Myriad: An Architecture for Contextualized Information Retrieval and Delivery

Cécile Paris[1], Mingfang Wu[1], Keith Vander Linden[2], Matthew Post[2], Shijian Lu[1]

[1] CSIRO ICT Center,
NSW, Sydney, Australia
{Cecile.Paris, Mingfang.Wu, Shijian.Lu}@csiro.au
[2] Department of Computer Science, Calvin College,
Grand Rapids, MI USA
{kvlinden, mpost89}@calvin.edu

Abstract. Users' information needs are largely driven by the context in which they make their decisions. This context is dynamic. It includes the users' characteristics, their current domain of application, the tasks they commonly perform and the device they are currently using. This context is also evolving. When one information need is satisfied, another is likely to emerge. An information access system must, therefore, be able to track this dynamic and evolving context, and exploit it to retrieve actionable information from appropriate sources and deliver it in a form suitable for the current situation. This paper presents a generic architecture that supports the construction of information retrieval and delivery systems that make use of context. The architecture, called Myriad, includes an adaptive virtual document planner, and explicit, dynamic representations of the user's current context.

1 Introduction

Knowledge workers are increasingly dependent upon the availability of the right information in the right form for their daily work. Their information needs are largely driven by the context in which they make their decisions. This context is dynamic. It includes their characteristics, their current domain of application, the tasks they commonly perform and the device they are currently using. This context is also evolving. When one information need is satisfied, another is likely to emerge. An information access system must, therefore, be able to track this dynamic and evolving context, to exploit it to retrieve information from appropriate sources, and to deliver that information in a form that is appropriately tailored to the user's context.

Thus, the challenges for such an information access system come in both retrieval and delivery. First, the system must interact with heterogeneous databases and document collections, so a unified interface or integrated framework for accessing these systems is essential. Second, the system must deliver the information retrieved from these various sources in a manner that is adapted and reorganized to suit the user's needs at that time.

This paper presents a generic architecture that supports the construction of an information access system for contextualized guided information retrieval and delivery. The architecture, called Myriad, includes an adaptive virtual document planner, and a set of models for explicit and dynamic representation of the user's current context. This architecture extends our previous work on tailored information delivery [17,18] by adding support for adaptation based on the users' tasks and interaction history.

2 Related Work

Delivering appropriate information in an appropriate form and adapted to an individual user's need is a key theme in the adaptive hypermedia community [4]. The aim of adaptive hypermedia systems is to provide a personalized information space for users to browse. This personalized access is achieved through content level adaptation (or adaptive presentation) and link-level adaptation (or adaptive navigation support) [3]. The adaptive presentation selects and presents information according to users' goal and knowledge backgrounds, while the link level adaptation provides navigation support to help users orient themselves within the hypermedia space.

Adaptive hypermedia systems adapt information to a user's needs by building an explicit user model and exploiting it throughout the interaction with the user, e.g., [9, 10, 12]. A user model may include users' search goals, knowledge of the application domain, and some personal characteristics such as preference and background. Traditional adaptive hypermedia systems assume the existence of a closed set of documents that are authored by humans and manually linked in a hyper-information space. A user model is usually implemented as an overlay of an underlying concept model. Each concept is associated with a value that indicates a user's current state of knowledge with respect to that concept. An adaptive engine, therefore, can simply compare the value of a concept in its user model with the conditions of a concept in its conceptual model in order to filter out a sub-hypermedia space that suits a user's goal and knowledge level. One of the assumptions behind these adaptive hypermedia systems is that the system designers/authors know the documents in a collection and have a global map of how the hypermedia space is to be presented to the user [7]. This assumption makes it hard for adaptive hypermedia systems to be scaled to large and open document collections. Adaptive hypermedia systems are hard, therefore, to deploy in the web environment because information access system designers cannot know in advance every document in the collection. The web and even some organizations' internal web sites are simply too large and too open. Although documents could, in principle, be annotated with meta-data, many are not.

