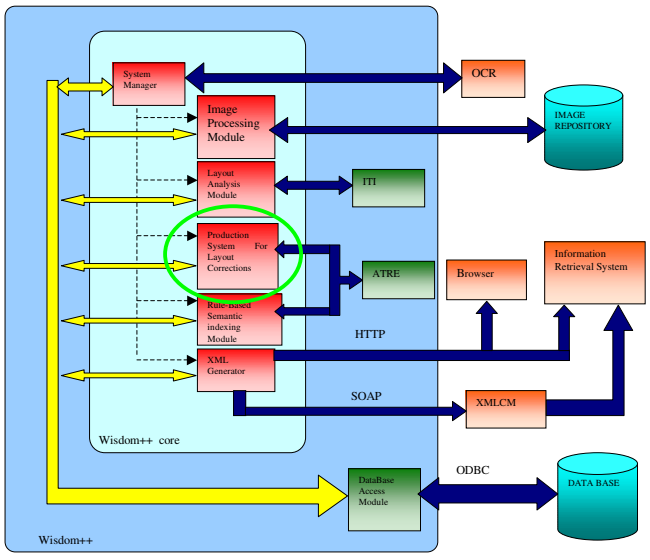
- Each path (root→leaf) is a conjunction of attribute tests, e.g.,
  - Outlook=Sunny AND Humidity=Normal
- The whole tree is the disjunction of these conjunctions.
  - (Outlook=Sunny AND Humidity=Normal) OR (…)….

Normal vs. Error-correction mode

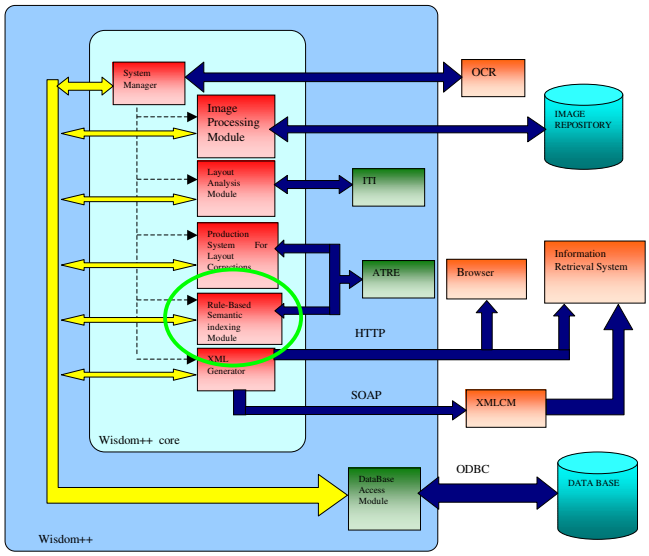
- ITI can operate in three different ways
  - Batch
  - Incremental
    - Normal: both examples misclassified and examples correctly classified are used to update the tree.
    - Error-correction: only examples misclassified are used to update the tree
- Normal operation mode returns trees equal to the batch mode (presentation order invariance)
- Error-correction mode is affected by the order in which examples are presented.



System architecture (3)



System architecture (4)



Document Classification & Understanding

Input: segmented page image (layout components)

The application of machine learning techniques to a *layout-based classification and understanding* requires a suitable representation of:

- the layout structure of the training documents
- the rules induced from the training documents

Requirements

- Capturing spatial relationships between layout components
- Efficient handling of numerical descriptors

Output: segmented page image (logical components)

Document Classification & Understanding: The representation problem

Zero-order representation

- Language primitives: attributes
- Expressive power: properties of a single layout component

vs

First-order representation

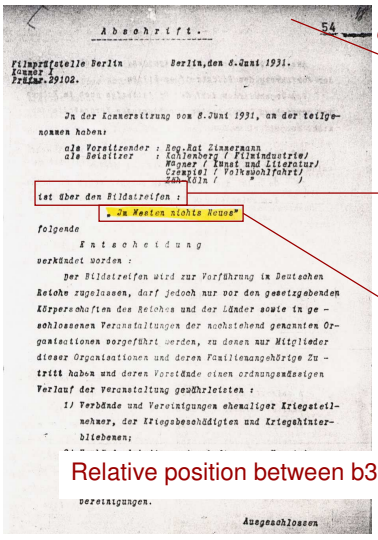
- Language primitives: attributes & relations
- Expressive power: properties of a single layout component & spatial relationships between logical components

Numeric/symbolic representation

System: ATRE (Malerba, 1997)

