The HCI Paradigm of HyperPrinting

Thomas Kieninger and Andreas Dengel
German Research Center for Artificial Intelligence (DFKI)
thomas.kieninger@dfki.de, andreas.dengel@dfki.de

Abstract

Today, printing and reverse printing (scanning, OCR, logical labeling etc.) technologies have become quite mature and thus allow for an easy transition of documents between physical and electronic world. However, there is no technology today which supports the lossless interpretation of paper-based user interaction with direct effects upon the electronic representation of that document.

The HyperPrinting environment tries to fill in this gap and thus accounts for the personal favors of a majority of office workers: Not only managers and knowledge workers prefer to read longer documents, articles or news from paper in contrast to a computer monitor or handheld computer. With the help of HyperPrinting, users can annotate, send notes or initiate tasks and it thus offers a completely new paradigm in the usage and treatment of paper documents.

As a side-effect, the use of HyperPrinting builds up a document repository which is not only searchable by full text but also by meta-information, which in turn is depending on the selected user scenario.

1. Introduction

Despite the prediction of the paperless office, paper consumption has ever increased over the recent 20+ years. Amongst all the reasons for this trend, there are a few which we consider most important: Humans prefer working/reading with paper as it is more natural and comfortable, it can easily be carried along, no technical devices are needed, it can be handled with little care or can be disposed anytime due to its little material value.

At the other hand, all notes and markups that the user writes or draws onto the paper document can only be interpreted by a human. All the semantics of the user’s notes will not be understood by the computer once the paper is scanned again.

But this information is meaningful and therefore it could be useful to some sort of application understanding and interpreting them. This is the point, where HyperPrinting jumps in: Giving a specific application domain and appropriate documents, one could specify a set of well defined use cases that can be associated with special gestures and markups on the printed page.

After starting with the general vision of HyperPrinting, the paper describes the basic components that are needed to build an appropriate environment – independent of a specific domain. Then we describe two alternative user scenarios, one in the domain of scientific papers and the other in the domain of patent applications. The latter one was focused for a prototypical implementation and will therefore be described in more detail. We will end with an outlook on future scenarios.

2. The HyperPrinting Vision

The vision of HyperPrinting was born out of the observation that today’s office environments are often equipped with combined printing and scanning devices, so called MFPs (Multi Functional Printer). But these MFPs are most typically used as printer, scanner or copier only. HyperPrinting aimed at the creation of a new type of service, which makes use of the combined functionalities and thus defines a unique selling point for those devices.

The idea was allowing the user to make his notes on the printed document in a natural and intuitive way, just as most users do today. And after scanning of that document the system will be able to identify the user annotations, classify them and interpret them in the context of the printed text.

There are in fact similar applications that found their way to the market already: The Forms Design Kit [1] allows the creation of forms which can be filled out with a digital pen. The user data can then e.g. be stored in a database record automatically. This tool also integrates script recognition technology [2] with very good recognition rates. However, this approach requires the use of a digital pen [3] and an almost invisible pattern [4] that is printed onto the paper together with the form in order to work properly.
Our approach differs significantly: Firstly, the user can use any arbitrary pen or pencil. And secondly, the annotations shall significantly differ from form fields that are to be filled out: The user may markup arbitrary text passages by e.g. underlining them or putting brackets around etc. He may write notes to the margins and check important paragraphs etc.

Nakai et al [5] proposed a fast method for the extraction of user annotations on printed documents by bitwise comparison of original and aligned annotated document; but: annotations will not be interpreted. In that point HyperPrinting differs significantly: Each type of annotation taken must be associated with a well defined semantics in order to make them interpretable by the computer. In order to best guide and assist the user in his actions, HyperPrinting also draws specific markers to the printed image, the so called HyperMarks. These unobtrusive imprints in selected blind colors enrich the native document with some kind of meta-data, which in turn is the result of a layout- and textual analysis of the document image that is performed before printing.

![Figure 1. Transformed Printout](image)

The types of HyperMarks provided and the associated kinds of interaction possible as well as the respective semantics are strongly depending on the use cases and the domain of documents for which the service is implemented. Figure 1 shows one page of a scientific paper with some added HyperMarks. Theses are for instance: title, author and affiliation as well as two headings – each of these elements are surrounded by their respective bounding box and also carry a small checkbox for user input. The example also captures other types of HyperMarks just expecting cross markers used to initiate further print actions after scanning the user feedback again. In general, user-annotation allows to confirm/correct/add labels or to ask for further information based on the inherent semantics of a HyperMark. The added value to an overall system will be respective structured meta-data in a repository, which allows a retrieval of documents via title, author etc.

However, the semantics of HyperMarks depends on the kind of information that can be extracted from documents. And the use cases must take into account in which situations users typically prefer reading and working with paper documents as compared to the PC.

