XLibris: An Automated Library Research Assistant

Andrew Crossen, Jay Budzik, Mason Warner, Larry Birnbaum, and **Kristian J. Hammond**
Intelligent Information Laboratory
Northwestern University
1890 Maple Ave.
Evanston, IL 60201 USA
+1 847 467-1265
{crossen, budzik, warner, birnbaum, hammond}@infolab.nwu.edu

ABSTRACT
While recent work has focused on providing tools and infrastructure for users to access electronic information over the Internet, the relationship between the physical world and information available online has been relatively unexplored. Information about a user's location, and the objects she interacts with, can be sufficient to recognize enough of the user's task to drive retrieval of online information relevant to the task at hand. The XLibris system automatically retrieves, aggregates, and delivers information about books to users as they are checked out of the library, using information about the books themselves and the user's task. XLibris locates books in the Dewey Decimal subject hierarchy to automatically search for the most relevant information about the book for the user, tailoring both the sources queried and the information returned based on the book's position in the hierarchy.

Keywords
Information aggregation, automated retrieval, metasearch, ubiquitous computing.

1. INTRODUCTION
The wealth of information available online is staggering. While all of this information is *available* on the Web, the knowledge contained in Web pages is not necessarily put to use by users in their day-to-day activities. A recent study [12] suggested that users primarily access the Web through search engines. Users of search engines must first decide they need information, navigate to the appropriate engine, and then distill their request into keywords describing it.

While this mode of information access may be useful for many *digital* tasks, that is, activities that occur on or around a computer, it tends not to be as useful for those activities we normally associate with the physical world, such as browsing books in a library or bookstore.

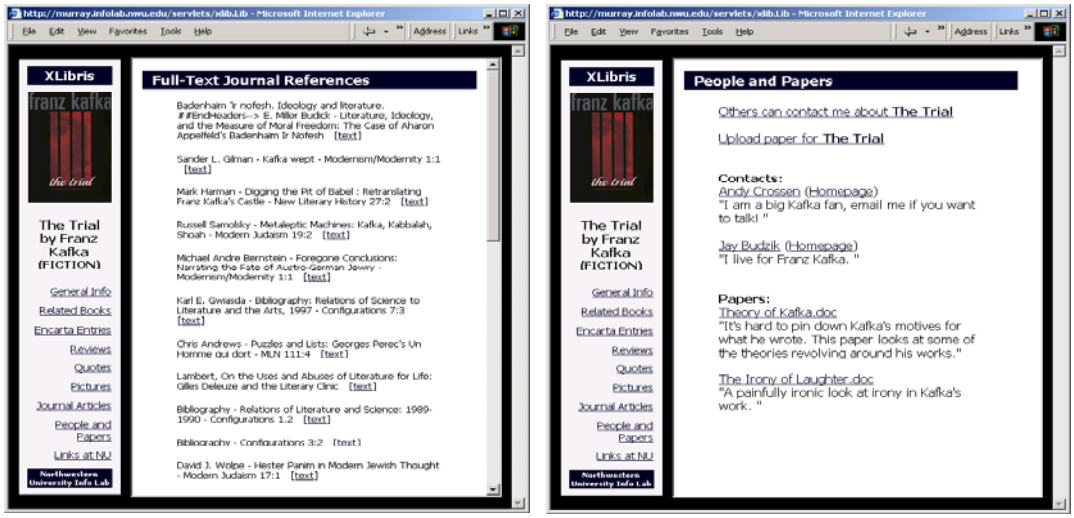
Furthermore, since many of our actions are opportunistic and reactive [2], the requirements of having a computer, an Internet connection, knowledge of appropriate information sources (and how to access them), and the patience to condense a need into keywords, severely limits the situations in which it is practical to look for online information. Interviews with Northwestern librarians underscore this point, indicating students rarely use very costly, specialized library databases, even though the content in such databases is of the highest quality.

Recent work (e.g. [8, 10]), has focused on exploring the relationship between the physical world we inhabit and the virtual world of information, some of which is aimed at improving the way information is accessed. Our focus is on how to leverage the objects the user interacts with, coupled with a limited knowledge of the user's task, to automatically gather and present information to the user. The XLibris system does this for the world of books. XLibris allows users to scan or enter a book's barcode into the system, and automatically receive on-point information about that book and related content that is delivered on a mobile device, or asynchronously via email.

