Augmenting Recommender Systems by Embedding Interfaces into Office Practices

Antonietta Grasso, Jean-Luc Meunier, Christopher Thompson
Xerox Research Centre Europe
Grenoble, France
{grasso, meunier, thompson}@xrce.xerox.com

Abstract

Automated collaborative filtering systems collect evaluations from users of the quality and relevance of stored information items, such as scientific papers, books, and movies. A number of users need to give evaluations for the systems to be able to produce statistically high quality predictions of an item's interest. Promoting the creation of a rich meta-layer of evaluations is essential for these systems, but several important issues remain to be resolved.

The work presented here first analyses the issues around the collection of recommendations, then proposes a set of design principles for improving and automating the collection of recommendations, and finally presents how these principles have been implemented in a real usage setting.

1. Systems to alleviate the information overload

'Information overload' may be an abused term but it is an increasingly apt description of our current experience in dealing with information. The increase in communication channels and publishing means has generated a myriad of new possibilities in the way to interact for both work and leisure activities. It has also generated an increased difficulty to make informed choices about what is/is not relevant and good quality information. To answer these user needs in a partly automated way several techniques have emerged, based on models of what is, and what is not interesting to a user. Some of these techniques model the user in isolation and consider his or her information space as the only context for evaluation of the relevance of documents. The work done in the fields of *Information Retrieval* and *Information Filtering* goes under this broad category.

Overall both *Information Retrieval* and *Information Filtering* are effective at identifying documents the content of which matches an interest pattern or a domain; they are much less effective in terms of evaluating the quality of a piece of information.

These quality evaluations are usually well performed by humans and the social process supporting the sharing of this knowledge is referenced to as "word of mouth". A broad category of systems and techniques aims at exploiting the "word of mouth" process and go under the name of *Collaborative Filtering* [7]. These systems filter information on the basis of the opinions of other users. They can be based on explicit actions of users (e.g. a document annotation sent to user Dave and Chris that something is relevant). In this case the system goes under the category of *active collaborative filtering*. However, as noted in [7], when communities of users are large (e.g. distributed organizations or Internet newsgroups), it is not possible to rely on the fact the each person knows the others and their interests. To address these needs and settings, several statistical techniques have been developed which identify groups of users, which share the same opinions and tastes in certain domains. These systems instead of relying on an active user specification of who should read what, attempt to automate the process of filtering the most relevant information on the basis of rating, and numerical expression of user preference. The systems using these techniques are therefore called *automated collaborative filtering*.

2. Automated collaborative filtering

The active collaborative filtering approach is based on the assumption that if users have agreed in the past, then they are likely to agree in the (near) future. The algorithm operates by finding users that have rated a number of items in common and by using those ratings to compute a correlation between the two users. The algorithm in Figure 1 is the one used in the GroupLens system [6] and describes the *Pearson Algorithm* to compute the correlation among two users, X and Y. X_i represents the rating of user X of item i. X -bar represents the average of all X's ratings. The algorithm provides values ranging from -1 (X and Y tend to disagree at all on the items they both rated), 0 (X and Y rating are not correlated) and 1 (X and Y tend to agree perfectly). It has to be noticed that the only items taken into account for these computations are only the ones X and Y both rated.

$$r_{XY} = \frac{Cov(X,Y)}{\sigma_X \sigma_y}$$

$$r_{XY} = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_i \left(X_i - \overline{X}\right)^2} \sqrt{\sum_i \left(Y_i - \overline{Y}\right)^2}}$$

Figure 1: The Pearson Algorithm

Once a correlation has been obtained between users, it is then possible to predict scores for items based on the ratings of other people for that item provided that they correlate.

