Xerox PARC PageMill Project Recollections

Leigh L. Klotz, Jr
2024-01-24
Draft 2024-03-09
https://klotz.me/2024/01/pagemill/xerox-parc-pagemill-project-recollections.pdf

The Xerox PARC PageMill project was a decade-long effort to bridge the Paper and Electronic worlds and prepare Xerox for a future digital transition from light-lens to digital copying. The goal was to make paper more valuable to consumers and business, and to provide an entree for Xerox to offer subscription services and to distinguish itself from rivals offering just copies.

I was involved in the research and productization efforts at Fuji Xerox and later at Xerox, from 1989-2005. As far as I know, there is no detailed history of this fascinating peri-web era, showing how several groups of people were all trying to invent the same thing, a way for people to use computers to communicate with each other bout documents seamlessly. We used Fax Modems, Copiers, Scanners, and checkboxes on Scanned images and 2d barcodes.

Below are notes and recollections. I will find and scan in artifacts.

Many more people were involved, at Xerox PARC, Xerox, and Fuji Xerox. The project eventually resulted in the creation of ScanSoft, Pagis, Windows 98 drag-and-drop and document summaries (“licensed” from ScanSoft :-), and eventually the creation of Nuance.

DocuStation IM 200

There’s little about Page Mill, DocuStation IM200, Paperworks, etc in open literature. Here’s one of my Fuji Xerox colleagues reminiscing a small bit:

https://logmi.jp/tech/articles/329157 ->
https://logmi-jp.translate.goog/tech/articles/329157?_x_tr_sl=ja&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp

PaperWorks

PaperWorks, the feature-reduced market probe we launched for Windows 3. It got a lot of attention, but was soon eclipsed by $99 fax and modem boards that enabled online BBS and eventually AOL and such on home and office PCs.

https://en.wikipedia.org/wiki/PaperWorks

Here's a John Markoff article about PaperWorks, the PC product: https://www.nytimes.com/1992/03/24/business/company-news-new-xerox-software-set-for-offices.html?unlocked_article_code=1.QE0.8HF2.6eSInBQ3yCv4&smid=url-share

Shortly after, we took out a full-page add that just had "PaperWorks", "Xerox" and a 1-800 number we set up to handle orders.

Original UX design for PaperWorks was done by Walter Johnson and me. I proposed the metaphor of documents in filing cabinets with index cards. After user testing dropped the card catalog and went with one type of folder for metadata / keywords, and one type with all documents. Removing from all documents deleted the document but removal from the keywords folders just changed the filing keywords. The UI was drag and drop, a first on Windows.

Portal Phi

Z Smith, who led the Page Mill project for the bulk of its research and productiziation, created many concepts for the future. One of the most interesting was a desktop scanner/printer called "Portal Phi," into which you could put documents and form/cover sheet pages. Hardware developed never matured on this product, but it was an important think piece for communicating research ideas and direction.

XaX and Xax2

XaX was developed by Herb Jellinek, Walter Johnson, and others. It was a shell-script and C-based Paper UI scanning and printing system.

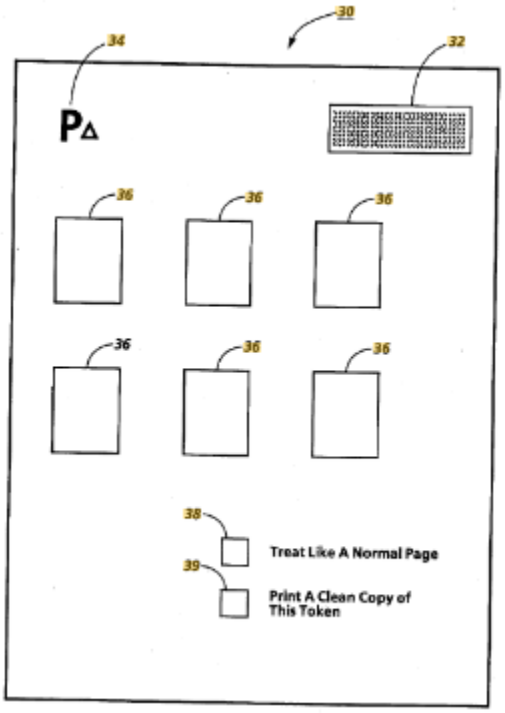
Stan Lanning and I developed XaX2. We investigated Elk Scheme and Python 0.0 as implementation layers for a virtual-machine that would power data and information transfer and communications across Fax via images, and locally via copies. I downloaded Python 0.0 from alt.sources or some similar Clarinet-fed Usenet group. It was a clean and simple language, missing some stuff, but no libraries or rendering model at all. Stan implemented a meta-object protocol (Gregor Kiczales was interested) and used it to generate an object-to-SQL mapping for us to store state and data. Xax2 was an event-driven system with this OODB for storage, so that it could keep track with the physical paper paths in scanner and printer and not get confused when pages got re-fed.