Before describing two exemplary user scenarios in different domains, we like to proceed with the overall system architecture, independent of the domain and implemented use cases.

### 3. The Basic System Architecture

HyperPrinting provides its services when printing and scanning and thus it is obvious, that we require correlated components that are invoked in these situations. They are called print module and scan module.

Furthermore, the scan module needs to have knowledge about the contents of each printed page. Therefore all printed information needs to be stored in a persistency layer, which we call the repository.

![Figure 2. Process Cycle](image)
3.1. Print Module

The print module appears to the user as a virtual printer device and is thus accessible for all applications that can print – alternatively it can be activated by scanning a native document. This module processes the raw document without user annotations and has thus access to the most native data. In case of input from the printer driver, there will even be no noise such as salt & pepper or skew. Running the OCR here not only achieves most accurate data but also provides the full text representation to the following processing steps.

Therefore, the print module transforms the input document into PDF as canonical format and feeds this representation to the OCR [6]. The vector space model based retrieval engine DynaQ [7] checks whether a duplicate of the present document is already stored in the repository. If this is not the case, the PDF version will be stored in the repository to allow the creation of exact copies of the original document upon retrieval. Also the most relevant OCR results such as the positions and contents of all word segments will be stored. The system will then also create a unique ID for the document.

Figure 3 shows the architecture of the print module with its components and the control flow inside. The layout- and textual analysis are shown as generic modules as they depend on the use case. All the analysis hypotheses will be saved as meta-data in the repository.

The user interaction widget allows canceling the real printing. This way the document will only be stored in the repository.

If the document will be printed, all use case specific HyperMarks need to be overlaid to the original PDF. This is done by using an application-depending fix set of blind colors all of which signal different options for interaction. Besides these, the system also generates a unique page ID which is composed of the document ID, the page number, and a user ID. This page ID will be encoded in a 2D-barcode [8] which in turn is printed to the top right corner of each page. The other three corners will carry small hair cross markers which are required for normalizing the scanned image to the coordinates of the original bitmap.

3.2. Scan Module

The HyperPrinting prototype has been developed for the Ricoh series of MFPs (branded as Nashuatec). These machines are characterized by an API that allows the customizing of the control panel. Besides the buttons for printing, scanning or copying, there is a new button on the touch sensitive display indicated as “HyperPrinting”. Selecting it will allow scanning the pages of the document and piping the resulting scan images to our scan module.
User-ID. Doc-ID and page-number determine the page-ID. The hair crosses can easily be spotted in the remaining corners.

The unique page-ID grants access to the repository to retrieve all information related to that page: At first, the set-values of the hair cross coordinates are compared with the actual values on the scan image and allow to define the transformation function that maps the set values of e.g. word bounding boxes as defined in the repository to the scan image. The associated component is called the \textit{scan image normalizer}.

The following steps are more or less use case specific. However, they can be classified in two groups:

- Predefined HyperMarks that are generated by the print module and whose positions are stored in the repository. Here, the input areas for the user are well defined and can be directly accessed on the scanned image. The user input itself can be manifold, ranging from simple crosses to characters, digits or even words; again depending on the use case.

- User annotations to be entered apart from predefined areas. This could for instance be the underlining of some words or brackets around them. Here it is harder to spot the user annotations as the positions cannot be read from the repository data. However, with the use of a robust bitmap comparison \cite{9} original and scanned image can be analyzed. The result is a subtracted bitmap, showing all user strokes only.

At this point in time it becomes clear, why we perform an OCR on the generated PDF before printing and then store the bounding box positions of word segments: This way we can identify the word that has e.g. been underlined or placed in brackets etc. However, the interpretation of user annotations that consist of multiple strokes – so called symbol recognition \cite{10} – is not yet integrated to the HyperPrinting system.

In the following sections we like to explain some domain specific use cases, giving an idea of possible applications and their impact.

4. Research Paper Domain

As already mentioned at the end of section 2, the definition of practical use cases depends on the information that can be extracted at all and the actions that can be triggered on this information. Furthermore, working with the printed documents should typically be more attractive to the user than working with the computer in the selected domain of documents and tasks.

By nature, researchers have a close affinity to scientific papers and thus it is straightforward, that we defined useful scenarios for HyperPrinting in this domain. Figure 5 shows a sample output of a research paper cover page. In the following we will have a closer look at this example and explain the semantics of the correlated use cases and the possible actions.
matched against the associated meta-data that is stored for each document in the repository (and possibly also verified; see above). Only for entries that can be found, an according HyperMark will be generated.

Finally, it will be possible to underline keywords, thus processing user annotations apart the generated HyperMarks. The associated data will again end up as meta-data for this document, allowing a keyword-based retrieval.

