Addressing Heterogeneity in the Networked Information Environment

Michelle Q Wang Baldonado and Steve B. Cousins
Computer Science Department, Stanford University, Stanford, CA 94305, USA
(michelle@cs.stanford.edu, cousins@cs.stanford.edu)

Abstract

Several ongoing Stanford University Digital Library projects address the issue of heterogeneity in networked information environments. A networked information environment has the following components: users, information repositories, information services, and payment mechanisms. This paper describes three of the heterogeneity-focused Stanford projects—**InfoBus**, **REACH**, and **DLITE**. The **InfoBus** project is at the protocol level, while the **REACH** and **DLITE** projects are both at the conceptual model level. The **InfoBus** project provides the infrastructure necessary for accessing heterogeneous services and utilizing heterogeneous payment mechanisms. The **REACH** project sets forth a uniform conceptual model for finding information in networked information repositories. The **DLITE** project presents a general task-based strategy for building user interfaces to heterogeneous networked information services.

1.0 Introduction

The recent surge of research in "digital libraries" has energized discussion about what it is that makes traditional libraries valuable. We adhere to the widely articulated view (1, 2) that libraries are much more than archives of information—they are also social institutions. An implication of this view is that networked information must also be considered in terms of the larger context in which it is situated. We refer here to this larger context as the *networked information environment*. In the Stanford University Digital Library project (3), we have focused on these key components of the networked information environment:

- users
- information repositories (which contain networked information)
- information services
- payment mechanisms

Historically, libraries and information systems have focused on environments in which each of the above components has been fairly homogeneous. However, as we move into the era of the networked information environment, we need to rethink this assumption of uniformity (4). The popularity of the World Wide Web (WWW), which allows users on a variety of hardware platforms to access and provide a useful and large cross-section of information and services, gives us an inkling of what is to come. We believe that understanding and designing for the types of heterogeneity that will arise in future networked information environments is a research area of great importance.

In this paper, we will discuss three ongoing Stanford University Digital Library projects (**InfoBus**, **REACH**, and **DLITE**) that revolve around the issue of heterogeneity in networked information environments. In the rest of this introduction, we will articulate the background necessary to understand how these projects interrelate. First, we will explain each of the components of the networked information environment in more detail and enumerate the types of heterogeneity found in each that are critical for networked information environment designs. Then we will discuss the different levels at which the design of networked information environments can occur.

1.1. Heterogeneity in networked information environments

Users: Well-known ethnographic studies (5, 6) have established that user populations exhibit great heterogeneity. Here, we distinguish five important dimensions of heterogeneity: range of activities, experience, style, geographic location, and available tools. The activities undertaken by an individual user might include writing and publishing information; collecting, organizing, and analyzing resources; communicating and collaborating with others; and so on. Thus, designers must think about styles of interacting that work for the spectrum of a user's activities. In addition, variations in a user's experience and style affect the designer's decisions about what must be done to enable the user to work effectively. Geographic location matters not only for cultural issues in design, but also for understanding how to facilitate collaboration and sharing across multiple locations. Furthermore, the tools (hardware/software platforms) available to the user for accessing the networked information environment may vary widely in terms of power and capabilities, especially in settings where legacy systems are widespread. Accordingly, designers must strike a balance between taking advanced features for granted and limiting designs to the lowest common denominator.

Information repositories: The set of information repositories available in the networked information environment will encompass a wide variety of existing information and meta-information sources. Examples include traditional library collections, digital images, e-mail archives, video, on-line books, and scientific article citation catalogs (containing only meta-information about the articles, not the articles themselves). These repository examples dramatically differ from each other in repository type, in the genre (books vs. movies), modality (images vs. text), and subject (entertainment vs. science) of the described materials, and in the schemes employed to do the describing (referred to as *cataloging schemes* in the library world). This variety becomes especially important for designers of networked information environments because users often want to compare materials found in one repository to materials found in another repository—or to search multiple repositories in a single interaction.

Information services: We expect that a diversity of independent, distributed information services will emerge in the networked information environment. Examples here include services such as summarization, translation, archiving, copy detection, publishing, information-finding assistance, and document delivery. Instances of these services are likely to require different access protocols, levels of expenditure, and execution times. For example, an automatic language translation service might take only a few minutes, whereas a service that employs human translators might take a week or more. Designers must be sensitive to how this variability is handled, because users have different expectations about how interactions should proceed depending on both the financial and time costs involved.

Payment mechanisms: Today, many research libraries charge departments individually for costly services such as online database access. We envision that payment will become an increasingly important part of the networked information environments of the future. Different forms of payment mechanisms (credit cards vs. cash) will abound, just as they do in our everyday world. Furthermore, charging at low levels of granularity (analogous to phone companies charging for individual phone calls) may become a common practice in the networked information environment. Employing mixtures of payment models, such as pay-per-view and subscription, may also become standard. For people to use different payment mechanisms and models as easily as they currently do, designers must understand and respond to these changes.

1.2. Levels of networked information environment design

Our analysis of the key components that make up the networked information environment makes it apparent why heterogeneity is a complex and important issue. We distinguish and briefly discuss here three levels of design that must take this heterogeneity into account:

- protocol
- conceptual model
- visualization

Protocol: Protocols form the base infrastructure for networked information environments. The design of protocols involves several important heterogeneity-related issues, including achieving interoperability and balancing access trade-offs. Interoperability refers to the need to access and pay for different information repositories and services in a uniform way. Some of the variables that must be balanced in deciding how to access each service include the time required for initially contacting the service, the time necessary to transport information back and forth, the billing schemes in effect, and the frequency of service update. Prior important work in this area includes the research behind System 33/GAIA (7, 8) and the development of the Z39.50 (9), and HTTP (10) standards. The first project that we describe in this paper (**InfoBus**) is centered at the protocol level. It has addressed the problem of accessing heterogeneous services and utilizing heterogeneous payment mechanisms.