- Discretization of numerical attributes
- Recursion and concepts dependencies

Examples of document descriptions in FOL



Relative position between b3 and b4

d1

class_dif_censorship_decision(d1) :-

…

b3

part_of(d1,b3),
width(b3,103),
height(b3,55),
type_of_text(b3),
x_pos_center(b3,115),
y_pos_center(b3,323),

b4

part_of(d1,b4),
width(b4,87),
height(b4,57),
type_of_text(b4),
x_pos_center(b4, 297),
y_pos_center(b4, 340),

on_top(b3,b4),
to_right(b3,b4),
…

Document Understanding
Dependencies among logical components

Learning rules for document understanding is
more difficult than
learning rules for document classification

Why?

Logical components refer to a part of the document rather than to the whole document and may be related each other

logic_type(X) = body ← to_right(Y,X), logic_type(Y) = abstract

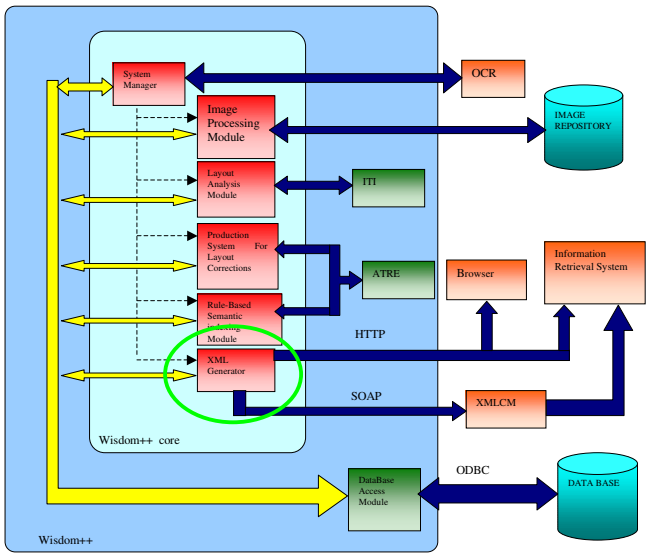
How to handle dependencies?

ATRE learns *multiple dependent concepts* starting from a set of training documents labeled by the expert user.

Which impact on experimental results?

Experimental results confirm that by taking into account concept dependencies it is possible to improve the predictive accuracy of the document understanding rules.

System architecture (5)



Generation of the XML document: the *D*ocument *T*ype *D*efinition (DTD)

```
<!-- standard DTD file for tpami class -->
<!ELEMENT tpami (logic-structure?, geometric-structure)>
<!ELEMENT logic-structure (title | author | abstract)*>
<!ELEMENT title (paragraph)*>
<!ATTLIST title      ID NMTOKEN #IMPLIED>
<!ELEMENT author (paragraph)*>
<!ATTLIST author ID NMTOKEN #IMPLIED>
<!ELEMENT abstract (paragraph)*>
<!ATTLIST abstract ID NMTOKEN #IMPLIED>
<!ELEMENT paragraph (#PCDATA | TAB)*>
<!ELEMENT TAB EMPTY>
<!ELEMENT geometric-structure (image, blocklevels)>
<!ELEMENT image EMPTY>
<!ATTLIST image... >
<!ELEMENT blocklevels (basic-block, line, setofline, frame1, frame2)>
.....
```

Generation of the XML document: the XML file (e*X*tensible *M*arkup *L*anguage)

```
<?xml-stylesheet href="icml16.XSL" type="text/xsl"?>
<!DOCTYPE icml SYSTEM "icml.DTD">
<icml>
<page-number><paragraph>108</paragraph></page-number>
<title><paragraph>K*: An Instance-based Learner Using an
  Entropic Distance Measure</paragraph>
<paragraph></paragraph></title>
<author ID="id4"><paragraph>John G. Cleary</paragraph>
<paragraph>Dept. of Computer Science</paragraph>
<paragraph>University of Waikato</paragraph>
<paragraph>New Zealand</paragraph>
<paragraph>jcleary@waikato.ac.nz</paragraph>
...
```