**To sum up:** The described use-cases constitute a repository for scientific papers that builds up en-passant by just printing the respective documents using our virtual printer driver, the print module. Various mature document analysis technologies are facilitating this application. Regardless of any given user feedback, this system represents a full text searchable document repository that grows and fills up by just printing documents. And this holds for all documents, not only those, for which specific analysis steps are provided, namely research papers.

Two types of user input are possible: the first is to verify hypotheses and to spot keywords, thus giving no direct benefit to the user but boosting the meta-data quality of the repository for the sake of better retrieval. However it is of minimal intrusion to his regular work.

The second type of input allows a convenient retrieval of related publications by either marking of references or searching similar documents based on title, author or abstract.

A straightforward extension to this concept is a user front-end for direct retrieval on the repository rather than just the retrieval via checked HyperMarks. A not yet solved task is the correction of false system hypothesis, e.g. the false segmentation of the abstract or a bibliography entry as seen in Figure 5 or a false label.

5. Patent Application Domain

When conducting the described research, we needed to realize a demonstrator in too short a time to consider all the use cases that we described in section 4. Thus, another scenario with lesser use cases but also significant impact to the work of R&D staff has been selected for the first prototypical implementation.

5.1. Motivation

Most companies that are conducting research will typically claim for patents to protect their intellectual property. Since the associated legal process is quite cost intensive, developers themselves search for related and possibly conflicting patent applications in relevant databases in order to estimate the chances for the new ideas to be granted as patent prior to filing the patent.

In the assumptions that motivated our scenario the person in charge of these preliminary investigations will typically query a database with up to date granted and claimed patents. Depending on the topic, he might retrieve up to 200 hits. But for a first evaluation, only fractions of the documents need to be investigated. These are for instance the abstract, or the “what is claimed” section. However, patent applications might sometimes be up to 50 pages long. It is thus inappropriate to print all those documents for later manual screening. Only pages with the relevant sections will be printed at first. The necessary manual selections for this task are simple but time consuming and inappropriate for the person that is in charge of it.

Next, the user will start with his paper based work. He will thereby be able to decide for most documents, whether the contained ideas are conflicting with the own ones or not. But in some cases, a closer look at the whole patent application turns out to be indispensable. The user somehow needs to remember those cases and print the related documents in whole. Here again, simple but time consuming work is to be done.

5.2. Applying HyperPrinting Assistance

Our prototype has been designed to assist in both cumbersome tasks: First, it prints a generated report which consists of those relevant sections only, which have been defined by the user for the initial evaluation. Second, it prints the full length applications for those patents, which the user judges as doubtful and thus require a closer investigation.

To do so, the original concept of the HyperPrinting print module was modified for the sake of integration with the preceding retrieval engine. The results of that retrieval engine are fed into the new print module.

Here again, a structural analysis of the documents is performed: The system does the segmentation and labeling of text blocks and headers. As patent applications have a very common structure, the following logical objects can be identified: Publication Number; Title; Publication Date; Inventor; Abstract; Field of the Invention; Summary of the Invention; What is claimed.

But the main purpose of the labeling here (in contrast to the research paper domain) is to isolate these elements as user-selectable regions of interest (ROI). Consequently, the system pops up with a widget on which the user can select those ROIs that he wants to be printed in the report. Based on that input, Hyper-Printing then synthesizes a report on-the-fly. A sample page of such a report can be seen in Figure 1. Here the use case independent artifacts 2D-barcode and the hair crosses are visible. The selected ROIs in this example are Publication Number and Title (both mandatory) as
well as Abstract and Summary of the Invention. To the left of the title, the only type of interaction area is visible: it allows the user to mark doubtful patents. By simply scanning any page that contains some user markup, the system will immediately print the related patent applications.

The layout of the report might be somewhat confusing and shall thus be explained: The generated report should under no circumstances contain any recognition errors. Drawings and mathematical formulae should be reproduced in their native way. In order to do so, we decided that the report shall be composed of image fragments of the original patent documents. As patents have a 2-column layout and the report is an aggregation of the originals blocks, these blocks have the same width as if we had a two column layout.

Although this report layout does not allocate space of the print page in an economic way, we did not focus on optimizing it yet, because report synthesis was not in primary scope of the project. However, we are aware of strategies to produce more economic report layout as e.g. seen in Figure 7.

Whereas the direct benefit of the described use cases is the compilation of the report and printing of doubtful patent documents – both with extremely reduced user effort, this scenario implicitly provides another assistance: The 2D-barcode encodes not only a unique page identifier for each dynamically created report, it also encodes the user-ID of the person using the system. It is thus able to log the event of requesting a full length version of the patent document. This matter of fact can implicitly be used to assume some expert knowledge in the topics of the given patent application.

This additional information will be printed to the report and thus provides a link to a person within the organization that has knowledge about the respective patent. Rather than e.g. reading that patent document, the user may also contact the linked person. While required information about that patent can probably be acquired faster, this scenario to some degree assists in community building as people get notice of others (not necessarily from within the same department) working on a similar topic.