Conceptual model: Conceptual models provide structure for the user's view of the networked information environment and make explicit the space of available actions. Research on this topic has taken place in the fields of information retrieval (11), library studies (especially the work on cataloging and classification, (12)), databases (13), graphical user interfaces (14), and computer supported cooperative work (15). Two of the projects we describe in this paper approach networked information environment heterogeneity from the user perspective. One project (**REACH**) addresses the particular problem of finding information in networked heterogeneous information repositories. The other project (**DLITE**) has developed a general task-based strategy for building user interfaces to heterogeneous networked information services.

Visualization: Visualization techniques are necessary for displaying the various components of the networked information environment to the user and for conveying visually the underlying conceptual model. Influential research in this area includes work on fisheye views (16) and on interactive 3D representations of information (17). In this paper, we do not discuss additional Stanford work on visualization.

2.0 The InfoBus Project

The goal of the Stanford **InfoBus** project is to provide easy access to all of the information and services that will be part of the Internet. We are building a testbed of information repositories and services related to computing.

Testbed Collections: Since the project is focused on the computing literature, our initial testbed consists of materials from commercial citation databases such as those in Dialog (from Knight-Ridder), from library catalogs, and from the WWW. In order to ensure that the utility of our tools is not limited to the computing literature, we make interesting collections from other digital library projects accessible in the testbed. For example, as part of our collaboration with the University of California at Santa Barbara, our users can search their digital map library.

Testbed Services: In addition to providing access to a wide range of information, our tools provide access to the services which help to organize and manipulate that information. Within the project, we are building services for query translation, citation management, and copy detection, and others. Our tools are designed to link in external services as well. We are linking in text summarizers, format translators, and image manipulation services which are run at other organizations and over which we have no direct control.

In the rest of this section, we describe the general structure of the **InfoBus**, an architecture for digital library interoperability. We will summarize two protocols we have developed, one for searching, and one for providing integrated access to payment mechanisms. All of this infrastructure is taken for granted in the user interface work we will describe.

2.1. The InfoBus architecture

The **InfoBus** is designed to make it easy to connect a wide variety of heterogeneous information objects and services together (18). It is based on the assumption that there will not be a single standard for information exchange forthcoming (even if there is, there will continue to be legacy systems to connect). Z39.50 and HTTP both have large followings, but neither seems likely to displace the other in the near future. The **InfoBus** technology is based on distributed object technology. We use a free implementation of CORBA (19) called ILU (20).

Distributed objects communicate with each other via method calls. In order to link in services which are not CORBA objects, we build proxies which are CORBA objects that act as service clients and speak the native protocol of the services. For example, a service may use Z39.50, so its proxy would convert method calls into appropriate Z39.50 requests. We have built proxies for services which are accessed via HTTP, Z39.50, and Telnet. The **InfoBus** architecture allows many user interfaces, many services, and many protocols to be integrated together.

2.2. The InterOp protocol

When distributed objects are designed to work together, the sequence of method calls that are possible can be thought of as a protocol. In collaboration with researchers at the University of Michigan, our project has developed an interoperability (InterOp) protocol for search, which provides flexibility to both client and server, and is more general than HTTP or Z39.50. We have written proxies for HTTP-based services and Z39.50-based services using this protocol, and we have used the protocol to interoperate with digital library projects at other universities.

The InterOp protocol is designed to provide the proxy builders flexibility to control information flow. Whereas HTTP is designed for stateless servers, and Z39.50 requires servers to keep state during a connection, the InterOp protocol allows those decisions to be made dynamically. The InterOp protocol is described in detail elsewhere (18).

2.3. InterPay

Digital information raises the issue of how information providers will be compensated for their efforts, since the doctrine of first sale may no longer apply. Various payment *models* are possible (pay-per-view, subscription, bulk orders, advertising) and various payment *mechanisms* are being proposed (credit cards, digital cash, micro-payment schemes, accounts). Our approach, InterPay, was to build a system which would allow various models and mechanisms to co-exist. A primary contribution of InterPay was to distinguish three layers of functionality, as shown in Figure 1.

Figure 1

Here is a simple scenario from a user's point of view to give the flavor or InterPay. Assuming that the user has set up a couple of accounts, she simply requests information as usual using her *client* (e.g. a WWW browser). The client passes a pointer to the *payment agent* along with the request, and if payment is required, the service's *collection agent* contacts the payment agent to arrange the transaction. The payment agent may bring up a dialog box on the user's screen to request confirmation before instructing one of the *payment capabilities* (PCs) to transfer funds. Once funds are transferred, the collection agent informs the service and the information is returned as usual.

InterPay is designed to support a wide variety of payment mechanisms and trust models. For example, a service might not mind sending information back to the browser before funds transfer has been confirmed, and this model is supported as well. Current work on InterPay is exploring various "shopping models" involving issues such as price negotiation and alternative delivery mechanisms.

3.0 The REACH Project

In this section, we explore **REACH**, a conceptual model for finding information in networked heterogeneous information repositories. First, we motivate the need for this model by looking at types of information-finding strategies and at how users employ these strategies in traditional libraries and on the WWW. Second, we present the main characteristics of the model. Finally, we give a brief description of SenseMaker, a tool built using the **REACH** principles.

3.1. Background and issues

Strategies for finding information are usually classified as either *searching* or *browsing*. Typically, a user who is *searching* formulates a partial description of the items desired, discovers which specific items match that description, and then begins the process again. In contrast, a user who is *browsing* navigates from one neighboring item to another. In traditional libraries, neighbors are items that are physically close together. On the WWW, neighbors are items that have links to one another.

The interleaving of searching and browsing strategies can be very powerful. A look at how users employ hybrid searching/browsing strategies in traditional libraries and on the WWW illustrates this point. Furthermore, these examples highlight issues that are important for information finding in networked heterogeneous information repositories.