1.1 Usage Scenario
Suppose our user is a college student at the library. She is in the check out line with a copy of Franz Kafka's *The Trial,* a book she needs for a research paper on 20th century German fiction. She hands it to the checkout clerk along with her student ID card. The clerk scans her card, and then the book. At this moment, the system has two key pieces of information: a unique identifier for the student, and a unique identifier for the book. The student ID is used to look up her email address. The email address and book ID are then processed by the XLibris system. The student leaves the library, and upon return to her dorm room has an email message containing a pointer to an automatically generated document about *The Trial* (see Figures 1a and 1b). The page contains sections referencing related books, humanities journal articles about *The Trial*, links to the home pages of classes using *The Trial* at other universities, biographies of Kafka, pointers to papers other students have written about *The Trial* and other German fiction works, and their email addresses, among other things.

In doing so, we estimate XLibris saves the student about two hours of research work.

2. THE XLIBRIS ARCHITECTURE
The system is activated by scanning a book's barcode. The *Request Context Selector* produces a list of information items to retrieve and the queries used to retrieve them (the *Request Context*), based on the type of book and the role of the user (professors, for instance receive a different kind of page, but this



(a) (b)

Figure 1. The XLibris Web interface for information about books. The left column contains pointers to various categories of information. Figure 1a displays automatically retrieved humanities journal articles about *The Trial*. Figure 1b displays people who have registered as contacts for this area, as well as papers that have been uploaded.

is beyond the scope of this paper). This information is fed into the *Access Planner*. The planner, a simplified version of the STRIPS classical planner [6], knows about specific data sources and what information they require as input and provide as output. Given the Request Context, the planner produces a directed graph of repositories capable of satisfying those requests (termed *Information Source Adapters*, or ISAs [4]), ordered by the dependencies inherent in the data collection process, queries to execute on them, and a display template to hold the final results. The graph representation reflects redundancies in the data collection process that allow the system to recover during execution when sources are unavailable. For example, given an initial book barcode and a request for that book's author, two ISAs are returned with associated queries that, when executed in sequence, will return the author of the book with that barcode. In this case, a barcode-to-ISBN translator and a Library of Congress adapter and associated queries are returned.

The XLibris *Plan Executor* is responsible for running the access plan. Gathering the requested data using the source-specific retrieval mechanisms of the data repositories represented, the Executor stores intermediate results in its memory. In the event of a retrieval failure, contingency clauses are invoked to retrieve information from alternate sources that have semantically similar input and output characteristics as those adapters that failed.

When all information goals have been satisfied in the plan, an aggregate data object with all requested information items is handed to the *Presentation Engine*. The Engine builds a results page from the information items gathered by the plan executor, and a pre-built display template to house the results.

Using an XML-based dialect that allows Java function calls (the entire system is written in Java) to be interspersed with standard markup language, the template language is a superset of standard display languages. Templates can be written and viewed in standard browsers without having information to fill them initially, in the same manner as a template query is built for each ISA.

The Presentation Engine uses the aggregated data from the plan executor to fill in the blanks in the template, producing the final results document for display to the user. This generic display mechanism allows the display format to be tailored to the characteristics of the many devices a user might employ to access the system. For example, versions of the XLibris system have been deployed on WAP-enabled cell phones and a Palm device equipped with a barcode scanner and cellular modem.

2.1 Source Representation and Query Generation

ISAs can represent any data source, including ODBC databases, Web search engines, and special-purpose repositories. Each source has its own retrieval language and content that are represented by the adapter and its position in the Dewey hierarchy. ISAs in the XLibris system map the source-specific query syntax into a standard query language that all XLibris adapters use. In addition, ISAs and queries with unbound variables are associated with appropriate subject areas in the Dewey tree. The position of a source and its associated queries in the Dewey tree determines if the source will be accessed for a given book that is scanned into the system. Queries can contain variables that are bound at runtime by the plan executor. The variables in queries associated with a source determine what information is required to run the adapter. In addition, each adapter contains a representation of what it produces. For example, the Library of Congress card catalog adapter requires an ISBN number, and produces, among other things, that book's title, author, publisher, and subject headings. This representation allows the access planner to automatically generate a chain of adapters that, when run, will gather the information necessary to fill in a presentation template.

2.2 Automatic Source and Data Selection

An important component of XLibris is its automatic source and data selection facilities. Unlike traditional metasearch facilities [9, 13, 14], XLibris' adaptive source and data selection components allow the information gathered to be more sensitive to the user's task, and the kind of object given as input to the system. While returning a generic set of information from a static group of sources for each query may be useful to a user, especially if those sources are broad in content and the system knows little about the user and her goals or the object in question, this is not sufficient when we consider finer-grained information sources that only address a specific domain (as are commonly available in libraries).