Cover Sheets

Cover Sheets were a big part of all the implementations. A cover sheet had 3 “fiducial” marks, a modified ‘P’ logo for recognition, a rectangular DataGlyph 2d-barcode with form UUIDs, and a content area with text, images, checkboxes, and write-in fields. The idea was to select options (“email to myself” or “store under Invoices”) and put the paper in front of the document and press the Big Green Button (similar in feel to the “Easy Button” of later years). Users were universally surprised by how well it worked and how easy it was to use, compared to the Xerox devices’ hard-button panels and small monochrome LCD of the era.

Cover sheets also had a “please reprint” checkbox at the bottom to get a clean copy.

Document Tokens



I invented Document Tokens in Tokyo and wrote the first doc on a Mac that Nobuhiko Ohki had in his Fax Product Planning office, in January of 1990. Z Smith re-did it as a Globalview document to appease Xerox interests and accidently changed the date to 1991.

A Document Token was designed to be a piece of paper that is a physical icon for a document. It contained a picture or summary of the document and no use of checkboxes (other than the re-print one). If you put a Document Token into a Page Mill enabled copier, it would give you a copy of the document itself, dereferencing it from the copier’s local storage. If you used a cover sheet, it would operate on the Document. Fax, ditto.

Meg Withgott and Ramana Rao independently and about a year later worked on a keyword-based summary sheet of a document, so we combined the ideas together in a single patent application, including Weather Map as well.

Protofoil

Protofil was pre-product at PARC, before some of the above. Protofoil was a thick client for a paper-based document management system. Ramana Rao was the principal author, and I contributed the conclusions and the backend implementation.
https://dl.acm.org/doi/pdf/10.1145/191666.191738 There’s also another paper.

I didn't like the thick client and wanted a thin client and a smart server, similar to Larry Masinter's System 33. Tim Berners-Lee also visited and got may of the same ideas from System 33.

DAE

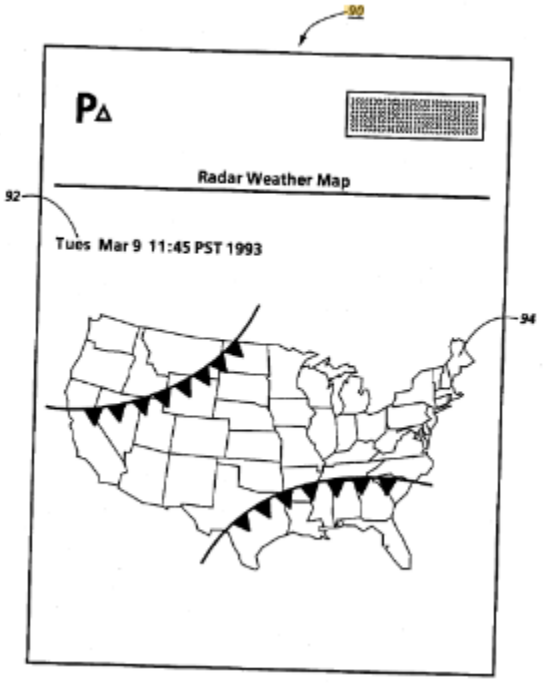
The DAE was my design for a Document Applications Environment. I came up with the idea of using PostScript as the virtual machine and adding image processing commands to PS in order to sell the idea of a server inside teh copier to Xerox and Fuji Xerox. It worked quite well but we found programming OODB and the like was difficult. Stan wrote the OODB system for it in PostScript, and Daniel Davies, Julia Craig, Xianing Zhu, and Dan Bloomberg worked on the imaging subsystem, and MLY worked on the C interface and implementation. Art Medlar and others worked on applications.