Traditional libraries deal with heterogeneity by organizing items both on the shelf and in the card catalog[1] via a consistent classification and cataloging system (12). On the shelf,

---

[1] We give the example here of traditional card catalogs, but many of the same principles carry over to online catalogs.

items are usually arranged according to a classification scheme that groups items together by subject. In the card catalog, several different organizational schemes are common. Almost all depend on catalogers preparing a title card, an author card, and multiple subject cards for a single library item. In a dictionary card catalog, all of these cards are combined together and arranged alphabetically. In a divided card catalog, there is one section for each card type. When library patrons make the transition from searching for an item in the card catalog according to a particular dimension (e.g., author) to browsing the shelf, they often find valuable items which did not turn up during the search. This phenomenon, often referred to as “serendipity,” occurs because patrons are likely to value items that are on the same subject as the original item that drew them to the stacks.

The infrastructure of the WWW offers similar rewards for interleaving searching and browsing. Web search engines (e.g., WebCrawler) give users lists of URLs for pages that match some specified criterion. These pages then serve as starting points for browsing because they contain links to neighboring items. This strategy is generally successful because links on a page of interest are likely to point to other pages of interest.

Abstracting from both of these cases, we observe that hybrid searching/browsing strategies work well when users can move easily from one type of organization to another. In the library, patrons might begin by looking at the resource collection organized by author and move to looking at it organized by subject. On the WWW, users might begin by looking at the resource collection by subject (approximated through keywords) and move to looking at it organized by references (which appear as links). In light of these examples, we argue that information-finding models for networked heterogeneous information repositories should facilitate the transitions from one finding strategy to another and from one type of organization to another. In the next section, we set forth a new conceptual model that we have formulated with the above criteria in mind.

3.2. REACH: a new conceptual model

The conceptual model we have developed for information finding in networked heterogeneous repositories is called **REACH**, which stands for **R**ecursive **E**xtensible **A**ctive **C**ard catalog for **H**eterogeneity. It rests upon two central ideas:

- Virtual card catalogs enable users to view collections of information in terms of multiple dimensions.
- Active card catalogs enable users to employ hybrid browsing/searching strategies.

In the rest of this section, we describe how **REACH** builds upon the traditional card catalog model to realize each of these ideas.

Virtual card catalogs: We discuss first the principles behind virtual card catalogs, then show how to construct them and how to make them work for very large dynamically defined collections (consisting of multiple information repositories).

Virtual card catalogs allow users to see collections of information according to multiple dimensions because each section of the virtual card catalog organizes the collection according to a different dimension. Traditional library card catalogs are often limited to the dimensions of author, subject, and title due to cataloging costs and physical space constraints. A section in a virtual card catalog is much more lightweight and can even be computed on the fly. Thus, a virtual card catalog can have a wide variety of sections. For example, a virtual card catalog for a research-oriented collection might add sections for research group and author institution to the traditional author-subject-title triplet (see Figure 2).

Figure 2

The crux of constructing such a virtual card catalog is determining its organizational scheme. For a heterogeneous collection, the main issue is that the included repositories are likely to describe their contents using a variety of meta-information schemas. Examples of existing schemas include USMARC, the Z39.50 bib-1 attribute set, and the Scientific and Technical Attribute Set (STAS), among others.

The **REACH** model solves this problem by introducing an interlingua schema, into and out of which we can translate meta-information encoded using existing schemas (see Figure 3). The **REACH** interlingua allows us to treat existing schemas in a uniform way, just as the **InfoBus** InterOp protocol allows us to treat existing services in a uniform way. In both cases, the development of an interlingua rather than a standard means that no changes need be made to schemas and services that are already in use.

Figure 3

The specific interlingua schema we have developed for **REACH** encodes hierarchical relationships among attributes, including both specialization and composition relationships. For example, we can represent the facts that a reporter is a specialization of an author and that an author name is composed of a first name and a last name (see Figure 4). Comparing the meta-information available from each repository of interest via this hierarchical interlingua schema allows us to arrive at a small common schema for the overall collection. The set of virtual card catalog sections (the organizational scheme) is then based upon the elements in this common schema. For example, consider a newspaper repository A that encodes “reporter,” and a technical article repository B that encodes “author.” The common schema for a collection composed of A and B will include only “author,” since **REACH** will observe that “reporter” is a specialization of “author.” Correspondingly, **REACH** will include an author section when it creates the virtual card catalog for the collection. Thus, this solution allows us to handle *heterogeneity*. In addition, we must also consider what happens when new repositories with new meta-information schemas enter the networked information environment. We address this issue by making the hierarchical interlingua schema *extensible*.

Figure 4

The question still remains of how to make virtual card catalogs work for very large dynamically defined collections. The **REACH** model adds two concepts to the card catalog model to achieve this goal: bundling and recursiveness.

First, we look at how bundling enhances the notion of a card catalog. **REACH** can bundle together cards that have the same or similar main values (e.g., the same author or nearby geographic locations) and replace each bundle by a “cover-card” describing the bundle’s common characteristics (see Figure 5). Adding this higher level structure to the virtual card catalog allows users to get an overview of its contents by browsing.This concept of bundling has roots both in the database and information retrieval fields. Database query languages such as SQL (22) provide constructs whereby users can group together results that have the same value for a specified field. In information retrieval, the study of algorithms that cluster together documents with statistically similar text is an important area of both algorithmic and conceptual model research (23, 11).

Figure 5

Second, we look at what it means to have a *recursive* card catalog. Not only does **REACH** construct virtual card catalogs for the initial collection, but it also does so for the subcollections upon which the user later focuses. For example, the results of a query are treated as a subcollection and are recursively organized into a virtual card catalog with multiple dimensions, as illustrated in Figure 6. The value of recursive organization has been demonstrated by Scatter/Gather, an information retrieval system which statistically clusters document sets in a recursive fashion (24).