In the Information Retrieval (IR) community, the user model tends to be implicit – the user model is simply a query. The advantage of this model is that it is very simple to implement and can be scaled to any open document collection; this model has been successful in retrieving documents from large document collections. However, this model has two major disadvantages. First, the resulting IR systems simply deliver the documents that are retrieved, without any contextual adaptation. Thus different users typing the same query will get the same set of responses. While this kind of delivery

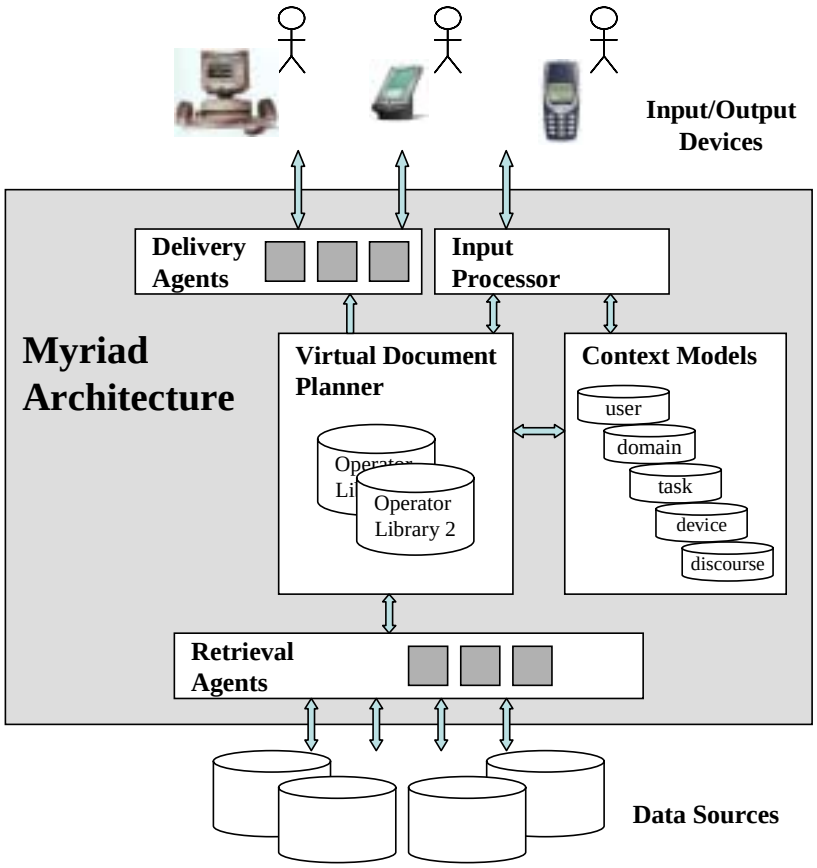

Fig. 1. The Myriad Architecture

gives users access to information, it doesn't concern itself with user understanding. Second, this query based model does not consider the user's evolving search context; every query is treated as a one-off retrieval task that has no connection to the previous queries that result from the same information need. Consequently, the search result is discrete and not presented as a coherent whole.

The proposed Myriad architecture attempts to make the best use of the advantages of both the delivery-driven approach of adaptive hypermedia systems and the retrieval-driven approach of information retrieval systems. This enables Myriad to support both the retrieval of information from open heterogeneous databases and the delivery of that information within the users' information-seeking context.