Figure 2 shows the part of algorithm described by [6] to predict the score for an item based on the ratings of other users with whom the user under consideration correlates. The algorithm provides a weighted average of all the ratings on that item with the weights being the correlations with the user for whom the score is being produced. Typically this algorithm is modified to only include correlations above a specified threshold - for example it would be possible to decide to use only positive correlations between users when applying this algorithm.

$$Z_{i_{pred}} = \overline{Z} + \frac{\sum_{W \varepsilon reviewers} \left(W_i - \overline{W}\right) r_{ZW}}{\sum_W |r_{ZW}|}$$

Figure 2: Predicting scores for a document based on other users' ratings

The approach presented here, based on identification of *neighbors* of user who tend to share the same opinion on subjects, suffers of some problems which have been summarized in [7]: the *early-rater* problem and the *sparsity* problem. The early rater problem [1], refer to the inner paradox that the better an automated collaborative filtering system works, the more it would be beneficial to wait for other neighbors to produce evaluations on items. Basically the early rater problem points to the question: why a person should produce the meta information about how good or bad an information item is.

Related to the early rater problem, the sparsity problem, as described in [7], is about:

> "*In high-quantity, low-quality environments, such as the Usenet news, users may cover only a tiny percentage of documents available [...]. On the one hand, this sparsity is the motivation behind filtering: most people do not want to read most available information. On the other hand sparsity poses a computational challenge as it becomes harder to find neighbors and harder to recommend documents since few people have rated most of them.*"

In other words, automated collaborative filtering based on statistical techniques as the one described above, need to rely on very dense matrix in order to produce good predictions.

In order to overcome these problems several approaches have been proposed, some of them focusing on obtaining the most from a given set of documents and ratings and some of them focusing on obtaining a bigger set of ratings and comments. We call the first one, the *algorithmic* approach and the second one, the *ergonomic* approach.

2.1. Some design principles: The algorithmic approach

The algorithmic approach aims at getting the best result from a set of documents, users and ratings, i.e. the idea is to have smarter and more complex ways to produce the evaluations. In the following we list the most used approaches so far.

Partitioning. This approach is based on the consideration that the information space can be divided into domains and that users are likely to be interested in only some of them. By considering each domain separately, the matrix (reviewed documents × users) tends to become more dense, and the confidence measure associated to the prediction increases. Such approach is used, for example in the Knowledge Pump [3, 4] and in the GroupLens [6] systems.

Hybrid systems. In hybrid systems, the collaborative approach based on neighbor identification is complemented by content-based analysis. This approach leverages the fact that content analysis can be performed on every document, independently of the number of ratings available. In the Fab system [2], for example, the neighbors of an user are created on the basis of linguistic profiles derived from the feedback given on documents. Another example is given by the filterbots agents in Grouplens [7], where simple linguistic agents learn user preferences with respect to the length, language accuracy and number of replies in the message (the GroupLens context are Usenet newsgroups).

2.2. Some design principles: The ergonomic approach

The ergonomic approach aims at supporting the low cost provision of recommendations, ratings and comments by users, and to make the recommendation activity more directly and clearly beneficial to the user. This approach is based on the consideration that the provision of recommendations can be considered as a common good [3, 4, 5, 14], likely to be happily consumed, but unlikely to be happily maintained and nursed. Some ergonomic approaches to alleviate the common good problem are presented below.

Immediate perceived benefit. The act of recommending can be considered either as being primarily an altruistic activity, or as an activity that is done in the first place for own purposes, and that is then published to other people. This change of perspective - putting first the personal purpose aspect and then the altruistic purpose - can have a strong impact on the number of recommendation made available. In the following we review some existing approaches which go in this direction.

- Better ranking when doing searches. The relevance feedback in Information Retrieval is a technique, which elaborates the feedback given by a user on past information items to produce an automatic score on new incoming information. The provision of dynamic scoring mechanisms derived from recommendation rankings is then an additional benefit in making the recommendations available. The Soap system provides an environment where the search function is integrated with a recommender function [12].
- Market model. Some authors have stressed that if the recommender system is a "common good", then the most effective way to maintain it is by market mechanisms. In the Knowledge Pump system [3, 4] some simple "economic" rules have been put in place: for each visit to a recommended item, the visitor pays one chit, which is redistributed as royalties shared among the item's reviewers.
- Personal storage of information. The most obvious personal benefit is to link the recommender to personal activities like bookmarking and annotating. Example of this has been implemented in the Knowledge Pump system [3, 4], where the recommender is built on top of a bookmark facility.

Integration with work practices. The idea is to minimize the cost of providing recommendations as much as possible, by integrating the function in the practices where the information is consumed.