Figure 6

Active card catalogs: Making the virtual card catalog *active* in order to support hybrid finding strategies requires adding another concept to the traditional card catalog model. **REACH** allows user-selected "cover-cards" to serve as partial descriptions of information that the user considers valuable. **REACH** can use these partial descriptions to query current repositories for additional information or even to query repositories that were not in the user's initial selection. This new information is then incorporated into the virtual card catalog. Figure 7 gives a high level view of such an action. The concept of an "active" card catalog also has roots in the field of information retrieval. Specifically, relevance feedback is a mechanism whereby users can ask for more results that have statistically similar texts. In our terms, relevance feedback allows sets of text to be treated as partial descriptions of desired information. **REACH** extends this idea to multiple dimensions. As an example, imagine a user who issues a keyword query “English playwright” to a set of repositories; browses through the virtual card catalog containing the results of that query; becomes interested in several author “cover-cards” that appear in the author section (perhaps Shakespeare, Jonson, and Hook); and then asks **REACH** to add more cards to those bundles of interest by doing a search based on the “cover-card” information. In this way, searching and browsing are smoothly integrated into **REACH**. This combination of a virtual and active card catalog opens up new possibilities for an integrated and fluid information-finding process. In the next section, we look at an example of a tool which embodies this new model.

Figure 7

3.3. SenseMaker: a prototype information-finding tool

SenseMaker is a prototype information-finding tool based on the **REACH** conceptual model. SenseMaker can mediate between the user and any of the information repositories that the **InfoBus** protocol makes accessible.

A SenseMaker user begins by defining the overall collection of interest by selecting a set of individual repositories. At the same time, the user specifies a query over that collection using a uniform front end query language. The **InfoBus** is responsible for translating the user query into its native equivalents (25), sending the native queries to the respective repositories, and managing the results returned from each repository.

SenseMaker takes the results and creates a virtual card catalog for them. The sections of the card catalog are determined on the fly by appealing to a hierarchical interlingua schema, as described earlier. Next, the SenseMaker user decides which section of the card catalog to view first and how the cards in that section should be bundled. For example, a user might choose to see the results organized by title and to have items with similar titles bundled together. In the current interface, users view the virtual card catalog section as a table in which each row corresponds to a card or “cover-card” and each column corresponds to a field in the common schema. Figure 8 shows an example of this high-level dimension-specific display of results.

From this initial organization, the user can select bundles of interest (by checking the boxes in the first column) and ask to see them organized according to different dimensions. In this way, the user can survey results at a high level, learn what dimension values characterize the results well, and use dimension values to direct the interaction. Figure 9 shows a SenseMaker display after a few iterations by the user. Currently, SenseMaker bundles are not active. In other words, the user cannot yet use a bundle as the basis for bringing in more results from the current repositories or for bringing in new results from untapped repositories. However, work is underway on SenseMaker II, which will incorporate the active aspect of the **REACH** model.

Figure 8

Figure 9

4.0 The DLITE Project

Most people do not use libraries for the thrill of the hunt, but they access information as part of a larger task. In this section, we describe our **D**igital **L**ibrary **I**ntegrated **T**ask **E**nvironment (**DLITE**) project, which is designed to support the broader concerns of digital library users. A task is a goal-based set of activities like monitoring a company's performance over time or doing background research before buying a color printer. Digital library tasks are bigger than individual searches.

4.1. Support user tasks

We support tasks by providing users with workcenters which contain resources appropriate to the task at hand and visually indicate the state of the current task. A kitchen provides a good real-world example of a workcenter. The tools for baking a cake are all ready-to-hand inside the kitchen, and the task is completed in that space. Workcenters in **DLITE** will contain the tools appropriate for the tasks that users have.

Workcenters in homes are effective because over hundreds of years we have evolved an appropriate set that is large enough to distribute our tools effectively, but not so large that the number induces a cognitive load. One challenge of this work is to come up with a similar set of workcenters for libraries. O'Day and Jeffries studied library users and found that search tasks fall into three categories: monitoring, following a plan, and exploring (25). These categories suggest initial workcenters for searching.

But there is more to digital library tasks than just search (Figure 10), as Paepcke's work has shown (6). A workcenter for following a search plan also needs tools for interpreting search results, managing retrieved documents, and sharing new insights. **DLITE** allows workcenters to include tools that support these other aspects of workers' tasks. This perspective is echoed by O'Day: "It was the accumulation of search results, not the final search result set, that had value for most of our library clients. When people finished searching, they often created summaries of the material they had found, including both overviews and detailed views and analyses." She continues: "A record of an entire interconnected search thread, comprised of both requests and results, should be saved by the system in such a way that it can be deactivated (stored persistently) and activated as the search dies down and then picks up again."

Figure 10

Workcenters in **DLITE** support the accumulation of search results. They can be easily replicated, unlike real-world workcenters, so users do not need to clean up **DLITE** workcenters. A user could have a dozen search tasks in progress at once, in twelve different copies of the same workcenter. Six months after the user has bought her color printer, she would still have the results, tools, and techniques around to pass on to a colleague who had a similar task to do.

A user's task corresponds to an instance of a workcenter. In **DLITE**, a workcenter instance contains components, which fall into one of five categories:

- *Queries* are source-independent expressions of what the user is looking for. Query translation is done automatically when a query is passed to a search service.
- *Documents* are information entities, ranging from encyclopedia entries to books to videos. Different types of documents have different attributes.

- *Collections* allow multiple objects to be manipulated as a group. The most common case is a collection of documents, but there can be collections of queries, collections of services, or heterogeneous collections.
- *Services* can be thought of as functions whose inputs and outputs may be other digital library objects. A search service takes a query as input and returns a collection of documents. Document processing services such as translation take documents as input and return other documents. Services can even take other services as input, as in the case of a Multi-search service that takes a set of search services and a query and returns a collection.
- Representations of *people* are included in the interface to support collaboration, including to indicate who else is “in” a workcenter, to express limits on access permissions, and otherwise enforce intellectual property contracts (26).

4.2. Support a variety of services

Digital library services operate on many different time scales. A service that computes the reading level of a document could take less than a second, while a service that translates a document from English to Japanese (with manual correction) might take days. A service that notifies people when their name appears in the press might persist for years. This observation has implications for the user interface design. The **DLITE** interface allows users to invoke a service and do other work while the service is processing. The user can check the status of the service, and terminate it if its results are no longer needed. The interface also supports persistence across logouts, allowing the user to leave a workcenter running for days or weeks at a time.

