Technical White Paper

INFORMATION MANAGEMENT TECHNOLOGIES SUPPORTING KNOWLEDGE MANAGEMENT

Cicada Cube Pte Ltd

20 Martin Road, #05-01 (Suite 9)
Singapore 239070
Tel: (65) 67322238 Fax: (65) 67387880
info@cicadacube.com
www.CicadaCube.com

INFORMATION MANAGEMENT TECHNOLOGIES SUPPORTING KNOWLEDGE MANAGEMENT

ABSTRACT

The advent of the Internet about a decade ago has added a new dimension in information overload. The increase in information is inevitable and will continue to grow as time goes on. Information organization and management has become a major concern in today's knowledge-based economy. For example, how does one manage information residing in multiple, disparate data-sources, and how much do Information Technology Providers understand about users' information search behavior in the rendering of information to them? This paper discusses management and usage issues in the light of information overload from the Intranet and Internet and examine current and potential technologies that could provide solutions to the problem.

1. INTRODUCTION

The amount of information at one's disposal is no longer restricted to those on one's desktop computer; it extends far beyond. With computers networked and communicating at neck-breaking speed, the information pool now includes those in the Intranet and the Internet. Information Management has to be more effective if an organization has to address the larger issue of Knowledge Management. The solution has to deal with managing multiple disparate data-sources and enabling users to find what they are looking for effectively.

2. MULTIPLE DISPARATE DATA-SOURCES

2.1 Information

Information comes from data. Three numbers 2, 3, 5 may just be numbers but when they are put into a database table column, they may well be treated as "prime numbers" or the "age of 3 children" or the "number of sales-persons in a company" or even the "amount of money in a bank account". The information applied on these 3 numbers is commonly known as meta-data – data about data. New information can be derived from these 3 numbers by applying a function on them; e.g. multiplying the numbers with another number or a complex formula would produce another set of numbers.

From the example above, we can see that the pool of information available to a user is not limited to the actual original data (numbers) but also the meta-information and derived information that are associated with them. What this means is that users can be overwhelmed by the amount of data available to them and an effective means of searching for the right data is certainly essential to overcome the complexity information overload brings.

2.2 Central Repository

The traditional approach to managing information has been to store them into a central repository with a front-end client application to deliver the data/information to the users. An administrator takes charge of managing the information in the central repository and all other applications wanting to use the information in the repository have to conform to a certain standard established by the administrator. There is thus a central control over the usage of organizational information.

There are many advantages in having a central repository of information. Primary among the advantages is the means for a central control that enhances ease in data management. The use of a relational database is an example of an implementation of central repository. A Database Administrator takes charge of defining rules for establishing database schemas and regulating the usage of the data. Applications are then built on top of the database to provide the necessary business functionalities. Another example is the Lotus Notes group-ware where electronic documents are stored in Notes standard files similar in structure to relational databases. Other applications such as SAP and PeopleSoft are further examples in which data are stored in central repositories.

2.3 Multiple Disparate Repositories

Having one central repository in a large organization is usually not possible since the amount of organizational data is built up incrementally:

- Technical reason: Information resides in different format and media and may not be possible to house them into a single repository.
- Political reason: Information owners want to have more control over the shared data. Information therefore may not be placed into the same repository, leading to multiple repositories.
- Temporal reason: The amount of information in an organization grows incrementally with time. What had been designed earlier for the central repository may not be applicable to the new situation anymore. There is thus a need to create repositories to house new data/information.

Having multiple disparate repositories is an inevitable situation in any large organization. From desktop networked personal computers to large relational databases to group-ware products such as Lotus Notes and/or Microsoft Exchange, users are now faced with huge amount of available information spread across multiple repositories. It has been difficult for users to use them as they need to familiarize themselves to the various protocols and interfaces demanded of them from the various repositories. Unless users have an effective system for them to know what information they have and how to search for them in whatever format the information is represented, it will be a futile effort on the part of the users to appreciate what are available at their fingertips.

3. INFORMATION MANAGEMENT

3.1 Storage

Data in an organization is accumulated over time. While it is desirable for data to be centralized into a single repository, it is not practical to insist on such an approach. There are various reasons why this is not possible:

- Systems become obsolete and it would not be practical to keep insisting new data be kept in out-dated systems.
- People change and it is unlikely that the person who made the first proposal for the central repository will be around to insist on such a practice.
- New technologies demand the implementation of new systems and practices. It may not be possible to implement new system data into the existing central repository.

A case in mind is data warehousing where data are stored within a single system to take advantage of the functionalities built into the system. Many organizations that have implemented data-warehouses now have data stored in other application systems too. They are no more centralized into a single system.

It is therefore natural to find in an organization data stored in various application systems. Difficulties arise when locating them, let alone attempting to draw analysis out of the data in the case of data mining.

Managers should be aware that the amount of data in an organization would continue to grow in the course of the business cycle. More and more data will be generated and they are unlikely to reside in one repository but across multiple disparate data sources/repositories. Managing these data then becomes an issue.