- The Knowledge Pump project has worked on the usability issue in the work place in many ways. One of its proposed approaches related to integration with practices is the possibility to embed the recommending function in the repositories of information. In this way the act of storing or retrieving an information item can be complemented at low cost by the possibility to rank and comment it.
- The seamless integration of the recommender function into selected points or tools that are part of document based practices, decreases the cost of providing recommendation for the users. It decreases because the possibility to annotate, rank and recommend is provided where the evaluation and enrichment happens. In the example of the coversheet discussed later, the recommendation possibility is associated to the act of printing. This is based on two assumptions: first that, very often, an evaluation can be given only after having read the document; and second that annotations are then often made by writing on the document or an associated sheet of paper.
- Information items are not always represented by electronic documents. Information is available in many other ways in everyday life: printed information, videos, recorded music and so on. By making more pervasive the possibility of recommending or annotating documents, users can access the system in the course of their actions and therefore in the proper context.

3. Paper based interfaces for collaborative filtering systems

Having reviewed both the algorithmic and ergonomic approaches, we believe that it is in combination that they can provide real advantages. In this paper, however, we concentrate on ergonomic approaches to overcome problems with automated collaborative filtering systems. More concretely, we will propose the usage of paper based user interfaces [9, 13].

3.1. Paper usage in the office environment

Despite the hype about the 'paperless office', important documents are usually printed at least once during their lifetime. This is due to the various affordances paper can provide over documents read on screen [9, 11], and includes easier reading, annotation and very often the process of sharing the document with colleagues.

This means the print room (or shared printer) of a group often provides a point where most of the information of the group, embodied in their printed documents, passes by. Where a printer is shared, the documents are often organized by means of 'cover sheets', on which the name of the owner of the document is printed together with usual other information such as a time stamp. Such cover sheets are discarded, in the best case recycled. The process of printing is without memory except, perhaps, for the print logs generated at the printer server which are only used for administrative or accounting purposes. Our observation of several aspects of the print process, the print cover sheet and the way printed documents are used, have enabled our proposal of several new print-based services for knowledge sharing. These services better manage and share the knowledge embodied in printed documents both at the personal and at the community level.

The relevant observations of the print process are:

- Choosing to print a document makes an implicit declaration that it has some relevance/importance to the person doing the printing. This declaration is preliminary; reading the document may either result in an increased evaluation, or its disposal.
- There's a reasonable probability the document is also relevant/important to other people in their group.
- Someone else in the group may have printed it before; it could be useful to be aware who they are and perhaps have their opinion.

Cover sheet aspects are:

- The cover sheet usually has a lot of free space and provides a convenient output space.
- The cover sheet is created together with a document but this relationship is not used beyond identifying the owner.

Printed document usage aspects are:

- Filing of printed documents is usually done some time after printing; this de-coupling of the actions can lead to problems in retrieving the document later on.

3.2. The Smart CoverSheet recommender interface

On the basis of the above observations of print process we constructed a new, paper user interface to the Knowledge Pump system. Document cover sheets, associated with every printed document, are used to both collect and provide contextual recommendations and as interfaces to scanned document storage and retrieval systems.

The functionality of the current Smart CoverSheets can be split in four categories:

- **Document token**. Each time a document is printed through the Smart CoverSheet application, an active coversheet is produced (see Figure 3). Its primary function is to provide access to the document via the Web by encoding the document URL in a printed DataGlyphs. It can also be used to store and subsequently retrieve the document from a document management system.