The number of services accessible over the Internet is constantly growing. We expect the number to explode when payment mechanisms are widely available and commonly used. **DLITE** is designed to allow new service providers to make their services available to users without requiring extensive software upgrades. We are working on tools to make it easy for service providers to add components to **DLITE**.

4.3. Example: Monitoring a company

Consider a Xerox researcher who wants to keep up on her company and clip articles mentioning her group (Figure 11). She subscribes to a standing order service which adds a few articles to a collection each day. When many articles are available, she uses a tool like SenseMaker to help understand and filter them. She sends potentially interesting articles to a summarization service to decide whether or not to read them further. She drags the articles that are worthy of “clipping” into local collection, and then gives the entire “scrap book” to a bibliography-creation service. Finally, she drops the bibliography document onto a publication service, which makes the collection accessible and announces it over normal company channels.

A professor preparing for a course could use a very similar workcenter, but would instantiate it with very different materials. He would use a different publication service that makes his course bibliography available to colleagues at other universities. He might add a specialized component, provided by the campus bookstore, that takes a collection of documents as input and produces a form that the professor can fill in to cause (paper) course readers to be printed and sold to students (a common student bookstore function).

Figure 11

4.4. Example: Following a plan

Another class of searches mentioned by O’Day and Jeffries are those in which the user is following a plan. As an example, consider someone preparing to buy a color printer. A workcenter for “buying computer peripherals” would contain services relevant to this task, and its visual layout would suggest the plan of action. Here we would find a tool to construct queries that would be seeded with the knowledge that certain databases have a field describing the article type, and that “evaluation” should be the value in this field of the query. Similarly, a good query in this domain would ask for only recent articles.

Next, the workcenter would have likely information sources available. In this example, Dialog database 275 for trade articles would be one such source. Further in the task, databases that find good prices on computer peripherals (e.g. PriceWeb), that provide information on vendors (e.g. The Better Business Bureau), or that help organize lists of features might be appropriate. These tools would all be available from the workcenter.

4.5. Status of DLITE

The examples above have suggested some of the ways that **DLITE** will be used. The current prototype of the system is written using the Tk toolkit, and is available over X Windows. We have nearly completed a Java implementation of the interface as well, and plan to deploy it over the WWW via Java-enabled browsers.

5.0 Conclusion

In this paper, we have examined some of the many forms of heterogeneity that are inherent to networked information environments. We have described three Stanford University Digital Library projects which are tackling this issue of heterogeneity from various perspectives.

The **InfoBus** project addresses heterogeneity from a protocol perspective. It defines both an architecture and two protocols, the InterOp and InterPay protocols. Through these developments, the **InfoBus** project provides network programmers with a uniform, high-level, object-oriented interface to a plethora of different services and payment mechanisms.

The **REACH** project focuses on information repository heterogeneity from a user conceptual model perspective. It sets forth the concept of a virtual, active card catalog. Users can compare information from heterogeneous repositories because the virtual card catalog provides a uniform structure over the information items. Furthermore, the **REACH** model allows users to see the information according to multiple dimensions and to employ hybrid browsing/searching strategies.

Finally, the **DLITE** project addresses heterogeneity in users' tasks and in the information services they access, also from a user conceptual model perspective. It presents the concept of a workcenter, a place that gathers together task-specific resources. Through the modeling of workcenters and workcenter components, **DLITE** provides users with a task-oriented interface to services of varying time scales and complexity.

As we have seen, each of these projects solves a different piece of the heterogeneity puzzle. Yet these pieces do not exist in isolation. Both **REACH** and **DLITE** rely upon the infrastructure provided by the **InfoBus** project. Plus, the **DLITE** interface can integrate the **REACH** conceptual model by incorporating SenseMaker as a new type of information service. As work on heterogeneity progresses in the Stanford University Digital Library Project and elsewhere, we expect to fit together still more pieces to our heterogeneity puzzle.

6.0 Acknowledgments

This work is supported by the NSF under Cooperative Agreement IRI-9411306. Funding for this cooperative agreement is also provided by ARPA, NASA, and the industrial partners of the Stanford Digital Library Project. Terry Winograd and Andreas Paepcke both read early drafts of this paper and gave us valuable comments.

7.0 References

1. PAEPCKE, A. Digital libraries is not enough: what we learned on site. D-lib Magazine, May 1996. <http://www.dlib.org>
2. LEVY, D. and MARSHALL, C. Going digital: a look at assumptions underlying digital libraries. Communications of the ACM, 38 (4), April 1995, 77-84.
3. THE STANFORD DIGITAL LIBRARIES GROUP. The Stanford Digital Library Project. Communications of the ACM, 38 (4), April 1995, 59-60.
4. LYNCH, C. Networked information resource discovery: an overview of current issues. IEEE Journal on Selected Areas in Communications, 13 (8), October 1995, 1505-22.
5. NARDI, B. and O’DAY, V. Intelligent agents: what we learned at the library. Libri, 46 (2), June 1996.
6. PAEPCKE, A. Information needs in technical work settings and their implications for the design of computer tools. CSCW Journal, 5 (1), July 1996.