Consider, for example, the kind of information that might be useful to a library user checking out a textbook on abstract algebra, in contrast with what would be useful to a different user, interested in *The Trial*. In the context of writing a research paper on German fiction, the user will generally be more interested in finding out about the author than in the case of the user checking out the algebra textbook. Moreover, in the case of the algebra student, she might find example algebra problems and solutions useful, whereas example problems are not appropriate for the German fiction student. Clearly, XLibris cannot be limited to a static list of information sources nor a fixed list of queries, if it is to make full use of more specific resources and more closely support the goals of a user in a given context.

The problem of selecting appropriate information sources is addressed by employing a mapping table linking groups of context-based data requests and the Information Source Adapters needed to retrieve them with appropriate nodes in the Dewey Decimal hierarchy. For *The Trial,* XLibris starts looking for ISAs and queries at category 833, German Fiction. The system drives up the tree through categories 833, 830, 800, and a global category, gathering ISAs and queries indexed by the nodes along this path, and adding them to the pool of sources and queries that will be used in gathering information for the scanned book. In our Kafka example, a German fiction journal database and specific queries are associated with node 833 and added to the access plan. Category 830 (Literature of Germanic Languages) has no associated sources, so the system moves up to category 800, Literature & Rhetoric. Sources and queries that gather information about the author are located at this node and are also added to the plan. The planner finally picks up the rest of its sources at the global category, including the initial card catalog lookup and an adapter that gathers information on related books, among others.

In contrast, consider when a user scans an abstract algebra textbook, a work of non-fiction, into XLibris. XLibris starts at category 512, Algebra and Number Theory. A general search engine is used to retrieve pages containing example algebra problems and solutions by automatically constructing a query based on the Dewey subject headings (in this case, "Abstract Algebra"), the title of the book, plus the words "solution," "example," and "problem". The parent category is 510, Mathematics, where queries and sources aimed at retrieving general mathematics sites are added. Moving upward, category 500, Natural Sciences & Mathematics, is ignored in this case because there are no sources or queries associated with it due to its generality. At the global category, additional sources and queries are picked up that generate results useful for any kind of book.

3. FOSTERING VIRTUAL COMMUNITIES

Each book presented to the system is situated in a specific conceptual node defined by the Dewey hierarchy. A natural extension to building information resources associated with particular topics is to form a user community organized in the same manner.

For example, consider two users of the XLibris system, both doing research on German fiction. Each user has checked out a book situated in the German fiction Dewey category. While the two books are different, the goals of the users may be similar. XLibris facilitates communication between the two users, leveraging the similarity of their goals, adding value to their experience by providing access to potential expertise of other users. In addition, users can upload relevant documents into the space, indexed conceptually by their Dewey category (see Figure 1b).

These two users, as well as the papers listed are not *object*-specific associations, but *concept*-specific ones. A user requesting an XLibris page for Herman Hesse's *Steppenwolf* (another German fiction piece) would see similar information because the books are related by the Dewey taxonomy. Unlike many community-oriented sites, XLibris automatically places the user in a relevant topic space based on the object they are manipulating. This kind of virtual community building is media-agnostic: it can be extended to use any kind of communication channel, including text-based chat, or videoconferencing.

4. RELATED WORK

Real-time aggregation of information from multiple data sources has representation in systems like SavvySearch [9], and MetaCrawler [14]. These systems save the user time by searching many sites simultaneously and retrieving a synthesis of the best results from each. However, they still require an initial query from the user, which can often be very ambiguous [4]. The results that are obtained from manual search using many tools of this type tend not to be organized in a coherent way. In this respect, XLibris is similar to ISI's Ariadne [11] system, which has focused mainly on the machinery of dynamic information integration in constrained settings.

Andersen Consulting's Pocket BargainFinder system [3] is also related to our work on XLibris in that it allows users to easily find price-point information about a book using a mobile device. The results XLibris provides go far beyond price point information in an attempt to support research and community building, instead of focusing on sales.

Because XLibris operates in the context of the objects the user interacts with, it offers a robust mechanism for determining what to look for, where to look for it, and how to organize the retrieved results, freeing the user from the difficult task of manual search. Additionally, XLibris builds virtual communities around the objects (and associated concepts) users encounter. XLibris provides immediate and automatic access to people and documents associated with the object in hand without requiring any explicit intervention on the part of the user. These aspects of the system integrate and advance previous research in information integration and ubiquitous computing.