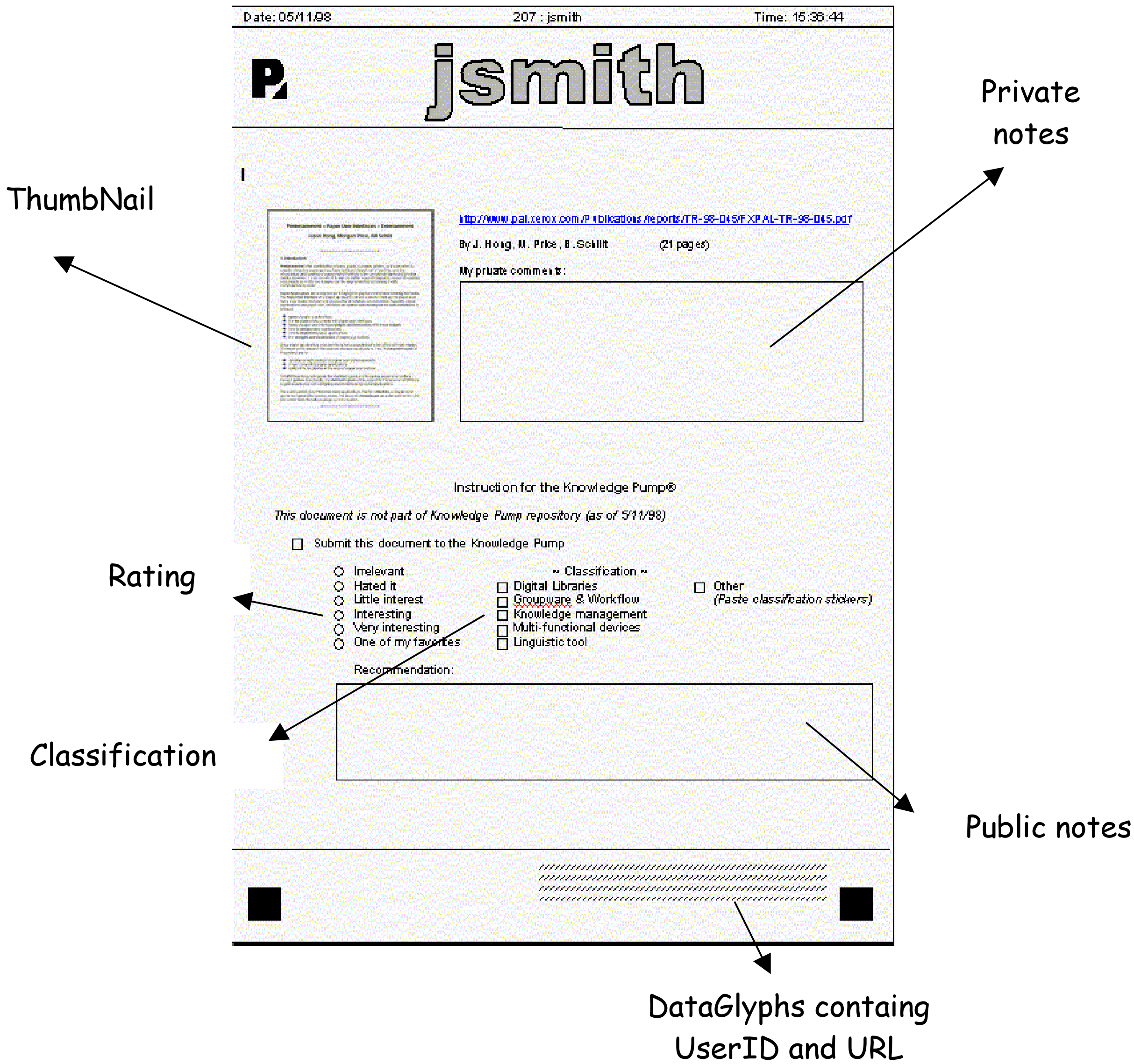

Figure 3: The Smart CoverSheet

- **Public recommendation and notes**. The cover sheet can be used to provide the means to collect contextual recommendations both when the document is printed or later, having read the document. This coupling leverages the observation that printed documents are likely to be of wider interest and allows the user to make the recommendation when a better judgment has been formed. Given that recommender systems usually provide large classification schema (communities of interest) where to submit the document, the cover sheet can be augmented with user interface stickers to specify application specific additional commands not directly available on the cover sheet, like the complete list of communities of interest available. These stickers could be available at the multi-function device (MFD for short).
- **Personal notes**. An alternative to the provision of public review is to use the coversheet to store in the recommender the personal notes and evaluation put on it when reading the document.
- **Contextualized list of reviews from other KP users**. Each print request activates a search in the recommender system to discover if users of the system have already evaluated the same document. If that is the case a list of reviews is printed on the back of the cover sheet with ratings and comments from other users (see Figure 4).

