4.1 Introduction
This paper outlines the development of an information system using collaborative filters. There are a number of resources related to metrology on the World Wide Web such as books, journals, articles or information published by experts in industry and academia. It is a challenge for the user to find best resources in a given context that fits his or her interests. Currently, the users have to spend a lot of effort to look for resources and spend time reading them to see if they meet their needs. One way to approach this problem is to use collaborative filtering technique. Collaborative filtering systems make use of reaction and opinion of people that have already seen a piece of information to make predictions about the value of that piece of information for people who have not yet seen it. By applying appropriate algorithms used to filter information, this system can guide the users to relevant resources or resources that have best quality in context.

4.2 Algorithm
At present, there is more information delivered than before. Engineers and scientists will be overwhelmed with the amount of information that they have to process everyday. The problem of information overload leaves the user unable to track information they want. In recent work, it has been shown that information can be filtered using several methods, such Kill files /Score files, Programmable Agents, or Intelligent Agent [1] using keyword search. The drawback of the keyword search is coarseness of information. For instance, some information contains keywords that are of interest to you, but their contents are not relevant. To resolve this problem, collaborative filtering technique must be used to filter or predict what information should be provided to the user. This technique tries to match the preferences of an audience to other audience in making recommendation. The filtering can be done by data mining [2], singular value decomposition (SVD) [3], or neighborhood-based methods [4]. A problem in prediction is spacing. If we form a matrix of users versus item to use prediction, we can have some empty cells in rows or column because someone will not enter feed back information after they check it out. To approach to this problem, SVD is used to reduce the dimension of the rating matrix to make more density of rating on data. This reduces the error in prediction due to sparsity of data. The following describes the SVD algorithm used to filter information [3].

SVD is a matrix factorization technique that factors matrix M into three matrices

\[ M = U \cdot S \cdot V' \tag{1} \]

where \( U \) and \( V \) are orthogonal matrices of size \( m \times r \) and \( n \times r \). \( S \) is diagonal matrix \( r \times r \), which contains all singular value of matrix \( M \).

\( S \), with rank \( r \) is reduced to \( S_k \), which contains \( k \) largest diagonal values. Reconstruction of \( M_k = U_k S_k V_k' \) can minimizes the Frobenius norm \( ||M - M_k|| \) over all rank-\( k \) matrices. \( M \) is a rating matrix.
matrix where its rows present the user elements and its columns present items that are rated. Following shows how to use SVD to predict rating scores.

1. Fill the null entries in the matrix by replacing each null entry with the column average of the corresponding column.
2. Normalize the matrix by replacing each entry $m_{i,j}$ by $(m_{i,j} - \bar{m}_i)$ where $\bar{m}_i$ is the row average of the $i^{th}$ row.
3. Decompose the matrix $M$ into $U$, $S$, an $V$ matrices
4. Reduce the $r \times r$ matrix $S$ to dimension $k$ by keeping $k$ largest diagonal values of the matrix $S$
5. Compute two resultant matrices $U_k S_k^{1/2}$ and $S_k^{1/2} V_k'$. This gives a prediction scores.
6. De-normalize the prediction scores by adding the user average back into each prediction score

SVD can be used to form neighbors and to generate top-N recommendations for the user. Following steps shows how to form neighbors in the reduced space and generate top-N recommendation.

1. Decompose the original user-product matrix $M$ into $S$, $U$, $V$ using SVD
2. Reduce the $r \times r$ matrix $S$ to dimension $k$ by keeping $k$ largest diagonal values of the matrix $S$
3. Reduce the matrix $U$, $V$ to dimension $k$
4. Compute the matrix product $U_k S_k^{1/2}$
5. Use Cosine similarity and the matrix product $U_k S_k^{1/2}$ to form the neighbors

After the neighbors are formed, the Top-N recommendation list can be found by two techniques, Center-based and Aggregate neighborhood [2].

Center-based: Top-N recommendation can be found by scanning through the rated documents of each of the $u$ neighbors and performing a frequency count on them. The rated document list is then sorted and the $N$ most frequently rated documents are returned as recommendations to the user [3].

Aggregate neighborhood: scheme forms a neighborhood of size $u$, for a customer $c$, by first picking the closest neighbor to $c$. Then the remaining $u-1$ neighbors are selected as follows. At a certain point, let there be $j$ neighbors in the neighborhood $N$, where $j < u$. The algorithm then computes the centroid of the neighborhood. The centroid of $N$ is defined as $\tilde{C}$ and is computed as $\tilde{C} = \frac{1}{j} \sum_{\forall v \in N} \tilde{v}$. The user $w$, such that $w \notin N$ is selected as the $j+1$-st neighbor only if, $w$ is closest to the centroid $\tilde{C}$. Then the centroid is recomputed for $j+1$ neighbors and the process continues until $|N|=u$ [3].

4.3 Features

The collaborative system is designed using object oriented programming technique. The main components of the system are the user interface classes and the background processing classes. All the classes of the system have been developed using Servlet. Component architecture of the system is shown in the figure below. Servlets connect to a database using Java Database Connectivity (JDBC) driver. They retrieve information from the database and input them to the
SVD algorithm to make a recommendation. Database is designed using relational databases and Oracle. Database stores information of documents being viewed and rated from the user, rating scores, and the user information.

The system consists of the following features:

- Provide a content management mechanism where a content provider can submit information to a central server. Content is reviewed by content editor before it can be published.
- Provide a surface finish catalog where all information is indexed in categories. The user can add links of their web site to the catalog or submit technical paper or surface finish data to a catalog.
- Provide a search engine where the user can search information from a database base on a categorized or general search.
- Run on Internet or Intranet.
- Provide a recommendation engine where it can guide the user top N list of relevant information used to analyze surface finish.
- View comments from other users. The user can rate and give comments on a particular document stored in a database.
- Provide the feature where the system can predict rating scores based on small and large size of rating data.
- Provide the user a unique interface that they can view, or rate information.

4.4 Interfaces

Figure 4.1 Component Architecture

Figure 4.2 Documents with their rating score is displayed on a web browser
Figure above (4.2) shows how documents and rating scores are displayed on Surface Finish Catalog Interface. This screen shows the documents under “basic filtering” category. On this screen, the user can see rating score, title, author, abstract of documents, and content provider information. To access a document in detail, the user clicks on a title and the screen shown in figure 4.3 is displayed. This screen contains three frames. On the left frame, the user can rate the document, view comments from other users and top 10 recommend documents, and give feedback. When the user clicks on links at the left frame, the output is displayed on the bottom right frame as shown below. At the top right frame, the user can view a document in detail. The document can be from a data file located from the web server or from other websites.

Figure 4.3 Rating document, top –N recommended document, and detail view of a documents screens

4.4 Conclusion
Collaborative filtering is one of the techniques used to filter information based on user references. Collaborative filtering is particularly useful in rating items that are hard to quantify such as ideas, feelings and thought. Also it can enhance information filtering systems by measuring, in dimensions beyond that of simple content, how well an item meets a user’s need or interest. Collaborative filtering system guides the users to relevant resources or resources that have best quality in context and in timely manner.

4.5 References