Content Management in Large-Scale Information Retrieval Systems

S. Beitzel
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL, U.S.A.
steve@ir.iit.edu

E. Jensen
Information Retrieval Laboratory
Computer Science Department
Illinois Institute of Technology
Chicago, IL, U.S.A.
ej@ir.iit.edu

Abstract ***Herein we introduce a system for managing the content of information systems in order to improve retrieval effectiveness over a large, constantly growing collection of structured and unstructured data. Specifically, we target the problems relating to the submission of new content from various sites, the seamless integration of said content into the main document corpus, and effective methods of searching the available information. In a unique interdisciplinary effort with the Institute of Design at IIT we have implemented a prototype for a document management system, which acts as a unifying abstraction layer that the various components of our architecture use to coordinate their efforts.***

Keywords: Content, Management, Information, Retrieval, Database

1 Introduction

As more and more information becomes electronically available, it is increasingly beneficial to carefully manage the content of large repositories of related data. Imposing structure on these repositories in order to maintain data quality and improve retrieval effectiveness can be highly useful to those members of the research community who are looking to build stable and informative knowledge bases for their fields [1].

As such a knowledge base begins to achieve wide acceptance, its growth will accelerate. In order to stay functional and relevant, the system will be forced to impose some kind of content management. Most existing large-scale systems, such as data warehouses [2, 3], do not, and cannot, pro-actively enforce rigid schemas on data in their sources as it is gathered. Other large-scale systems may use mediators [4, 5, 6] to solve the problem of schema integration across multiple heterogeneous sources as late as query time, attempting to cope with data sources as they currently exist.

Our development has been an interdisciplinary effort with the Institute of Design at the Illinois Institute of Technology. Our goal has been to develop a collaborative database for research in the emerging discipline of industrial design. As such, our prototype system is focused on actively managing content as it is created, avoiding the complicated problems of schema reconciliation and data integration [6, 7], and data quality investigation [8, 9, 10] as they apply to large collections of legacy data.

We have developed a prototype which uses a content management system that enforces schema compliance and data quality at data-entry time, avoiding the processing overhead of running schema reconciliation algorithms across a massive repository of data. Our prototype implements this content management system on top of the proven technology of a Rela-

tional Database Management System back-end and the flexibility of a sophisticated information retrieval engine to create a scalable and flexible platform for deploying large-scale collaborative repositories of information.

Section 2 will give background information on the problems at hand. Section 3 will discuss the main issues involved in maintaining the repository. Section 4 will discuss the architecture for our prototype, and Section 5 will contain a summary and directions for future work.

2 Background

One of the foremost problems in conducting research is performing an adequate literature review of a potential topic. Information is available in many venues, ranging from electronically available research papers and reports to conventional peer-reviewed printed publications. The large amount of content that is available is spread across various sources and mediums. As the amount of information grows, it becomes increasingly difficult to obtain high-precision results for typical user queries. It is clear that a large, collaborative database of research and related information for a discipline would be an invaluable tool for researchers looking to examine the existing knowledge base in their field.

In order for such a system to remain current and relevant to the field, all members of the research community must have an equal opportunity to submit candidate content for consideration. This allows for a widely dispersed collection of contributors to have a common forum in which to present their ideas for review. In addition, having access to a system such as this will allow for the state of the art to become almost immediately available to researchers, saving them the time that is necessary to attend a conference or await the latest publication of a particular journal.

If the repository is to be expected to handle traffic from a large number of research sites, it must have a highly powerful and stable back-end that will ensure data availability and integrity. This back-end must have facilities to support incoming queries as well as the submission of new candidate material.

A final key component that the repository must have is a highly effective searching mechanism. This will enable users of the system to efficiently locate data and research that is relevant to them. The search must be scalable to a large collection of documents and client users alike, and should provide a robust interface for both simple and advanced expressions of a query.

Much research has been conducted in reconciling schemas of, and integrating data from existing sources [2, 3, 6, 7]. Other research has included evaluating the quality of existing data [8, 9, 10]. These techniques have been limited in their application by a focus on minimizing the problems with large bodies of legacy data which is often heterogenous and of poor quality. We propose an innovative use of these proven technologies by applying them to pro-active content management. We display the effectiveness of this solution in a collaborative research database for the growing micro-context of industrial design which has a small but rapidly expanding data set.

3 Key Research Problems

In order for this database of information to be effective and widely accepted, it must meet certain guarantees - namely that its content remain current, reliable, and effectively searchable. In the following sections, we will discuss how each of these characteristics is crucial to the widespread acceptance and use of the database.

