Demo Abstract: SmartPublisher – Document Creation on Pen-Based Systems Via Document Element Reuse

Fabrice Matulic
Ricoh Software R&D Group
1-1-17 Koishikawa, Bunkyo-ku,Tokyo, Japan
{lastname}.dm.{firstname}@nts.ricoh.co.jp

ABSTRACT

SmartPublisher is a powerful, all-in-one application for pen-based devices with which users can quickly and intuitively create new documents by reusing individual image and text elements acquired from analogue and/or digital documents. The application is especially targeted at scanning devices with touch screen operating panels or tablet PCs connected to them (e.g. modern multifunction printers with large touch screen displays), as one of its main purposes is reuse of material obtained from scanned paper documents.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *GUI, Input devices and strategies, Interaction Styles;* I.7.5 [**Document and Text Processing**]: Document Capture – *Document Analysis*

General Terms

Performance, Design, Human Factors

Keywords

GUI, reuse, scanned document, document creation and editing

1. INTRODUCTION

When creating a new document, it is very often the case that one wants to reuse elements from other documents and external information sources, rather than compose the document entirely from scratch. A typical example is picking out particular pieces of text and images from a collection of heterogeneous documents and using them to create a report or a presentation, much like cutting out selected articles of newspapers and laying them out on a white sheet of paper. It is both a technical and a design challenge to provide adequate software tools to support this task in an efficient and intuitive way.

SmartPublisher is an integrated system, which enables the user to compile and publish documents in a few minutes using only a pointing device, such as a stylus pen. The concept is somewhat reminiscent of Wellner's DigitalDesk [1], where new digital documents could be composed using elements extracted manually from paper originals. However SmartPublisher operates in an entirely different setting (scanners with touch screens instead of a desk with overhead cameras) and the level of integration and functionality is taken much further.

Using SmartPublisher installed on a copier for example, elements of documents acquired from the scanner, digital cameras, removable media or from the network can be immediately reassembled to create new documents for printing "on the spot". Document component extraction is quick and intuitive, as users can take content material both from analogue and digital information sources seamlessly using uniform graphical interfaces. Text elements are automatically extracted as needed using segmentation and OCR for document images or retrieved directly from the document structure for digital documents.

2. THE APPLICATION

2.1 Software Architecture

SmartPublisher is made up of several modules built around a common Document Element Object Model to allow easy transfer of document components (mainly text and image objects) between modules using simple drag-and-drop or copy-paste actions. Input material can be gathered from various sources such as scanners, digital cameras, network file systems and the Internet and laid out on a virtual canvas with page structure to create new documents.

In the software architecture, each module is defined either as a Document Content Source (DCS), a Document Content Target (DCT) or both. When a document is opened in a DCS, its structure is analysed (through document image segmentation or parsing of the data structure) and a content graph is built internally containing objects that encapsulate the different constituent elements of the document with their properties. This approach of considering document elements as separate transferable objects allows their immediate reuse while the graph structure is used to support various kinds of relationships between those objects such as reading order, text-image associations, tagging etc. DCTs are modules that accept or import document elements and use them for various purposes. Examples of DCTs are the Editor, the Media Browser and the Content Pool (see Description of the Modules further below).

2.2 User Interface

The basic framework of the user interface shows two modules side by side so that contents can be easily moved from left to right and vice-versa, depending on whether they are DCSs, DCTs or both (see Figure 1). The user can switch to other modules via two pull-down menus located at the top of the main frame. This layout

DocEng'06, October 10–13, 2006, Amsterdam, The Netherlands.
ACM 1-59593-515-0/06/0010.

is suitable for medium-sized screens such as Tablet PCs but different arrangements are conceivable for larger displays like wall screens, where more modules could be shown to make use of the larger screen space.