Henry Minsky Internship / Wall of Applications

We hired Henry Minsky as an intern and he developed a number of applications in DocuScript. Others worked with him as well. There was a video of about 15 applications Henry wrote. We put all the paper on the wall and showed it to visitors.

Henry made multiple contributions over the years, including the Universal Access work mentioned later. He also pioneered using Reed-Solomon coding in DataGlyphs to correct against two dimensions of correlated errors and random single bit errors.

Other Cute Applications



WeatherMap - When you copied it, it gave you the new weather map. So the copier copied the type rather than the instance. We thought this showed a new way of thinking about documents and of making paper more valuable. It used FTP to get the weather.

PaperFiche - this was one of my inventions. N-to-1 copying: put in 18 pages, get out one 3x3 tiled double-sided copy at 200dpi fax resolution. And the inverse operations as well. No patent.

Paper Floppy: Another of mine, which did get a patent. Drag and drop documents to an icon on your desktop, get a printout of the data in DataGlyphs. Put it back in your

copier and it acts like a document token for the document: you can print it (if it's printable) or email it with a cover sheet, or just scan it and have it re-appear in your inbox. DataGlyphs weren't very dense (at least 25x expansion of bits-to-pixels) so this invention had a limited lifetime due to increasing data sizes.

Double Spacing Copier: Lawyers would print out contracts they received as WordPerfect documents via fax and have them re-keyboarded by typists double spaced, then make markup with pen, then send it back to be updated in Word Perfect, then printed and faxed. The Double Spacing copier eliminated the first few steps. Companion single-spacing copier patent 🙂

DocuStation IM 200

The DocuStation IM 200 was a Fuji Xerox multifunction printer/fax/scanner and we embedded Solaris and DAE with GhostScript in it. Yoishiki Shibata ran the hardware interface layer and OS. Jun Miyazaki ran the applications layer at Fuji Xerox. We also had a team at Xerox doing development. There is open web info on this product. It supposed Cover Sheets and Document Tokens and the ability to dynamically download and add new applications.

DocuScript

We created DocuScript from GhostScript and licensed GS from Peter L. Deutsch. Peter adopted a GPL variant and dual-licensed it for GPL and commercial use, a first at the time and created partially for us and for other customers of Aladdin. We added the imaging operators based on Dan Bloomberg's Alpaca library.

Alpaca

Dan Bloomberg developed expertise with document image processing and used Mathematical Morphology, championed by Luc Vincent and others. It's based on convolution kernels, but unlike the CNN networks of today, these were hand-tuned structuring elements.

Some version of Alpaca was defensively published in patents, for example: https://patentimages.storage.googleapis.com/7b/29/9e/9ddd6790ca9683/US5570435.pdf

IPShared / IPCore

We split the Alpaca codebase into two libraries, a C library for direct use in programs, with its own memory management (malloc/free) and bitmap model, and a core model with no memory allocation and only bit buffers. Dan Davies was chief architect of IPCore and I designed the API

between them. We used IPCore in GhostScript and could perform the morphology operations and other imaging operations on rough 1MB buffers.

PdB

I found the Turning Institute in Edinburgh had developed a C++ to PostScript compiler and we licensed it for $10k and I extended it to support Stan's OO notation and other hacks such as if/else with different parameters in the arms through the stack. We wrote tons of code in it, with folks such as Richard Hyde and David Sobeck.

PdB was developed by Arthur van Hoff, Tim Niblett, and later Don Hopkins. I'd known Don Hopkins since he was a young teen hacker and we paid him $50 for his Logo adventure game for the Commodore 64 and published it with Logo for the C64. Shortly after we licensed it, Arthur left and joined First Person and took the PdB code with him and made it the Oak compiler, so it retargetd a register machine instead of the original PostScript stack machine.

In other words, we were using web concepts adn Java before there was web Java and using language virtual machines to run it, with many of the folks on the same team as the original Java team, aimed at a similar space.

OODB

We used a commercial OODB vendor or two but since I'd worked on Statice at PARC (porting it to C++ using a Lisp to C++ compiler I wrote), when we began to have vendor cost and performance problems I hired Brian Fox to write a version in GPL using GDBM, the ISAM database. For the product I ported it to Faircom ISAM and we gained performance, reliability, and cost.