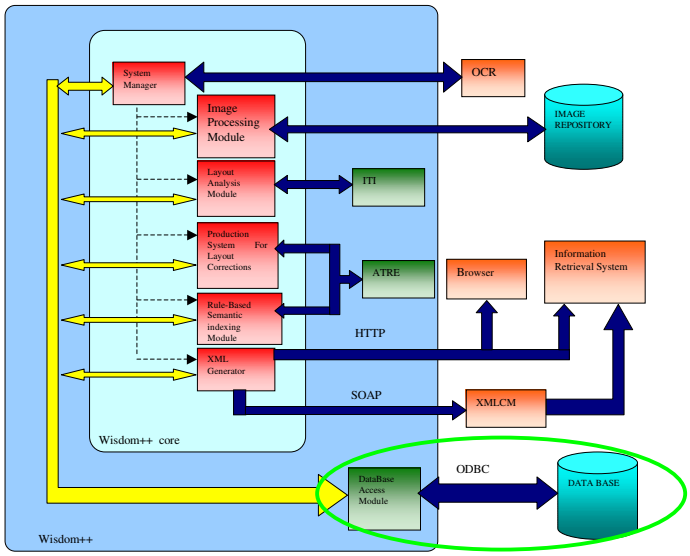
Generation of the XML document: the *XSL* file (e*X*tensible *S*tyle *L*anguage)

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl='http://www.w3.org/TR/WD-xsl'>
<xsl:template match='/'>
<HTML>
<HEAD>
<TITLE>K*: An Instance-based Learner Using an Entropic
  Distance Measure </TITLE>
<LINK rel="stylesheet" href="icml.css"></LINK>
</HEAD>
<BODY TEXT="BLACK" BGCOLOR="WHITE">
<TABLE WIDTH='100%' BORDER='0'>
<TR>
<TD WIDTH='99%'></TD>
<TD WIDTH='0%' VALIGN='TOP'><BR/>
<IMG SRC="icml16j11.jpg"/></TD>
<TD WIDTH='1%'></TD>
</TR> ...
```

Generation of the XML document: the *CSS* file (*C*ascading *S*tyle *S*heets)

```
TD {font: 7pt Times New Roman;text-align: justify;}
TD.title {font-size: 14pt;  font-weight: bold; text-
  align: center;}
TD.author {font-size: 12pt; text-align: center;}
TD.abstract {font-size:11pt;}
TD.body {font-size: 12pt;}
TD.page-number {font-size: 10pt;}
BR {font-size: 3pt;}
```

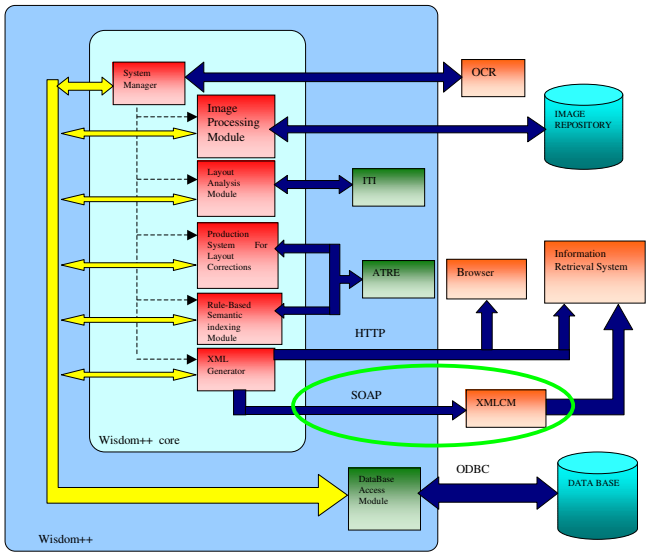
System architecture (6)



The database component

- Functional omogeneity with both ORACLE DB and MS-Access
  - Design of schemas for all information
- ODBC interface
- Intensive use of database
  - Rules, documents, users, layout, text, …
- Number of accesses to the database optimized
  - Layout of the whole multipage document loaded in main memory
  - store procedures for layout loading
  - indexes definition on external keys

System architecture (7)



The SOAP (Simple Object Access Protocol) client

- SOAP provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using XML
- SOAP defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems ranging from messaging systems to RPC

Conclusions

Empirical results prove the applicability of symbolic learning techniques to the problem of automating the capture of data contained in a document image

Research issues

- The space inefficiency of incremental decision tree learning systems when examples are described by many numerical features
- The importance of first-order symbolic/numeric descriptions for document classification and understanding
- The importance of taking into account dependencies among logical components for document understanding

Future work

- Segmentation algorithm handling color images and forms, where textual content is typically surrounded by frames
- Application of similar techniques (classification, understanding, etc.) to map processing in GIS applications and to web document processing
- Design of an information retrieval component to extract information *not-retrievable* through layout structure