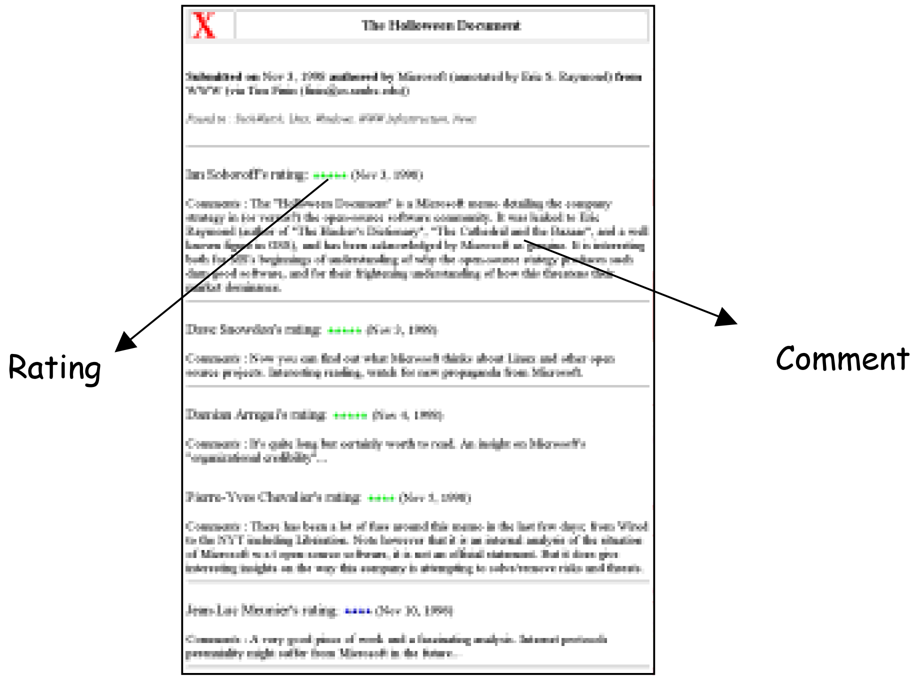

Figure 4: The backside of the coversheet containing the reviews

Immediate perceived benefit: As we described earlier on, a way to get more recommendations is to connect their provision to personal functions. In the Smart CoverSheet both the document token and the storage of personal notes are functions that have a personal utility independently from the publishing of the notes and comments to the community. Moreover the possibility to take and store the notes on a sheet of paper that can be reused to get the document from the Web, fits with the observation of why and how paper is still so much in use in office environments. Finally the royalties mechanism of KP still applies when submitting reviews through the Smart CoverSheet.

Integration with work practices: From the point of view of integrating the recommendation activity more in the work practices, the Smart CoverSheet supports provision of recommendation from the print moment later on. This fits when the observation that most of the reading activities are still carried on paper printed documents [12]. The availability of the coversheet during the reading activity is meant to support notes taking to be reused later on, when accessing to the document again.

4. Implementing paper based user interfaces

Producing recommender system aware coversheets requires three main functions:

1. The ability to recognize the document that is being printed;
2. The ability to query the recommender system for reviews and to equip the coversheet with machine-readable instructions so that they can be automatically processed later;
3. The ability to trigger computation steps when the user scans in an annotated coversheet.

We will discuss below the implementation of these three main parts, grayed out in Figure 5.