Universal Desktop Browser

Herb Jellinek worked on the Universal Desktop Browser, which was supposed to retrieve from the server and offer a Paperworks-like experience on the desktop. This was clearly office-based as it required TCP. After the web browsers hit the world, Herb left and went to First Person, where he wrote a real web browser, Hot Java, in Java, newly renamed from Oak. It was compiled using the descendent of the PdB compiler that Arthur wrote.

Pierre Wellner and Mik Lamming, EuroPARC

Pierre Wellner and Mik Lamming wrote a paper on a desktop Paper UI system with an overhead camera, and beat our publication date for the citation of "paper user interface."

DataGlyphs

DataGlyphs were a 2-d barcode designed to look like a grey patch to the eye but be easily scanned, using orthogonal forward and back slashes (45 degree angle) to represent 0 and 1. 3x3, 5x5, etc were small sizes but eventually we had to use 200dpi fax resolution glyphs. There were multiple implementations and designs over the years by Rob Tow, Dan Bloomberg, David Hecht, Dan Davies and Julia Craig. Xerox made a few attempts at commercializing it, and we used it for cover sheets, but as the web and URLs began to make progress, and hand-held scanners did not, the window of opportunity closed. QR codes instead came about, and they are highly visible, instead of highly invisible. There is much open literature on DataGlyphs but even more in the darkness of history.

Epcot 2000

I was selected by Xerox to oversee ⅔ of an Epcot center exhibit to be opened in World of Tomorrow for the year 2000. Working with Rich Gold for ideas, I drew up plans to answer the request to have a copier do something interesting on stage for a 7-10 year old child with an audience. I spent days under Epcot center in Buena Vista, FL working on making DocuStamp work, stickers we had custom-cut from avery with just glyphs. We used them to make shortcuts to actions on the copier, for example you could paste a sticker to a cover sheet, put in a document after it, and get a sheet full of those stickers back, registered to be a document token for that document. They were, in essence, macros fro other paper UI components.

At Epcot they used it to refer to a scanned map and menu of Disney restaurants in Epcot.

David Hecht had “Glyph Glasses,” which we would now call 2d-bar-code enabled virtual reality. It was a more fragile exhibit, ahead of its time.

There was a third exhibit called “Vortex of Knowledge” and it was a multi-media presentation projected on the inside of a tip.

Also with my hacking away underground was a team from Disney Innovention labs, and they wrote a car race game in Squeak, the Disney SmallTalk implementation.

When the exhibit was opened, it was next to last, right after the Motorola “try out these new flipphone MP3 players and give us feedback,” extremely compelling at the time. The last exhibit was Varian Oncology devices, ending the World of Tomorrow on a somber note.

Slashdot Articles / Comments

https://hardware.slashdot.org/comments.pl?sid=65366&cid=6033275
leighklotz

Actually, Xerox did sell it, in Japan, as the DocuStation IM 200. When Java came out, we and otehrs worked with Sun to add the image processing features that were necessary (which became java2d) it was re-written in Java and sold again as FlowPort, and is still sold.

At the time the choice was made, we were examining Scheme, but felt a lot of resistance from the industrial engineering community we were targeting. So, although I helped develop 6.001 [mit.edu] and the book "Structure and Interpretation of Computer Programs" that introduced Scheme, we abandoned that approach and looked for a language that would be more palatable to the printer and copier engineers. The system was written in PostScript because it was an interpreted language that was capable of running inside hardware such as copies, scanners, and printers. There were hired industry pundits who had suggested that we use Visual Basic, but that was even harder to fit into a copier in 1991, so PostScript it was.

Just as we were making the decision, I saw on alt.sources a new small object-oriented language announced and tried it out, but it had absolutely no class libraries, and no tools, and nobody had hever heard about it before (some guy named Guido) so we passed up on Python...

The goal was to make paper be the universal access portal to information, and to piggyback on images as the universal information transfer medium. We did hyperlinks on paper, used dialup modems for transferring information, etc. Basically it was the web and web forms on paper. Now the focus is on capturing paper documents and their metadata and making them first-class citizens in the office network.