3.2 Search and Retrieval

Classification of documents has been used traditionally to enhance the search and retrieval process of documents. Until automatic document classification becomes possible, the document search process will always be limited and the result of a search will continue to depend on the know-how of the users in the application of search strategies in information search.

Having large amount of data and information and looking for information without effective search and retrieval techniques is like finding a needle in a hay-stack. Studies on user behavior on information searches carried out using Library online catalogs have shown that users generally have problems finding information. For example, users have problems matching their search terms with those indexed in a data-source.

3.3 Search Expression

Besides deploying search strategies in helping users get the most out of an information database, search expressions such as the ubiquitous Boolean expression has been used to let users formulate their own search query. Unfortunately, users have difficulty using Boolean expressions to broaden or narrow their searches. Many users have also disregarded the Boolean expression provided by search engines and prefer the system to provide the results of their search automatically. In other words, a single-line search method with the users entering what they are looking for as search query terms is preferred.

As is evident in the Internet, more and more sites are offering such single-line search. However, the performance and results of the search depends largely on how the search engine indexes its terms and ranks its results.

3.4 Static Relevance Ranking

The ranking of hits found in a search is a very important aspect of a good search and retrieval solution. Some search engines, for commercial reason, treat their ranking as a form of advertising revenue generator. Companies can bid for their website to be listed in the result list by paying a premium. The more they pay the likelihood of being listed high in the result list for a given query is greater. Other search engines uses popularity as a means of determining which sites get to the top of the list. Another approach adopted by search engines is to use the number of links on a site as a means of determining the ranking of a site. The more sites that are linked to a site, the higher the ranking the site will get for a search term found in the site. Nevertheless, all these approaches do not apply well to non-Internet-website-related information commonly found in organizational data residing in databases.

The way search engines rank their results is based on a set of pre-determined criteria such as those highlighted above. By placing weights on sites or data, search engines are able to rank their results. Such search engines are said to exhibit ***static relevance ranking***. The advantage of such an approach is the increase in speed in delivering results for a search query. The main disadvantage is the biasness of the search results towards criteria that may not necessarily conform to what the users are looking for.

Static relevance ranking applies to situations where the database is available during indexing time. However, there are situations where this is not possible. For example, searching for information using meta-search engines. Another form of ranking is therefore required.

3.5 Dynamic Relevance Ranking

In a meta-search situation, search results from various search engines are retrieved at search time; unless they are re-ranked the set of result will not be very useful. Re-ranking may not be so straightforward as much of the critical information may not be available for the re-rank. This form of ranking is known as ***Dynamic Relevance Ranking*** since the ranking is done at search time (rather than indexing time).

What is the best approach to dynamically rank results when the ranking has to be done at search time?

Consider the following example: a user enters “Singapore Airlines” as the search query terms. Obviously, a “Singapore Airlines” site is more relevant to the search query than one that is an “Airlines of Singapore” site. Consider a separate query: a user enters “apply for a passport” query. A hit that has keywords “apply for a passport” would be more relevant to what the user is looking for than one that merely has these keywords but may not necessarily form the phrase “apply for a passport”.

Research has shown that users generally know what they are looking for and they express them through the keywords they enter in the search query box. A site will be considered as relevant if the site contains the keywords that had been entered by the user. Given this understanding of relevancy, search engines should therefore deliver hits that have terms that conform to those entered by the users as the topmost relevant hits. Given that every search query is a new query and that the keywords and the sequence in which the keywords may appear in search queries may differ, it would be difficult to pre-determine the weight of the ranking.

Figure 1: CicadaSearch "Singapore Airlines"

Figure 2: CicadaSearch "Singapore Airlines" subsequent page

Cicada Cube has defined an implementation of *Dynamic Relevance Ranking* whereby hits with keywords similar to the search query terms are ranked higher in the relevant list. The more similar the hit is to the search query terms the higher the ranking value. This technique of ranking search results is applied in CicadaSearch, a meta-search engine (developed by Cicada Cube Pte Ltd) that connects to 4 other Internet website search engines such as Excite, NorthernLight, AltaVista and AllTheWeb. CicadaSearch can be found at www.cicadasearch.com

To illustrate the application of **Dynamic Relevance Ranking**, a sample of the result of a search from CicadaSearch is shown in Figure 1. The number of cicadas indicates the relevance value of a hit with respect to the search query terms entered – in this case, "Singapore airlines". Figure 1 shows hits number 1 to 6 while Figure 2 shows hits number 23 to 28. Note that the number of cicadas decreases with the increase in the hit number suggesting that less relevant hits (with respect to the search query terms entered) are found lower in the result list. More relevant hits are displayed at the top of the list while the less relevant ones are displayed later. Hit number 26 is the website of the United Airlines in Singapore. Notice the keywords "Singapore" and "Airlines" are found in hit 26 but it is not ranked to the top because the words do not conform well to the user's entered query terms ("Singapore airlines" in this case). Sites with "Singapore Airlines" are ranked to the top of the list ensuring the more relevant hits are displayed to the user first.

Since users typically view the top 20 hits from a search result, a search engine that is able to deliver the hits that are most relevant to users' query terms to the top 20 of the result list would be able to render to users what they are looking for.