7. PUTZ, S. Design and implementation of the system 33 document service. Xerox Palo Alto Research Center, 1993 (ISTL Tech Report P93-00112).
8. RAO, R., RUSSELL, D., and MACKINLAY, J. System components for embedded information retrieval from multiple disparate information sources. In: Proceedings of the ACM Symposium on User Interface Software and Technology, ACM Press, November 1993.
9. Information Retrieval: Application Service Definition and Protocol Specification. ANSI/NISO, Bethesda, Md., 1995.
10. BERNERS-LEE, T., FIELDING, R., and FRYSTYK, H. Hypertext Transfer Protocol -- HTTP 1.0. RFC 1945, May 1996. <ftp://ds.internic.net/rfc/rfc1945.txt>
11. MARCHIONINI, G. Information Seeking in Electronic Environments. Cambridge: Cambridge University Press, 1995.
12. WYNAR, B. and TAYLOR, A. Introduction to Cataloging and Classification. Englewood, Co.: Libraries Unlimited, Inc., 1992.
13. SAWYER, P. and MARIANI, J. Database systems: challenges and opportunities for graphical HCI. Interacting with Computers, 7 (3), 1995, 273-303.
14. LIDDLE, D. Design of the conceptual model. In: T. Winograd, ed. Bringing Design to Software. New York: ACM Press, 1996.
15. WINOGRAD, T. and FLORES, F. Understanding Computers and Cognition: A New Foundation for Design. Reading, Mass.: Addison-Wesley Publishing Company, Inc., 1987.
16. FURNAS, G. Generalized fisheye views. In: Proceedings of CHI ’86, 16-23.
17. ROBERTSON, G., CARD, S., and MACKINLAY, J. Information visualization using 3D interactive animation. Communications of the ACM, 36 (4), 1993, 56-71.
18. PAEPCKE, A., COUSINS, S., GARCIA-MOLINA, H., HASSAN, S., KETCHPEL, S., RÖSCHEISEN, M. and WINOGRAD, T. Using distributed objects for digital library interoperability. IEEE Computer Magazine, 29 (5), May 1996, 61-68.
19. YANG, Z. and DUDDY, K. CORBA: a platform for distributed object computing. Operating Systems Review, 30 (2), April 1996, 4-31.
20. CUTTING, D., JANSSEN, W., SPREITZER, M., and WYMORE, F. ILU Reference Manual. Xerox Palo Alto Research Center, December 1993. <http://www.xerox.com/PARC/ilu/index.html>
21. MELTON, J. and SIMON, R. Understanding the new SQL: a complete guide. San Mateo, Ca: Morgan Kaufmann Publishers, 1993.
22. FRAKES, W. and BAEZA-YATES, R., eds. Information Retrieval: Data Structures and Algorithms. Englewood Cliffs, NJ: P T R Prentice-Hall, Inc., 1992.
23. CUTTING, D., KARGER, D., PEDERSEN, J. and TUKEY, J. Scatter/Gather: a cluster-based approach to browsing large document collections. In: SIGIR ’92, 318-29.
24. CHANG, K., GARCIA-MOLINA, H., and PAEPCKE, A. Boolean query mapping across heterogeneous information sources. In: IEEE Transactions on Knowledge and Database Engineering, 1996.

25. O’DAY, V.L. and JEFFRIES, R. Orienteering in an information landscape: how information seekers get from here to there. In INTERCHI '93, 438-45
26. RÖSCHEISEN, R. and WINOGRAD, T. A Communication Agreement Framework for Access/Action Control. In: Proceedings of the IEEE Symposium on Research in Security and Privacy, 1996.

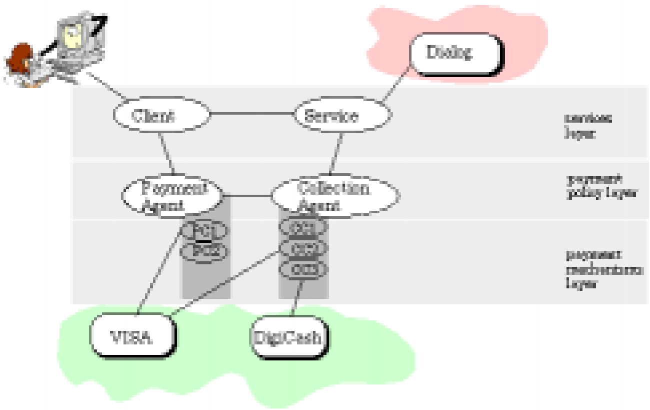


Figure 1. Interpay runs “under the hood” to abstract payment details from the user and to provide a single interface to many payment mechanisms