To sum up: While this scenario implements only one markup related use case, it still represents a proof of concept for the accurate spotting of HyperMarks and the test for user input inside the markers – regardless of any skew or displacement. Also the precise identification of a previously printed page could be demonstrated. Additional types of HyperMarks with different semantics can be realized straightforwardly.

While in the research paper domain a repository was built up that can be accessed by full text or structured queries, here the repository is only required to remember the context of the search when a report-page is scanned again. Nonetheless, its application opens up a huge potential for cutting down the time required for selecting and printing regions of interest and finding and printing doubtful patent applications in full length.

The side-effect of community building can foster synergies, specifically in large organizations with several hundreds or more employees.

6. Future Applications of the HyperPrinting Technology

The research described herein has to be considered as just the beginning of a completely new paradigm of user interaction by means of pen and paper. It requires no special hardware such as e.g. a digital pen, but only
well established devices such as a printer and scanner (or ideally a combined system like the MFPs).

Thus, the full potential of this new interaction paradigm could not yet be implemented. Many existing technologies like offline script recognition or gesture recognition can furthermore boost the possibilities. The following figures show examples for more possible scenarios that we can think of.

Figure 8 shows a newspaper article in which numerous words have been marked with bounding boxes of different color. The words are defined in a personal profile and can thus be typed (as e.g. being a person, a city, a product), indicated by different colors. The semantics of the different colors are explained in a legend that is printed on the bottom margin. In addition, document specific keywords can be spotted with appropriate IR technologies [7].

In this scenario, the analysis may be initiated by scanning the native news article into the system. As there exists neither a 2D-barcode nor hair crosses on the scanned page, HyperPrinting recognizes the data as native document and will thus analyze it with the print module, just as if that document was printed. The respective analysis will then generate the HyperMarks and print the page as seen above. The markup guides the reader to the important passages (based on the personal keywords) allowing a quicker reception of the core message.

Keywords that were not yet defined in the personal profile but are of interest to the user can be marked in the text by e.g. underlining them or putting brackets around. These words may then be marked with a unique number beside the textblock. A dedicated input field on the right margin allows the user to manually type those new words or concepts as e.g. the name of a city, person or event. Thus, the personal profile will be growing with each new document that is processed.

Another scenario is shown in Figure 9. The most important difference here: The scanned document that contains the user annotations, does not show any of the typical HyperPrinting artifacts like 2D-barcode or hair crosses. This is the case because the user made the annotations on the original newspaper page already. Thus, no comparison of the scanned image with some previously printed page can be applied and we therefore require pen-color based technologies for identifying the user markup.

Once this user markup is identified, HyperPrinting will identify vertical user-strokes and pair those of similar height and similar y-positions. From the applications point of view, the user will spot regions on the page with those strokes. He can also mark personal notes which may be written to the margin with those strokes. The application will compose an e-Mail that contains the bitmap fractions of all the so marked areas. The recipient can be selected in the menu of the scanner.

The described scenario allows marking and sending of news articles including some handwritten notes to e.g. a secretary without further processing of the scanned image on a computer by the sender.

If we might think of MFPs as terminals in public areas (e.g. airport lounges, post offices) one might use this service from anywhere in the world. In order to specify the processing mode or recipient, we could
think of some encoding in reusable 2D-barcodes as Post-It like labels which the user may stick to the page. As the system works on the original scanned image bitmap, no errors as they might occur when working on OCR text are possible. Also, all images and graphics that are found within those areas will be preserved in the generated mail, as the data transfer is based on the image.

7. Summary and Conclusion

We described a system for the lossless interpretation of user annotations on printed documents. As side effect, the system builds up a full text searchable document repository by simply printing any document to a dedicated virtual printer. Arbitrary document analysis technologies (such as e.g. logical labeling, keyword-spotting) can be applied to the documents to enrich the repository with associated meta-information. Based on these meta-data (but also the full text), several use cases can be developed that are related to a well defined type of user interaction on the printed document. These user interactions can be done within dedicated marked areas (HyperMarks), which are generated as a new layer on the original print image.

This technology may find interesting use cases in all domains, in which users are predominantly working on the printed documents – that means, they not only read the document but also leave comments and annotations. By means of HyperPrinting, these annotations will become accessible and interpretable to the computer with no extra manual work than just scanning of the pages.

The project under which this research has been conducted was focused on the prototypical development of the described technology. Observation and evaluation of pilot users as proof for the effectiveness of the approach are the consequent next steps but have not been started yet. The paper at hand thus has to stay limited to the description of the overall idea and the system architecture with its integrated components as well as sample user scenarios.

8. Acknowledgements

We like to thank the Ricoh Software Research and Development Group (SRDG) for funding major parts of the presented research.

9. References