5. CURRENT AND FUTURE WORK

We have developed several additional systems that leverage the generality of the XLibris architecture, including an over-the-counter drug interaction warning system (deployed on a mobile

device), an information assistant for music (that operates with common MP3 and CD audio players), and a pre-purchase consumer electronics product comparison agent. Building such systems required developers write the necessary ISAs and display templates, as well as define an object hierarchy and task context for the system. Even though this process is fairly straightforward for developers, our goal is to deploy the system in libraries and stores without requiring them to hire full-time programmers to maintain them. To this end, we are in the process of creating a suite of tools for generating new systems using the XLibris architecture, as well as modify existing ones. Current versions of the tools make use of wrapper induction techniques (e.g., [1]) to make creating ISAs easier, and include graphical knowledge engineering tools so users can easily map ISAs and task contexts directly onto the object hierarchy for their domain.

In addition, template translation tools are being built to facilitate translation between the different display characteristics of the devices used to deploy the system, so that a single representation can be used to generate multiple templates for devices with widely different display capabilities, as well as exploit synergies between different kinds of devices becoming available to users (e.g., ubiquitous displays and handheld devices).

After the initial XLibris book system was developed, students and teachers at Evanston Township High School (ETHS), a local public high school, evaluated the system. Students thought the information provided by the system would be useful to them, and found the interface easy to use. They especially liked the fact that content would be delivered to them automatically, without requiring explicit intervention on their part. The teachers were also excited about the system, although they said they wanted more control over what kind of information was delivered to the students so it could be more on point with the curriculum they were teaching in their classes. As a result, we have been working with ETHS teachers to design tools that allow teachers with no programming skills to encapsulate online data sources, as well as select from pre-existing sources in the system, defining a system context for their class.

6. CONCLUSION

The XLibris system automatically retrieves, aggregates, and presents information about objects in the physical world, using information about the objects themselves and the user's task. Users interact with XLibris by scanning the barcode of the object. XLibris then locates this object in a concept hierarchy and automatically searches for information about the object for the user, based on its location in the hierarchy. The XLibris system attempts to bridge the gap between the physical world of objects and tasks, and the virtual world of information by automatically delivering custom content to users as they interact with objects in the world.

7. REFERENCES

1. Adelberg, B., "NoDoSe: A Tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents," in *Proceedings of ACM SIGMOD*, ACM Press, 1998.
2. Agre, P., and Chapman, D., "What are Plans For?," *Robotics and Autonomous Systems*, 6,17-34, 1990.
3. Brody, A. B., and Gottsman,E. J., "Pocket BargainFinder: A Handheld Device for Augmented Commerce," in *Proceedings of First International Symposium on Handheld and Ubiquitous Computing (HUC '99)*, (Karlsruhe, Germany)1999.
4. Budzik, J., and Hammond, K. J., "User Interactions with Everyday Applications as Context for Just-in-time Information Access," in *Proceedings of The 2000 International Conference on Intelligent User Interfaces*, (New Orleans, Louisiana, USA), ACM Press, 2000.
5. Dewey, M., "Catalogs and cataloging: a Decimal Classification and Subject Index.," *U.S. Bureau of Education. Public Libraries in the United States of America: special report, part I.*, 623-648, 1876.
6. Fikes, R. and Nilsson., N. J., "STRIPS: A new approach to the application of theorem proving to problem solving.," *Artificial Intelligence*, 2,189-208, 1971.
7. Firby, J., *Adaptive Execution in Complex Dynamic Worlds*, Ph.D. Thesis, Yale University Technical Report, YALEU/CSD/RR #672, January 1989.
8. Höllerer, T., Feiner, S., Terauchi, T., Rashid, G., Hallaway, D., "Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System," *Computers and Graphics*, 23(6), 779-785, 1999.
9. Howe, A. E., Dreilinger, D., "SavvySearch: A Meta-Search Engine that Learns which Search Engines to Query," *AI Magazine*, 18(2), 19-25, 1997.
10. Ishii, H., "Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms," in *Proceedings of Conference on Human Factors in Computing Systems (CHI '95)*, (Denver, USA), ACM Press, 1997.
11. Knoblock, C., Minton, S., Ambite, J., Musela, I., and Philpot, A., "Compiling Source Descriptions for Efficient and Flexible Information Integration," in *Proceedings of The Fifteenth National Conference on Artificial Intelligence*, (Orlando, FL, USA), AAAI Press, 1999.
12. Lawrence, S., and Giles, L., "Accessibility of information on the web," *Nature*, 400,107-109, 1999.
13. Marcus, R. S., "An Experimental Comparison of the Effectiveness of Computers and Humans as Search Intermediaries.," *Journal of the American Society for Information Science.*, 34(6), 381-404, 1983.
14. Selberg, E., and Etzioni, O., "The MetaCrawler Architecture for Resource Aggregation on the Web," *IEEE Expert*, November), 1996.