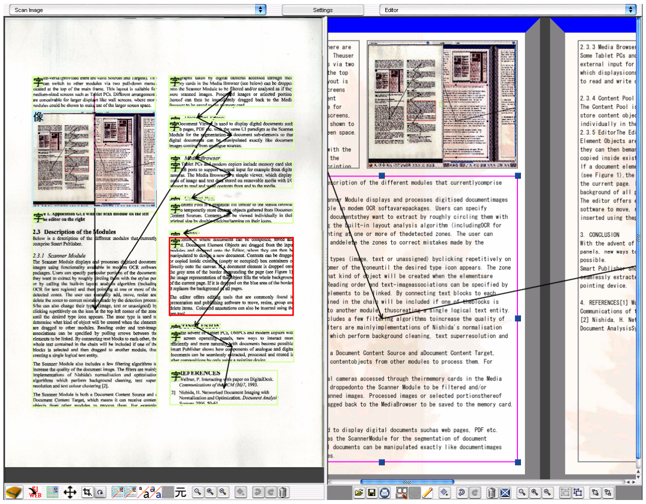

Figure 1. GUI with the Scanner Module on the left and the Editor on the right. Black lines indicate reading order. Kanji icon and bounding box colour indicate object type.

2.3 Description of the Modules

2.3.1 The Scanner Module

The Scanner Module displays and processes digitised documents using similar functionality to that available in some commercial OCR packages. Users can specify particular portions of the documents they want to extract by roughly circling them with the stylus pen or by calling the built-in layout analysis algorithm (with OCR for text regions) and then selecting one or more of the detected zones. The user can manually add, move, resize and delete the zones to correct mistakes made in the segmentation process. S/he can also change their types (image, text or unassigned) by clicking repetitively on the kanji icon at the top left corner of the zone until the desired type icon appears. The zone type is used to determine what kind of object will be created when the elements are dragged to DCT modules. Reading order and text-image associations can be specified by pulling arrows between the elements to be linked. Connecting text blocks together will ensure that the whole text contained in the chain will be included if one of the blocks is selected and then dragged to a DCT module, thus creating a single logical text entity.

The Scanner Module also includes a few filtering algorithms to increase the quality of the document image and hence its subelements. The filters are mainly implementations of Nishida's normalisation and optimisation algorithms which perform background cleaning, text super-resolution and text colour clustering [2].

The Scanner Module is both a DCS and a DCT, which means it can accept Content Objects from other modules to process them. For example, photographs taken by digital cameras accessed through their memory cards in the Media Browser (see below) can be dropped onto the Scanner Module to be filtered and/or analysed as if they were scanned images. Processed images or selected portions thereof can then be immediately dragged back to the Media Browser to be saved to the memory card.

2.3.2 The Document Viewer

The Document Viewer is used to display digital documents such as web pages, PDF etc. with the same UI paradigm as the Scanner Module for document segmentation so that digital documents can be manipulated exactly like document images coming from analogue sources.

2.3.3 The Media Browser

Some Tablet PCs and modern multifunction printers include memory card slots and USB ports to support external input for example from digital cameras. The Media Browser is a simple viewer, which displays icons of image and text data stored on removable media with I/O support to read from and write contents to the media.

2.3.4 The Content Pool

The Content Pool is a graphical list similar to the Media Browser used to temporarily store content objects gathered from DCSs. Contents can be viewed individually in their original size by double-clicking/tapping on their icons.

2.3.5 The Editor

The Editor is where documents can be composed, saved and printed. Document Element Objects are dragged from the input modules and dropped onto the Editor, where they can then be manipulated to design a new document. Contents can be dragged or copied inside existing (empty or occupied) box containers or directly onto the canvas. If a document element is dropped onto the grey area of the border surrounding the page (see Figure 1), the image representation of that object fills the whole background of the current page. If it is dropped onto the top blue area of the border, it replaces the background of all pages.

The editor offers editing tools that are commonly found in presentation and diagramming software to move, resize, group and delete items. Text is automatically reflowed inside the container chain if one of the boxes is resized. Coloured annotations can also be added using the pen tool.

3. CONCLUSION

With the advent of Tablet PCs, UMPCs and modern copiers with touch screen operating panels, new ways to interact more efficiently and more naturally with documents become possible. SmartPublisher shows how pieces of analogue and digital documents can be seamlessly extracted, processed and reused in other compositions without resorting to complex DTP software.

4. REFERENCES

[1] Wellner, P. Interacting with paper on DigitalDesk. *Communications of the ACM (36)7*, 1993.

[2] Nishida, H. Networked Document Imaging with Normalization and Optimization. In *Proceedings of Document Analysis Systems VII*, 2006, 50-61.