3.1 Content Management

In order for the repository to be most useful to researchers, the information it contains must be kept current. It is problematic to build new research from prior work that does not represent the current state of the art. In order to maintain a current representation of the

state of the art, active researchers in the field must be allowed to submit new information for inclusion into the repository. This has the added benefit of affording a common forum for even a highly dispersed research community. We propose that a system of content management be implemented, whereby participating clients of the system can submit new information in a uniform, verifiable fashion, which is then queued for peer-review. The submission system will enforce a rigid schema upon information being submitted, ensuring that all entries are uniform, making data reconciliation unnecessary. This may also be extended so that different types of documents will each have their own specific schema (e.g., technical reports, abstracts, bibliography entries, etc.). Additionally, all reviews and further revision of the proposed new content can take place electronically, and once it is approved for inclusion into the repository, the information is instantly available for public view.

This method of integrating new content has several advantages, most notably that the information becomes widely available in a short period of time – there is no need to wait for a conference date or publication of a journal. It is also advantageous to have an electronic method of submission and review, as it allows for uniform submission format, strict adherence to a submission schema for each given document type, and faster integration of the new data to the repository.

3.2 System Reliability

Regardless of how the content in the system is managed, the availability of the data and general reliability of the system are issues of paramount importance. This becomes especially evident as the system begins to receive an increasing amount of traffic. It clearly must be able to handle a very large number of concurrent requests if it is to be released for use to a community of sufficient size. In this way, we propose that the most effective method of implementing the back-end of the system is with the use of a Relational Database Management System.

Using an RDBMS as the platform for information storage brings tremendous enhancements to the robustness of the system. The usage of features such as database replication, transactions, and concurrency control all help to guarantee both the integrity of the data as well as high performance. In addition, virtually all RDBMS implementations focus significant research on query optimization and efficient data storage, so as to be scalable to very large collections.

3.3 Effective Search

The core purpose of a large-scale repository of information is to provide researchers with a large and diverse knowledge base pertaining to their field, which they can use to further the advances of their own research. Integral to this process is the ability of a client to quickly and effectively search through the database for information which they deem relevant. We considered two primary approaches to the problem of implementing an effective search function for the repository: using the RDBMS itself, and using a conventional information retrieval engine.

3.3.1 Searching with an RDBMS

On the surface, there appear to be many advantages in using the RDBMS to execute client queries. In this case, queries are entered in the native query language of the database, so the overhead of performing query translation is avoided. In addition, giving the end user access to the low-level features in the query language allows queries to be expressed with a very high degree of granularity. Finally, we would be able to take advantage of the database's query optimizer, which would save still more overhead in not having to implement that task ourselves.

One major difficulty with this approach is that many potential users may not be well-versed in the annals of database query languages, and may well find such an interface to be tedious and inhibiting. An obvious way

around this problem is to design algorithms for query translation, however, this approach also has several problems. Such algorithms would have to be highly efficient in order to return results to the client in a reasonable time given the overhead of query translation. It has also been found that this type of query translation is a very difficult research problem, especially if the client query is highly similar to natural language [11, 12, 13, 14, 15].

In the past, our group has done much prior work on using an RDBMS for information retrieval [16, 17, 18]. Given these concerns, however, we felt that a conventional information retrieval engine would be better suited for this task as it is capable of handling a wide variety of queries.

3.3.2 Searching with an Information Retrieval Engine

The use of a sophisticated IR engine for search alleviates the problems presented in translating natural language queries to equivalent SQL, but it introduces new difficulties, particularly if a client wishes to perform searches on only a particular field in the repository's documents. In a traditional information retrieval engine, data is generally assumed to be unstructured in nature. Because of this, an IR system is not usually concerned with the preservation of specific fields in the data. Traditionally, this type of application would be well suited to an RDBMS, as a database is designed to hold structured data, and is very much concerned with expressing its data as a collection of fields. We were able to integrate these aspects of the IR and database disciplines by extending our research prototype information retrieval engine, AIRE [19], to support the attachment of structured data "containers" to each document in the repository. These containers hold each field contained in a searchable document, as well as the type for that field. This integration of structured data and text, an extension of our IR lab's prior work in that area [16, 18], is used to provide a great improvement to the precision and granularity of the query system. This is done by extending information retrieval's vector space model [20] to allow fields to be indexed as various types. Each term type can then be weighted according to whether or not it is perceived as more or less relevant than other term types. The usefulness of this can be seen when one considers a document with a "title" field. If the IR engine finds that some query terms appear in the title field of a document, that document may be viewed as having greater relevance than other possible results. This extension of VSM is also a great improvement to the flexibility of the system, in that if a client issues a search, but wishes it to be limited only to author's names, we are able to use our information retrieval engine to query only the "Author" fields in its indexing structures, and return results accordingly. Another way in which we exploit the structured nature of the data to improve system effectiveness is through intelligent phrasing. In addition to statistical phrases, we build phrases based on the types of fields of text we index. For example, an "Author" field of "Ophir Frieder" would be parsed into several phrases including: "Ophir Frieder," "O. Frieder," and "Frieder, Ophir." This improves retrieval capability as users querying for specific authors are able to search for these phrases, minimizing irrelevant results caused by duplicate first or last names.