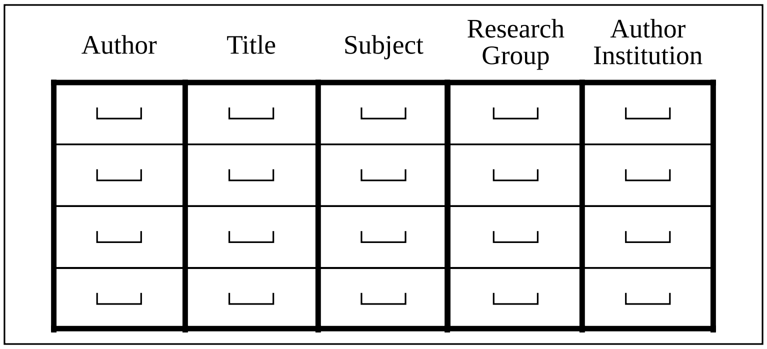


Figure 2. Virtual card catalog for a research collection

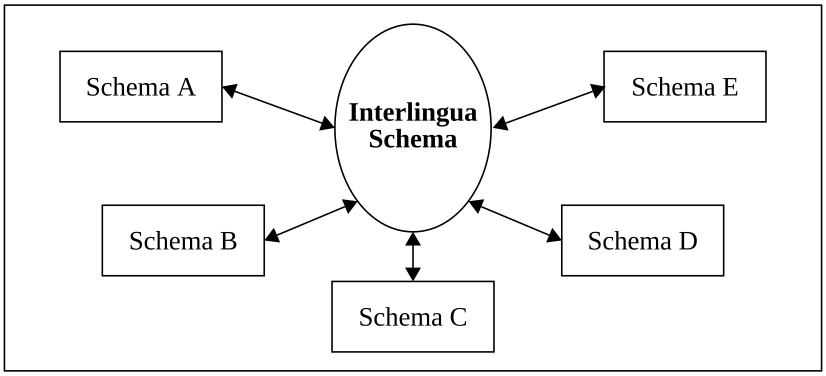


Figure 3. Interlingua schema

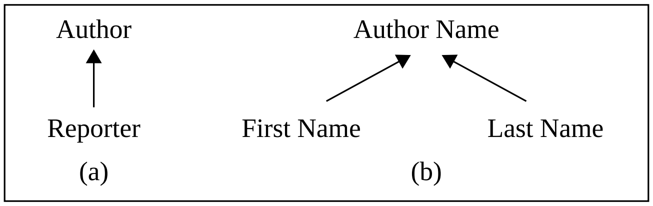


Figure 4. (a) Specialization relationship; (b) Composition relationship

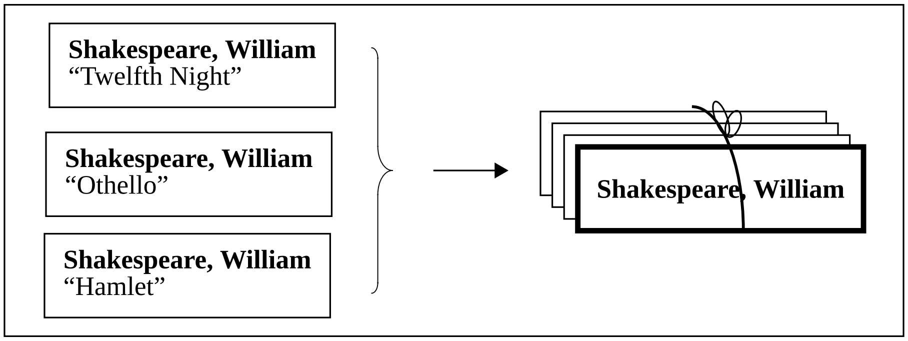


Figure 5. Bundling cards with the same author value

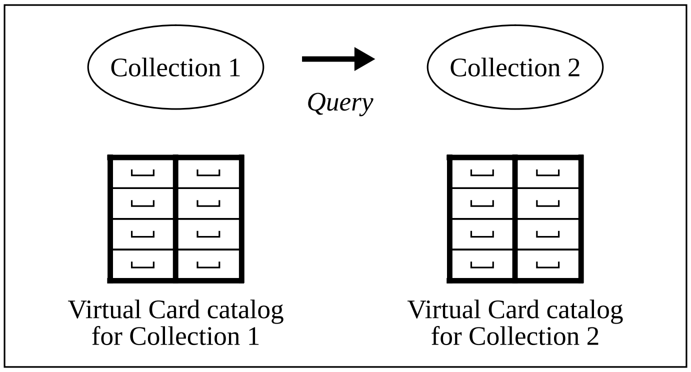


Figure 6. Recursiveness in the virtual card catalog

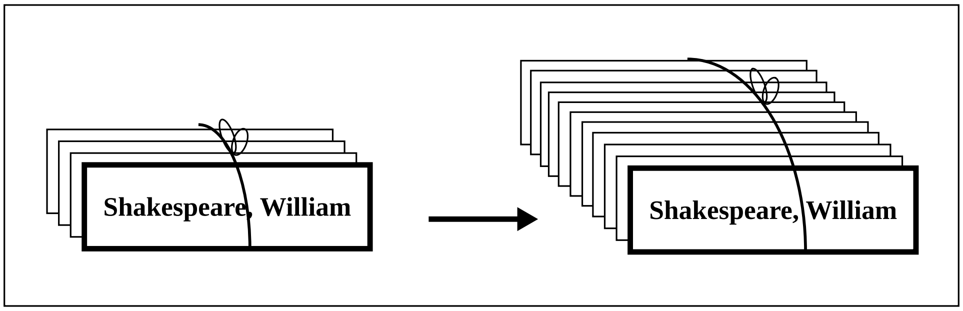
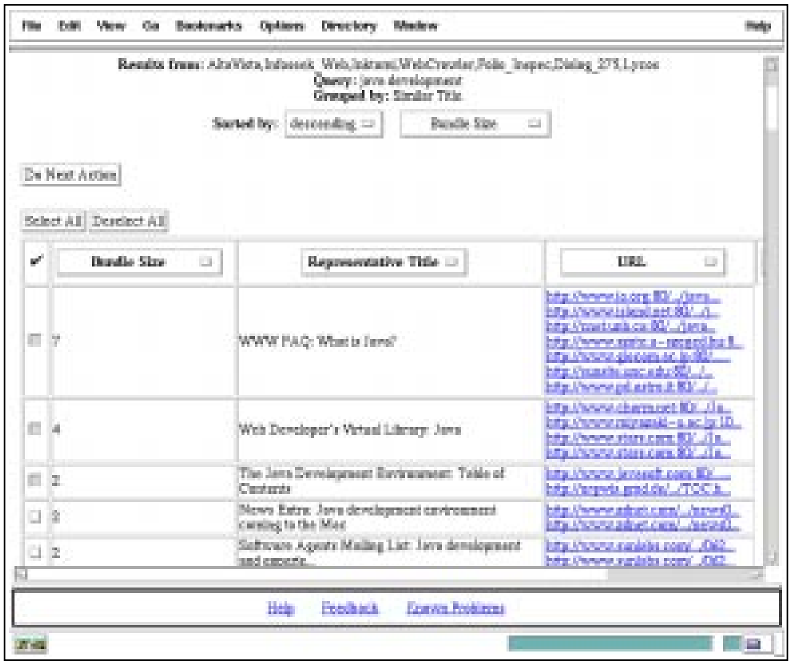