Figure 3: CicadaSearch "Monetary Authority of Singapore"

Figure 4: CicadaSearch "Monetary Authority of Singapore" subsequent page

As another example of the power of Dynamic Relevance Ranking in CicadaSearch, consider a search on "Monetary Authority of Singapore". Figure 3 shows hits number 1-6 of a search from CicadaSearch. Figure 4 shows hits number 35-39. It is clear from the above examples that Cicada Cube's

Dynamic Relevance Ranking does bring forth much utility and value for naïve users in looking for information that they need.

3.6 Unified Search Interface

Applying the above two concepts of Multiple Disparate Data-Source Search and Dynamic Relevance Ranking to a corporate context, the same advantage found in CicadaSearch can be realized in it too. Each of the above Internet search engine in CicadaSearch can be replaced by a corporate data-source. In a large corporation, it is quite natural to find multiple data-sources; these data-sources can be Intranet-based or Internet-based. A meta-search engine designed with a Unified Search Interface will be able to value-add to the entire search experience of the ordinary users.

4. ARCHITECTURE

It is clear from above that an effective search and retrieval solution ought to encompass two types of search engines: *native search engine* and *meta-search engine*. A native search engine is one created to search into the database of a data-source whereas a meta-search engine is one created to search into a data-source that already has a search engine. Further analysis can be done on the search results via a meta-search engine. The performance of a meta-search engine is dependent on the search facilities provided by the native search engine. However, it does not mean that the performance of the meta-search engine will be weak if the native search engine is weak. On the contrary, research carried out at the National University of Singapore shows that meta-search engines could be designed to search better than the native search engine of a data-source.

4.1 A mix of Native and Meta-Search

The proposed architecture for an effective and scalable solution for managing and using information in a multiple disparate data-source environment is made up of a mix of native search and meta-search. Figure 5 shows the proposed architecture. With this architecture, new data-sources can be added without affecting the other components in the architecture. As an example, Data-Source A could be a Lotus Notes data-source and Data-Source B could be an MS SQL Server database incorporated with a native search engine. To provide a unified search, Meta-Search Engine A is developed to encapsulate the Lotus Notes search engine. A Meta-Search Engine B can similarly be developed for Data-Source B. A master Meta-Search Engine X is then developed to provide a uniform search interface for all the search engines under its charge. With such an interface, future data-sources can be added as part of the unified search without much difficulty.

5. SUMMARY

The amount of data owned by an organization is never static. It grows with time. Putting all the data into a central repository can at best be temporary. Additional disparate repositories are often created leading an organization to have multiple disparate repositories/data-sources. Such a situation poses great difficulties from a data usage point of view since it would be rather difficult to search for information when they reside in multiple repositories.

This paper recognizes the problem and proposes a system architecture that is made up of a mix of native search and meta-search. In addition, the paper also proposes a form of ranking known as Dynamic Relevance Ranking that eliminates the need for using Boolean Expressions in a search query. The ranking technique does so by ranking search results dynamically (when the results are available during a search session) with respect to the sequence of search query terms entered by the users. The effectiveness of the ranking technique ensures that hits relevant with respect to user query terms are positioned at the top of the result list. In this way, even when the search returns a large number of hits, the most relevant hits (with respect to the user's query terms) are always available for viewing first. In other words, the overheads for a search are very low.

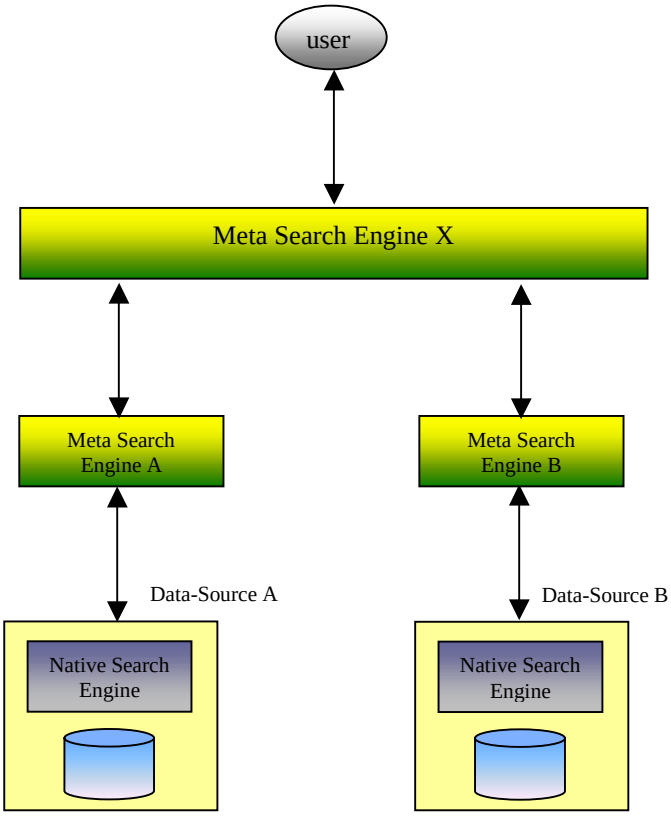

Figure 5: System Architecture