The DocuScript language was actually much more like Java than like Smalltalk. It did have an object-oriented database, which Java lacks, but consider the following:

> Much of the PostScript code was written in PdB, a C++-like language compiled to PostScript. PdB was written at The Turing Institute by Arthur van Hoff, who later went on to write the first Java compiler, with a remarkably similar syntax. So, the system was written in a precursor to Java with GS as the virtual machine.
> Herb Jellinek worked on the "configurable desktop universal browser" part of the project at PARC. He left and went to Sun to work on Oak and in the meantime, WWW happened and became the protocol for the "universal browser", and he wrote HotJava, which was the web browser that kicked off the Java revolution.
> The Paper User Interface forms were all done as small PostScript programs that, depending on which set of definitions was loaded into the environment, either rendered a printable image to the image buffer, or read the scanned image from the image buffer and read the checkboxes. The layout decisions were all done with PostScript routines.
> So, in that sense, the layout was like LaTex, where the formatting commands are actually short programs or macros that bottom out into an implementation of primitive operations. After the product was launched, Larry Masinter of PARC

convinced me that the LaTex-programmmatic approach was wrong, and that we needed to use a static description language, a path I had resisted because there were no good ones. But in the interim, again WWW had hit, and HTML seemed good. We did a Paper User Interface version of the WWW (now going full circle from our original idea of paper access to information to paper being a proexy for access to information via the WWW) and we made a tool to print Paper UI on any web page.
Initially we did this as well in PostScript, but found that we needed something faster for the HTML parsing and layout, so we got a company called Universal Access to do that for us. They had a tool they were developing, and they prototyped it for us, and their other customer was a company called Unwired Planet that wanted to make a transcoder to convert HTML to a smaller binary markup language called HDML. So, UA did the converter for HDML and for Paper UI web forms. Unwired Planet changed its name to Openwave, and HDML became WML, the XML-based markup language for mobile phones.
In the meantime, Larry Masinter and others on the W3c HTML WG [w3c.org] looked at our problems building a reliable markup language that could be accessed both from paper and from desktops, and the successes and failures in the WAP/WML world, and started a new working group to separate forms from HTML, called XForms [w3c.org].

So, in summary, the Xerox effort JerryAsher describes above was intertwined with Java, HTML, WAP/WML, HotJava, and XForms, and was sold in two different products, and continues to be sold, and also continues to have an influence on the future direction of the web and document management in the office.

P.S. A good deal of the system was written in PostScript directly, so I figure that I've probably written 10K lines of PostScript. I'd say that other than NeWS, the done-in-PostScript window system written at Sun by James Gosling (there's the Java/PostScript connection again), it's probably the largest system ever written in PostScript.

Patents

I joined the Xerox TAP-20i patent committee an learned a lot about software patents, and acquired a few. Here are ones related to the Page Mill project.

These summaries are copied from Justia.com: https://patents.justia.com/inventor/leigh-l-klotz

Method for generating optical codes for a print-context

Patent number: 8711407

Abstract: One embodiment provides a system for printing a document from a portable device. During operation, the system captures an image of an optical code that identifies a printing device, wherein the optical code is displayed on a panel of the printing device

or is printed by the printing device. Next, the system transfers information identifying the printing device to a remote printing service, thereby allowing the remote printing service to print a document at the printing device.
Type: Grant
Filed: April 4, 2011
Date of Patent: April 29, 2014
Assignees: Xerox Corporation, Palo Alto Research Center Incorporated
Inventors: Kurt E. Partridge, Leigh L. Klotz, Jr., James M. A. Begole

User interface tag for use in processing a document

Patent number: 8640018
Abstract: A user interface tag for use in processing a document is provided. A printable surface is on one side of a document and an adhesive surface is on an other side of the document. The printable surface further includes a printed data field, including machine-readable marks of digital data encoding a service and a user identity; and a printed border surrounding the printed data field to define an iconic representation. A scanned representation of the machine-readable marks is decoded from the iconic representation to specify the user identity and the service.
Type: Grant
Filed: January 22, 2007
Date of Patent: January 28, 2014
Assignee: Xerox Corporation
Inventors: Leigh L. Klotz, Jr., Glen W. Petrie, Robert S. Bauer, Daniel Davies, Julia A. Craig

User interface tag for use in processing a service on a scannable document

Patent number: 8640019
Abstract: A user interface tag for use in processing a service on a scannable document is provided. A printable surface is on one side of the scannable document and an adhesive surface is on another side of the scannable document. The printable surface further includes a printed data field specified substantially within the printable surface, including machine-readable marks of digital data encoding a service code and a user identification number; and a printed rectilinear border surrounding the printed data field to define a rectilinear iconic representation. A scanned representation of the machine-readable marks is located by identifying the printed rectilinear border using corner candidates oriented in diametric opposition from among connected components identified on the document and the scanned representation of the machine-readable

marks are decoded from the rectilinear iconic representation to specify the user identification number and the service code.
Type: Grant
Filed: September 2, 2009
Date of Patent: January 28, 2014
Assignee: Xerox Corporation
Inventors: Leigh L. Klotz, Jr., Glen W. Petrie, Robert S. Bauer, Daniel Davies, Julia A. Craig