The information retrieval engine serves in tandem with the RDBMS to form the core for the back-end of this system. It periodically polls the database for new additions and updates its indexing structures accordingly. In this way, we are able to take advantage of the reliability of an RDBMS and the rich features of an IR engine, while maintaining a search interface with a very high degree of user-specified query granularity.

4 Architecture

We decided that since this repository must be made available to a highly dispersed population, a forms-based web interface would be most effective.

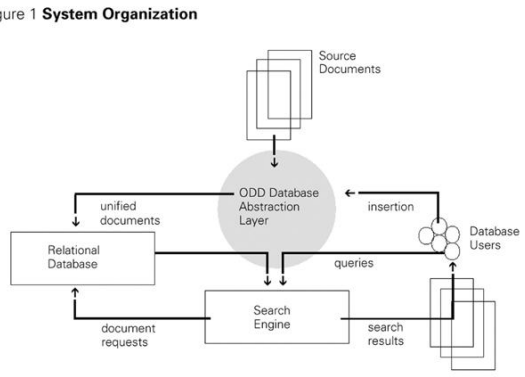

Figure 1: Content Management System Architecture [1]. Reprinted with permission.

4.1 Overview

The architecture for our repository prototype must support three fundamental operations. First, researchers must be able to submit new material for review. Secondly, reviewers must be able to view submitted content and either edit, approve, or discard it. Finally, the clients must be able to submit queries and view the relevant results. An overview of the architecture is shown in Figure 1.

The content management operations (document submission and reviewer approval) are each supported by a collection of forms-based web interfaces that are driven by Java Servlets. Each of these servlets ties in to our online document database (ODD) abstraction layer, which is a set of routines responsible for managing the propagation of data through the system. Data that is entered by clients is passed to ODD from the user interface, and ODD performs the necessary tasks of securing a connection to the RDBMS and ensuring that the data is properly added to the repository.

The search operations are also forms-based, servlet driven interfaces, but they interact directly with our IR engine AIRE and do not pass through the ODD Abstraction Layer. The architectural components for each of these fundamental operations in our prototype will be discussed in detail below.

4.2 Document Submission

In order to submit a document for review and possible addition to the repository, a researcher must navigate through a simple three-step process.

When a researcher visits the document submission section of the user interface, he or she is presented with a list of templates for each supported document type. After choosing the appropriate document type, the researcher must fill in all required data for the record (Author, Title, etc.). This data is verified to ensure data quality through a simple, but strict, set of rules. For example, author names must be entered as "Lastname, Firstname" so that when they are parsed by the IR engine it can weight the resulting terms and build phrases intelligently. Once the summary information has been entered, the researcher is presented with the option of attaching a PDF to the record, which would contain the full text of the document. If provided, the text of this PDF file will be indexed by our IR engine along with the summary information, in order to maximize the amount of searchable information about the record. Once the data entry is complete, the record may be previewed, and finally it is sent to the submission queue in the database. This is the final step of automated data quality assurance in that the database is designed using the most specific types possible, ensuring that dates are stored as type "Date," etc.

4.2.1 Submission Review and Approval

In order to examine submitted documents for approval, a peer reviewer must visit the review page and enter a valid password. Once he or she has been authenticated, a list of all available submissions is presented, and entries to be reviewed are selected. When a selection is made, the summary information for that record is displayed, along with a link to the attached PDF file, if one was provided. The reviewer is then given the opportunity to edit the information contained in the record, and approve or discard it accordingly.

All submission requests are handled by the ODD layer. ODD translates the submission requests into a set of SQL statements that inserts the summary data provided by the client into the submission queue, which resides in our back-end RDBMS. Any uploaded files are stored in a temporary holding directory and tagged with a unique identifier that signifies which record in the database they are attached to. If a record is approved, the ODD layer generates the SQL necessary to move the record from the submissions queue to the main document collection.

4.3 Query Interface

The query interface consists of a simple forms-based HTML page which interacts with the IR engine. This page contains a text box for entering a client's desired query, and the ability to filter search results by document type through the use of a series of checkboxes. There is also an advanced search option, where the user is presented with the ability to search only in specific fields, or to limit the search to a given date range. Range queries of this type are facilitated by the structured container extension to AIRE. The interface is simple, yet very powerful. It allows the user to express a very fine-grained query to the system without having to be trained in a complex query language.