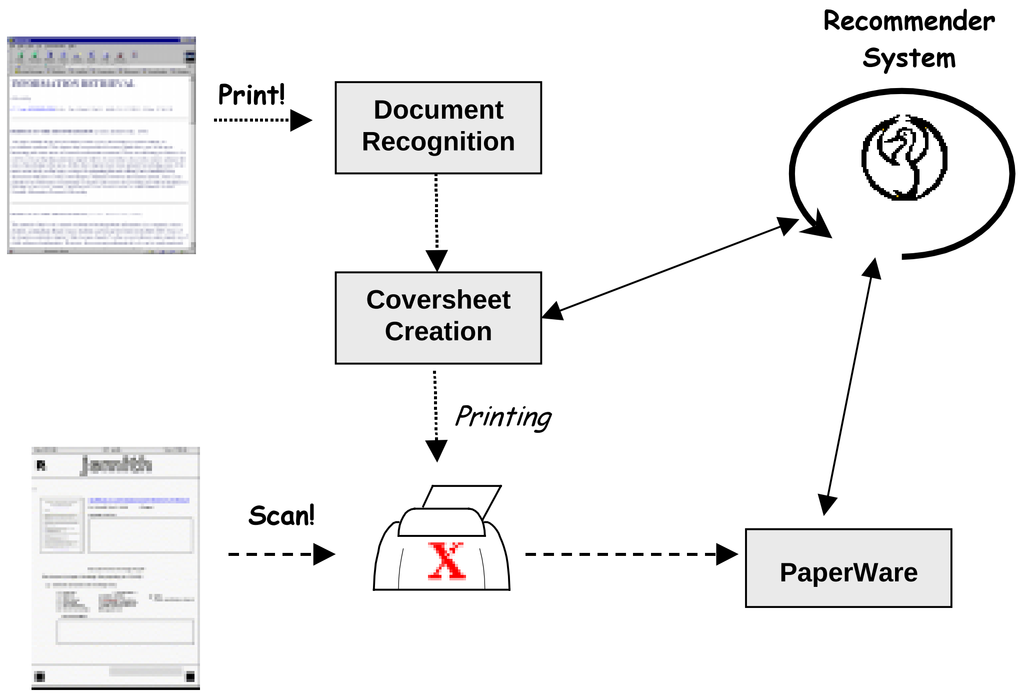

Figure 5: System architecture

Document recognition: A print job consists in a postscript representation of the document sent to a printer. To produce the appropriate coversheet the system needs the document reference, which in our case is the url. For the sake of simplicity, we have chosen to rule out document content analysis. Making the assumption that the url appears either in the header or in the footer of the document pages, we extract it from the PostScript representation of the document. While seeming reductive, this approach is satisfactory in the sense that it works with the two major Web browsers on the most used platform.

Coversheet creation: While querying the recommender system is quite straightforward, the way we equip the coversheets to permit their processing is discussed below. The solution we are using is based on several technologies from Xerox, namely SmartPaper(tm), DataGlyphs(tm) and the PaperWare(r) environment [13].

To enable automatic processing of paper forms, the elements on the form have to be described electronically, and this description has to be available for the processing software.

Using elements from the Xerox SmartPaper toolkit and from the Campiello project [8], we took advantage of an XML based language for specifying the layout of forms. In addition to the standard elements for placing and formatting text and images (taken from HTML) there are tags for inserting active feedback elements into the form. Currently, feedback elements can be single checkboxes, sets of checkboxes and scan areas, i.e. rectangles whose content is captured as bitmap for storing or further processing.

With all the interaction elements, you can specify in the form description itself an action that should be performed when the form is scanned in later. Examples for such actions are registering feedback or storing comments.

This approach gives enough flexibility to account in the future for user preferences concerning the coversheet aspect and content.

The table in Figure 6 shows descriptions of the (main) extra tags for building paper forms.

<RATING>	Displays a set of checkboxes allowing the user to make a rating on a specified numeric scale Parameters ▪ Which item to rate, ▪ In which context is the rating valid; ▪ Command to be executed when one of the checkboxes is selected; ▪ The number of checkboxes that should be displayed and the numeric values associated with each checkbox.
<COMMENT>	Specifies a clip area which will be returned to the application Parameters ▪ Which item the comment should be associated with; ▪ In which context; ▪ The size of the clip area; ▪ Where the clip area should be located.
<FEEDBACK>	Creates a checkbox in front of or after the tagged text and associates an action with the checkbox Parameters ▪ Command to be executed when one of the checkboxes is selected.

Figure 6: Description of the Campiello language tags

Scanning an annotated coversheet: Binary information is needed not only to store information about the form itself (form identifier, additional information for the processing module), but also to record the user identity as well as the concerned document, i.e. its url.