METHOD FOR GENERATING OPTICAL CODES FOR A PRINT-CONTEXT

Publication number: 20120250065
Abstract: One embodiment provides a system for printing a document from a portable device. During operation, the system captures an image of an optical code that identifies a printing device, wherein the optical code is displayed on a panel of the printing device or is printed by the printing device. Next, the system transfers information identifying the printing device to a remote printing service, thereby allowing the remote printing service to print a document at the printing device.
Type: Application
Filed: April 4, 2011
Publication date: October 4, 2012
Applicants: XEROX CORPORATION, PALO ALTO RESEARCH CENTER INCORPORATED
Inventors: Kurt E. Partridge, Leigh L. Klotz, JR., James M.A. Begole

User Interface Tag For Use In Processing A Service On A Scannable Document

Publication number: 20090323126
Abstract: A user interface tag for use in processing a service on a scannable document is provided. A printable surface is on one side of the scannable document and an adhesive surface is on another side of the scannable document. The printable surface further includes a printed data field specified substantially within the printable surface, including machine-readable marks of digital data encoding a service code and a user identification number; and a printed rectilinear border surrounding the printed data field to define a rectilinear iconic representation. A scanned representation of the machine-readable marks is located by identifying the printed rectilinear border using corner candidates oriented in diametric opposition from among connected components identified on the document and the scanned representation of the machine-readable marks are decoded from the rectilinear iconic representation to specify the user

identification number and the service code.
Type: Application
Filed: September 2, 2009
Publication date: December 31, 2009
Applicant: Xerox Corporation
Inventors: Leigh L. Klotz, JR., Glen W. Petrie, Robert S. Bauer, Daniel Davies, Julia A. Craig

User interface identification and service tags for a document processing system

Patent number: 7168036
Abstract: A tag-based user interface scheme for digitizing and processing hardcopy documents utilizes a sticker that includes a printed data code representative of a user identity code and a service code. When the sticker is applied to a hardcopy document and scanned, the sticker is located, the data code is parsed, and a desired service is performed based upon the information stored in the data code.
Type: Grant
Filed: November 13, 1998
Date of Patent: January 23, 2007
Assignee: Xerox Corporation
Inventors: Leigh L. Klotz, Jr., Glen W. Petrie, Robert S. Bauer, Daniel Davies, Julia A. Craig

Methods and systems for providing status information for reprographic operations

Patent number: 7057752
Abstract: Systems and methods for providing status information corresponding to a reprographic operation are described. The disclosed systems and methods create an audio message to provide the status information to a voice mailbox, pager, or telephone number. The audio message may include any relevant information regarding a reprographic operation such as status of the operation, success or failure of the operation, a telephone number to which the document was transmitted, document scan time, and the number of pages processed. When a request for a reprographic operation is received, the system obtains information identifying a location to provide status information corresponding to the reprographic operation. After receiving the request for a reprographic operation, the system may monitor the reprographic operation and obtain status information corresponding to the reprographic operation.
Type: Grant
Filed: December 14, 1999

Date of Patent: June 6, 2006
Assignee: Xerox Corporation
Inventor: Leigh L. Klotz, Jr.

USER INTERFACE IDENTIFICATION AND SERVICE TAGS FOR A DOCUMENT PROCEESSING SYSTEM

Publication number: 20040205626
Abstract: A tag-based user interface scheme for digitizing and processing hardcopy documents utilizes a sticker that includes a printed data code representative of a user identity code and a service code. When the sticker is applied to a hardcopy document and scanned, the sticker is located, the data code is parsed, and a desired service is performed based upon the information stored in the data code.
Type: Application
Filed: November 13, 1998
Publication date: October 14, 2004
Inventors: LEIGH L. KLOTZ, GLEN W. PETRIE, ROBERT S. BAUER, DANIEL DAVIES, JULIA A. CRAIG

Method for avoiding creation of duplicate keyword objects representing user entered data on a machine readable form