Figure 7. Using a “cover-card” to ask for more information

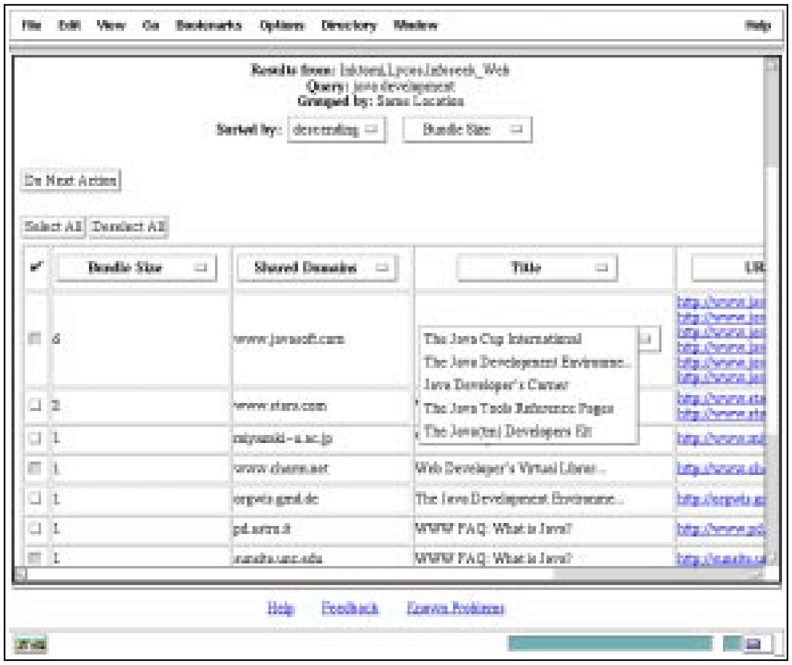


Results from: AltaVista,Infoseek_Web,Inktomi,WebCrawler,Folio_Inspec,Dialog_275,Lycos
Query: java development
Grouped by: Similar Title

Sorted by: descending Bundle Size

✓	Bundle Size	Representative Title	URL
☑	7	WWW FAQ: What is Java?	[illegible]
☑	4	Web Developer's Virtual Library: Java	[illegible]
☑	2	The Java Development Environment: Table of Contents	[illegible]
☐	2	News Extra: Java development environment coming to the Mac	[illegible]
☐	2	Software Agents Mailing List: Java development and experts...	[illegible]

Figure 8. SenseMaker initial display



Results from: Inktomi,Lycos,Infoseek_Web
Query: java development
Grouped by: Same Location

Sorted by: descending Bundle Size

✔	Bundle Size	Shared Domains	Title	UR
☑	6	www.javasoft.com	The Java Cup International The Java Development Environme... Java Developer's Corner The Java Tools Reference Pages The Java(tm) Developers Kit	http://www.ja http://www.ja http://www.ja http://www.ja http://www.ja http://www.ja
☐	2	www.stars.com		http://www.sta http://www.sta
☐	1	miyazaki-u.ac.jp		http://www.mi
☑	1	www.charm.net	Web Developer's Virtual Librar...	http://www.ch
☐	1	orgwis.gmd.de	The Java Development Environme...	http://orgwis.g
☐	1	pd.astro.it	WWW FAQ: What is Java?	http://www.pd
☑	1	sunsite.unc.edu	WWW FAQ: What is Java?	http://sunsite.u

Figure 9. SenseMaker display after a few iterations

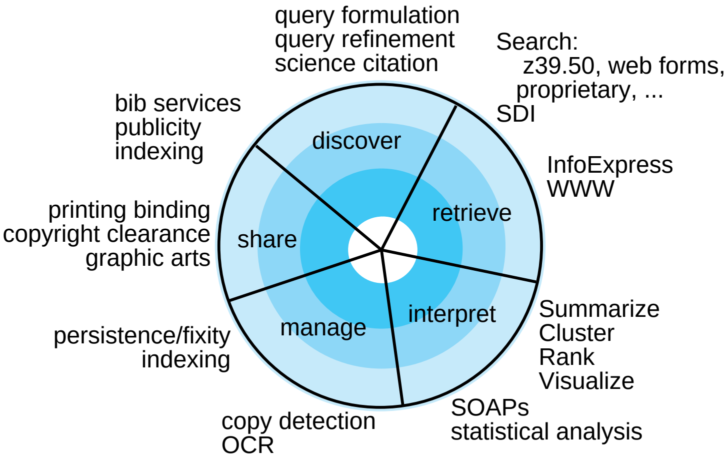
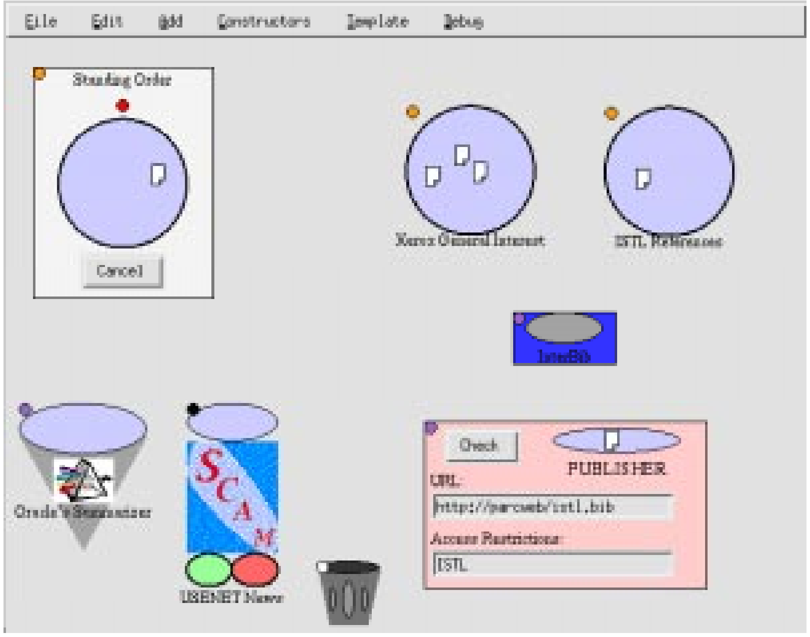


Figure 10. There is more to digital libraries than just search. This wheel depicts five components of an information management task, along with sample digital library services corresponding to each component.



URL: http://parcweb/istl.bib

Access Restrictions: ISTL

Figure 11. A screen dump taken from the DLITE interface, showing a workspace with tools from the "Monitoring tasks" example. Documents arrive in the standing orders collection, and can be dragged to the summarizer or copy detector (SCAM) for processing. They can be dropped into one of the collections, and the collections can be processed using the InterBib bibliography-generation service. Finally, the bibliography document can be made available by dropping it onto the publisher service.