4.4 Indexing Structures

As mentioned previously, the RDBMS and our information retrieval engine, AIRE, work in tandem to form the core of the system. In our architecture, the DBMS is used primarily for data management and storage, and AIRE provides all search capability. In order to do this, the database must be periodically polled for new documents, and AIRE must build a fresh index any time data has been added to the repository in order for it to be made searchable. Since the AIRE indexes and the contents of the database are not always perfectly synchronized, a small period of delay is experienced from the time that a document is approved to the time it will appear in the searchable index. This small cost is offset by the combined advantages of the database's stability and AIRE's rich collection of features.

5 Summary and Future Work

We have proposed an innovative content management system which works to improve retrieval through the integration of an RDBMS-based document management system and an information retrieval engine. Its effectiveness has been displayed in a prototype system which provides a collaborative research repository for the emerging discipline of industrial design. It is clear that the benefits gained from content management, combined with data integrity and stability from an RDBMS and the flexible search capability provided by a sophisticated information retrieval engine, make for an excellent platform on which to base the development of a large-scale information system.

There is much future work to be done in this area. Experiments will be conducted to determine the optimal method for updating the IR engine's index, including an optimal interval between updates. Once the optimal interval has been calculated, we will be able to further maximize the availability of data in the repository. We will use these results to extrapolate optimal intervals for varying collection sizes in an attempt to determine the effect that new documents have on increasingly large collections.

References

[1] Poggenpohl, S., O. Frieder, S. Beitzel, A. Friedman, E. Jensen, and E. Kim Infrastructure for Remote Collaborative Text Database Environments *Submitted for publication, Journal of Documentation, 2001.*

[2] Inmon, W., Building the Data Warehouse, John Wiley and Sons, 1993.

[3] Kimball, R., The Data Warehouse Toolkit, John Wiley and Sons, 1996.

[4] The Cooperative Database Project. http://www.cobase.cs.ucla.edu/index.html.

[5] Grossman, D., S. Beitzel, E. Jensen, and O. Frieder The IIT Intranet Mediator: An Overview *Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries, Switzerland, December 11, 2000.*

[6] Saelee, M., S. Beitzel, E. Jensen, D. Grossman, and O. Frieder Intranet Mediators: A Prototype *Proceedings of the 2001 IEEE International Conference on Information Technology - Coding and Computing (ITCC), Las Vegas, April 2001.*

[7] Cheng Hian Goh, Stphane Bressan, Stuart Madnick and Michael Siegel, Context interchange: new features and formalisms for the intelligent integration of information *ACM Transactions on Information Systems*, 17 (3), 1999.

[8] David Kaplan, Ramayya Krishnan, Rema Padman and James Peters, Assessing data quality in accounting information systems *Communications of the Association for Computing Machinery*, 41 (2), February 1998.

[9] Amir Parssian, Sumit Sarkar and Varghese S. Jacob, Assessing data quality for information products *International Conference on Information Systems*, December 1999.

[10] Donald P. Ballou and Giri Kumar Tayi, Enhancing data quality in data warehouse environments *Communications of the Association for Computing Machinery*, 42 (1), January 1999.

[11] Liddy, E. D., Enhanced Text Retrieval Using Natural Language Processing. *Bulletin of the American Society for Information Science*. Vol. 24, No. 4, 1998.

[12] Feldman, Susan. NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval. Accepted for publication in *Online, Inc., 1999*.

[13] Thompson, C. A., R. J. Mooney, L. R. Tang. Learning to Parse Natural Language Database Queries into Logical Form. Accepted for publication in *1997 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition*.

[14] Yang, C., O'Shaughnessy, D. Development of the INRS ATIS System. Accepted for publication in *ACM Intelligent User Interfaces, 1993*.

[15] Damerau, F. J. Problems and Some Solutions in Customization of Natural Language Database Front Ends. Accepted for publication in *ACM Transactions on Office Information Systems, Vol. 3, No. 2, April 1985, Pages 165-184*.

[16] Grossman, D. A., O. Frieder, D. O. Holmes and D. C. Roberts Integrating Structured Data and Text: A Relational Approach *Journal of the American Society of Information Science*, 48 (2), February 1997.

[17] Lundquist, C., O. Frieder, D. Holmes, and D. Grossman. A Parallel Relational Database Management System Approach to Relevance Feedback in Information Retrieval. *Journal of the American Society of Information Science*, 50 (5), April 1999.

[18] Frieder, 0., A. Chowdhury, D. Grossman, M. C. McCabe On the Integration of Structured Data and Text: A Review of the SIRE Architecture *DELOS Workshop on Information Seeking, Searching, and Querying in Digital Libraries, Zurich, Switzerland, December 2000.*

[19] Advanced Information Retrieval Engine. http://ir.iit.edu/aire/.

[20] Salton, G., C.S. Yang, and A. Wong A Vector-Space Model for Information Retrieval *Communications of the ACM, 18, 1975.*