Patent number: 5793495
Abstract: A method for maintaining the uniqueness of user entered data in a system for processing machine readable forms. Use of such a method in a system for processing machine readable forms facilitates forms reuse by creating keyword objects for new instances of user entered data, thus avoiding the creation of duplicate data. When user entered data is encountered in a machine readable form, the user entered data is compared to keyword objects of previously encountered user entered data. If no match is found, a new keyword object is created for the user entered data. Otherwise, it is assumed that the user entered data is a copy of previously entered data.
Type: Grant
Filed: June 24, 1996
Date of Patent: August 11, 1998
Assignee: Xerox Corporation
Inventor: Leigh L. Klotz, Jr.

System for representing electronic files using a paper based medium

Patent number: 5682540

Abstract: The present invention involves a novel form and methods for creating and using such forms. These forms are summaries (hereinafter "document surrogates") of associated objects, such as original documents, processes, or their copies, stored in a document processing system. A document surrogate made in accordance with the principles of the present invention comprises at least one sheet of a information storing substrate material that has a human readable area and a machine readable area. The human readable area may contain at least one area of material which summarizes the associated object. Such a summary may either be made manually by the user or created automatically by the document processing system. The machine readable area comprises a document reference code that is readable and recognizable by the document processing system. The code is located by the system and recognized from an image of the entire page. The reference code encodes an indicator to the storage location of the associated object.

Type: Grant

Filed: December 8, 1994

Date of Patent: October 28, 1997

Assignee: Xerox Corporation

Inventors: Leigh L. Klotz, Jr., Ramana B. Rao, Walter A. L. Johnson, M. Margaret Withgott

Paper saving reprographic device

Patent number: 5642473

Abstract: A reprographic device having means for eliminating undesired spacings between lines of text in copies of a source medium. Such means will often have the desired effect of reducing the number of pages in a multi-page document. The present invention accomplishes this by scanning the multiple pages to create bit-mapped images, segmenting the images to identify lines of text and graphics, identifying a distance X between segments, identifying a segment spacing factor which will cause reduction in the spacing between lines of text, laying out the segments in a print page memory so that the segments are separated by a distance based on the distance X and the segment spacing factor and printing out the contents of the page memory.

Type: Grant

Filed: October 17, 1994

Date of Patent: June 24, 1997

Assignee: Xerox Corporation
Inventor: Leigh L. Klotz, Jr.

Reprographic device for making copies with multi-spaced lines

Patent number: 5528732
Abstract: A reprographic device providing means for increasing the amount of space between lines of text when copying a source medium. The present invention provides a means to obtain a copy of a document having room between lines of text for hand written annotations e.g. it is double spaced.
Type: Grant
Filed: October 17, 1994
Date of Patent: June 18, 1996
Assignee: Xerox Corporation
Inventor: Leigh L. Klotz, Jr.

System for storage and retrieval of digitally encoded information on a medium

Patent number: 5459307
Abstract: The present invention involves a novel form and method for creating and using document sized file storage sheets containing one or more files. The present invention employs a user model similar to that of a floppy disk. A storage sheet made in accordance with the principles of the present invention comprise at least one sheet of an information storing substrate material, commonly paper. A system for the inputting and processing of storage sheet images is disclosed. The storage sheet comprises a machine readable mark (a "file storage sheet flag") which alerts the system is currently reading a file storage sheet. Upon image input, this flag is deliberately scanned for by the system. The storage sheet also comprises a machine readable directory that informs the system about the general characteristics of the files stored on the sheet. Such general information may include the name of the files, the number of bytes, creation date, owner, the location of the first byte of the file on the sheet, and the like.
Type: Grant
Filed: November 30, 1993
Date of Patent: October 17, 1995
Assignee: Xerox Corporation
Inventor: Leigh L. Klotz, Jr.

More References

There were many more people involved, and much more published in the academic press.

- https://www.researchgate.net/publication/221253781_Chipless_ID_for_paper_documents and https://www.cse.lehigh.edu/~lopresti/Talks/2004/ChiplessID.pdf [mentions PaperWorks as prior art]
- http://www.tauzero.com/Rob_Tow/DataGlyph.html [Rob Tow on Data Glyphs]
- https://www.wired.com/1994/12/smart-paper/
- https://landley.net/history/mirror/timelines/xeroxparchist.html [PaperWorks 1992]