Introduction

- Despite widespread use of computers, paper still used extensively.
- Benefits of Paper:
  - Easy to annotate
  - Fairly cheap and durable
  - High-resolution display
- Limitations of Paper:
  - Expensive to duplicate and archive
  - Markings made on paper are static and immovable (no way to transfer back to digital source)

Introduction (cont)

- Digital word processing is widely used.
- Benefits of digital:
  - Change content and reflow text easily
  - Easy to duplicate, search
- Limitations of digital:
  - Many people prefer to print documents for reading and proofreading/editing
Motivation

• Augment paper interfaces while preserving the flexibility of digital document.

• Ability to interact with documents via paper or computer (whichever is more convenient)

• Ultimate Goal: Seamless integration of paper and computers

Previous Work

• **Tight coupling**
  - DigitalDesk [Wellner 93], Ariel [Mackay]
  - A-Book [Mackay 02]
  - PaperLink [Arai 97]
  - Intelligent Paper [Dymetman 98]

• **Paper as input device**
  - Xax [Johnson 93]
  - Anoto
  - Paper PDA [Heiner 99], [Avrahami 01]

• **Digital emulation**
  - FreeStyle system [Wang 89]
  - MATE [Hardock 93]
  - XLibris [Schilit 98], [Golovchinsky 02]

• **PADD** [Guimbretière 03]

Previous Work (cont)

• PADD [03] and XLibris [02] seem to offer best solution
  - Paper and digital sources are decoupled
    • User can continue to use each in the manner in which they are familiar
  - Supports digital marking and reflow

• What’s missing?
  - PADD [03] system lacks support for distributed use
  - Document contents can’t change (only layout)
Our Work

• Set of primitives that support transfer between paper and digital source
  – PADD Infrastructure: Infrastructure for distributing storage of documents and annotations
  – TAbiWord: Provide digital support for paper-based markings that are able to reflow as the document’s content, structure, or layout changes

PADD Infrastructure

• Four use cases

<table>
<thead>
<tr>
<th></th>
<th>Single version</th>
<th>Multiple versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single user</td>
<td>Print, annotate</td>
<td>Print, annotate, update digitally, print again</td>
</tr>
<tr>
<td>Multiple users</td>
<td>Multiple printings, multiple annotations</td>
<td>Multiple printings, multiple annotations, updates made digitally, selective reprints</td>
</tr>
</tbody>
</table>

PADD - Motivation

• Single user – single version
  – Immediately able to do
• Single user – multiple versions
  – Need a way to map strokes to documents
• Multiple user – single versions
  – Need a way to map users to strokes
• Multiple users – multiple versions
  – Need a way to provide file permissions
PADD Database

- Maps strokes to documents
- Maps users to strokes
- Maps files to file stores (FTP, ...)
- Records all events
  - Printings
  - Strokes storing
- Map users to files and events
- Provides a standard interface to the data
  - PADD Server API
  - Clear DB design
- Remains protocol-independent
  - Paper
  - Editor

PADD DB Client

- Need to inform the database when documents are printed
- Who knows when this happens?
  - Editors (Acrobat, TAbiWord)
  - Printer drivers
- Clients are very thin
  - 7 different operations to support
    - Editors need only 3 of these
    - Library available

PADD DB - Limitation

- PADD Database solves the issues for all use cases
- **Does not scale**
  - All printings and annotations at one database?
- Need a way to distribute the load
  - Clear delineation: paper distribution
    - Anoto sells page ranges (28.0.47.0 – 28.0.47.127)
PADD Directory

- Each database is responsible for a given (set of) page ranges
  - All printings and annotations
  - Database per administrative domain
- PADD Directory
  - Similar to DNS
  - Provides the mapping from page address to administrative domain (PADD DB)
- Databases remotely accessible

1. Locate the DB
2. Store the info
1. Locate the DB
2. Store the info
3. Store the files

Distributed DBs - Results

- Distributing the PADD infrastructure brings about:
  - The desire to store files in a distributed way
  - A decoupling between paper and digital
    - Manual distribution still allows digital access
  - The ability to employ orthogonal file processing
    - Asynchronous "strokes collectors"
    - CVS
  - The need for file permissions
  - A peering of the servers

File Processing

- Strokes collector
  - Incorporates a given strokes file and document
- Would like this to happen asynchronously
  - Save time
    - Large documents
    - Complex incorporation (doesn’t have to be so straightforward)
  - Cache results

Notification System

- General scheme: notifier and subscriber
- DB Server knows when events happen
- Strokes collectors can be separate processes that subscribe to events
Notification System (cont)

- Other processes can subscribe to events
  - Authors of documents
    - When are annotations made
    - When are documents printed
  - Annotators of documents
    - Auto-open documents digitally once annotated
      - Having one sheet of a physical copy gives you access to the document digitally
TAbiWord

- Extension of AbiWord (www.abiword.org) with support for digital annotation and reflow
- Addresses issues raised by XLibris [02] as well as issues of digital reflow in documents with dynamic content

Ink Source

- Mouse, WACOM, Tablet PC stylus
- PADD system allows for paper-based annotations
- Allows users to select the annotation interface most appropriate for their task

Ink Association

- Determine context in which annotation is made
- Semantic meaning of the mark must be preserved
- Focused on ANSI standard proofreading marks
  - Contextually associated with a single letter or word
  - Drawn over several words or lines

Ink Association (cont)

- Determine sharpest change in direction
  - Law of Cosines + Sliding Window (size >= 13)
- Associate with text that overlaps this point

(Insertion point denoted by blue circle.)
Ink Storage
- Stroke anchored near target text as a document element
- Digital reflow more efficient
- Allows document’s content to change

Ink Reflow
- Ink must stay with semantically correct text during layout
- Content changes (e.g. add/delete text)
- Formatting change (e.g. margins/font size)
- Zooming and scrolling

Ink Reflow (cont)
- Each stroke saves initial location in layout
- Stroke stored as a document element
  - Layout engine will process it in as it renders semantically correct text
- When stroke redrawn by layout engine, check to see if position has change and translate to appropriate position

Before Reflow: Initial content with annotation
Ink Clipping

- Some markings are large (e.g. cross out mark)
- After reflow, semantic text may span two or more lines
- Break up large strokes into smaller strokes which are associated with semantically correct text
  - Clip stroke to non-hyphenable words
- Allows reflow across multiple lines

**Before Reflow**: Initial content with annotation

**After Reflow**: Initial content clipped over several lines
Analysis

- Recall our ultimate goal: Seamless integration of paper and computers
- Bridge between paper & computer
  - Users able to print documents and interact in a familiar, natural manner
  - Markings translated automatically from paper to computer
  - Digital reflow allows users to continue editing and writing process on the same revision

Analysis (cont)

- Writing process (existing):
  - (new revision)

Analysis (cont)

- Writing process with TAbiWord/PADD:
  - (same or new revision)

Analysis (cont)

- Writing process with TAbiWord:
  - (same or new revision)
Future Work

PADD
- File permissions
  - Varying granularities (private annotations)
- Exploit peering of the servers
  - Many existing peer-to-peer benefits
    - Redundancy, censorship resistance, etc.
- Solve the paper problem
  - Finite set of paper – how to deal with page address collisions?
- Caching at the client
  - Remove the assumption that a connection to a server always exists

Future Work (cont)

TAbiWord
- Additional work needs to be done to support reflow of freeform annotations
- Users have expressed interest in automated correction/application of proofreading markups
  - Limited by accuracy of handwriting recognition engine