Since we do not have invisible ink in today's printers we have to print the required data with standard methods but in a non-intrusive way.

There are several technologies for storing binary information on paper. The best known among them is Barcodes. We are using Xerox DataGlyphs(tm) [9], which have some advantage to Barcodes:

- higher density
- can have any form
- can be hidden in images

The DataGlyphs appear on Figure 3 under the form of a grayed rectangular area at the bottom right part of the coversheet. Black squares at the bottom and the upper left capital P are positioning marks. Equipped in this way, the coversheet materializes a function call to the recommender system.

5. Conclusion

In this paper we presented some new approaches to overcome identified issues in automated collaborative filtering. We concentrated on how the use of paper interfaces can help to integrate collaborative filtering based information system into the real practices and so improve the quality of their results.

Regarding the current status of our implementation work, it is too early to draw a conclusion. We have not yet deployed the Smart Coversheet interface in a real setting for several reasons:

- We intend to polish the interface and strongly test the system before any real usage. This is particularly important in our approach, precisely because it is so integrated in every day work practice, where even small problems may be felt as too disruptive. Permitting user acceptance is at this price.
- We also face a potential problem of limited space on the coversheet, and need to establish a strategy for filtering out information.
- Finally, as often quoted, a piece of paper won't run out of power. Smart Coversheets constitute the client part of the system, from a client-server perspective. But this sort of client cannot be upgraded easily and might last for years! The server software must correctly handle all Smart Coversheets ever produced. This puts severe reliability constraints on the server, requiring backward compatibility as well.

We intend to conduct field studies in the coming months.

6. References

[1] Avery, C. and Zeckhauser, R., "Recommender Systems for Evaluating Computer Messages", *Communication of ACM* 40, 3, 1997, pp. 88-89.

[2] M. Babalanovic & Yoav Shoham "Fab: Content-Based, Collaborative Recommendation", *Communication of ACM*, 40(3), 1997, pp. 66-72.

[3] N. Glance, D. Arregui, M. Dardenne "Knowledge Pump: Supporting the Flow and Use of Knowledge in Networked Organizations", U. Borghoff, R. Pareschi (eds), *Information Technology for Knowledge Management*, Springer Verlag, Berlin, 1998.

[4] N. Glance, D. Arregui, M. Dardenne "Making Recommender Systems Work for Organizations", *Proceedings of PAAM'99*, 1999.

[5] Hardin, G., "The tragedy of the Commons", *Science*, 162, 1968, pp.1243-1248.

[6] Resnick et al., 1994] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom & J. Riedl "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", *Proceeding of CSCW'94*, October 22-26, Chapel Hill, NC, 1994.

[7] Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., and Riedl, J., "Using Filtering Agents to Improve Prediction Quality in the Grouplens Research Collaborative Filtering System", *Proceedings of CSCW '98* (Seattle CA, November 1998), ACM Press, pp. 345-354.

[8] Grasso, A., Koch, M., and Snowdon, D. "Campiello – New user interface approaches for community networks", in *Proceedings of Workshop on Designing Across Borders: The Community Design of Community Networks*, Seattle, WA, D. Schuler (ed.), Nov. 1998

[9] Johnson, W., Card, S. K., Jellinek, H., Klotz, L., and Rao, R., "Bridging the paper and electronic worlds: The paper user interface", in *Proceedings of INTERCHI*, ACM Press, 1993.

[10] Resnick, P. and Varian, H. R., "Recommender Systems", *Commununication of ACM* 40, 3, 1997, pp. 56-60.

[11] Sellen, A. & Harper, R. "Paper as an Analytic Resource for the Design of New Technologies". *Proceedings CHI'97*, Atlanta, GA..

[12] Voss, A. and Kreiflets, T. "SOAP: Social Agents Providing People With Useful Information", *Proceedings of GROUP'97*, ACM Press, 1997, pp. 291-298.

[13] Xerox PaperWare® concept and multifunctional devices, http://www.xerox.com/paperware/, 1998.

[14] Ginsburg Mark, Kambil Ajit, "Annotate: A Web-based Knowledge Management Support System for Document Collections", *Proceedings of HICSS-32, 1999.*