3 An Adaptive Hypermedia Architecture

The Myriad architecture supports the construction of information access systems that provide information retrieval from heterogeneous databases and adaptive hypermedia delivery. Fig. 1 shows the basic components of this architecture. The core of this architecture is the Virtual Document Planner (VDP). This planner receives information requests from the users via the input processor and uses the contents of various context models to drive its operation. It deploys retrieval agents to select contents appropriate to the users' current information need and context, and delivery agents to present this content with appropriate links in a coherent way. A number of information application systems have been, and are currently being built using the Myriad architecture:

- **Tiddler** – a tailored information delivery system. It used the document planner on which the Myriad VDP is based to produce personalized travel information [18]. Tiddler combines structured information from databases with text from existing web pages, delivering this information in a form appropriate for the user's device.
- **PERCY** – an application in the domain of corporate knowledge memory. It generates a coherent e-brochure about a corporation, personalized to the user's query and user type [17]. This application exploits a user profile, a set of discourse roles and a set of web data resources. Given a user query, the application system uses the discourse roles and the user model to decide which information to present and how to present it in a coherent way.
- **DFDMSA** – a new system being developed to mediate the user interface in an aviation application. This application aims to support operators and maximize their efficiency by automatically providing them with information relevant to their tasks at the time. The system provides a task-based interaction environment that delivers tailored multimedia information from the analysis and recognition of the operator's activity [5, 6].
- **Skil** – a new system being developed to provide a computer assisted training environment in a medical domain. This application integrates recent advances in the area of haptic virtual environments and natural language generation. It aims to reduce the complexity of a three dimensional scene in order to allow reasoning about the user's actions and to deliver tailored information.
- **Tiger** – a new system being developed to provide actionable information in the context of a organization. Here, actionable information is information that has been retrieved, processed, analyzed, and synthesized in context for a person to consider and then act upon

This section discusses the basic components of the Myriad architecture, using the current Tiger prototype system as an example. Fig. 2 shows a sample output of this system. Here, we see a web-based presentation of information requested by a professor about a Calvin student. The information includes data that is private to the professor and other data that is more generally accessible.

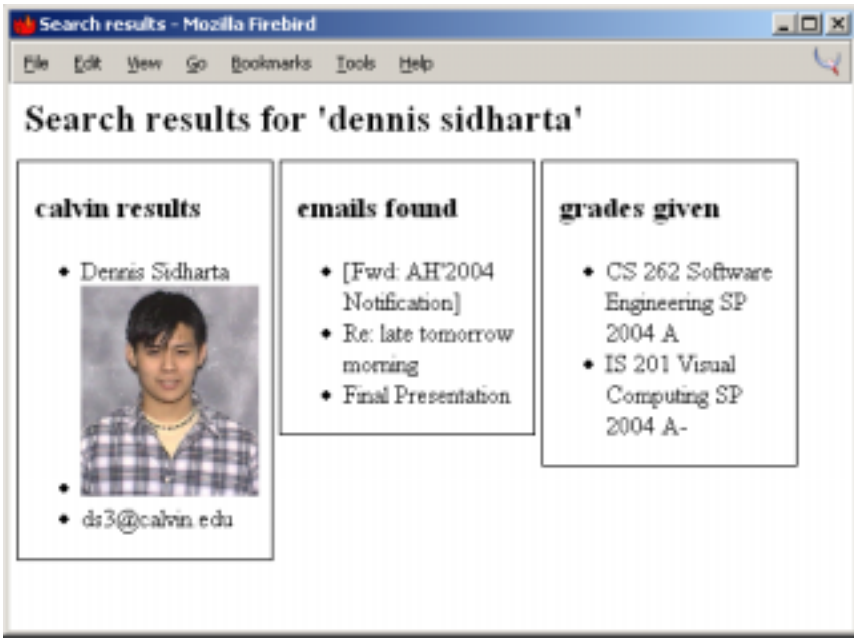

Fig. 2. Sample Tiger Output

3.1 Input Processor

The input processor is responsible for receiving input from the user, identifying the information need and posting it to the VDP as the communicative goal. It assumes that the user has been authenticated and that their input device determined. Any input received from the user, whether it be through a typed or spoken query or a hyperlink, will be processed by the input processor and stored in the user's discourse history.

The Myriad architecture currently uses a finite state parser as the key technology in the input processor [13]. This parser uses an augmented transition language to translate natural language queries from the user into goal statements appropriate as input to the VDP. Different information application systems can specify different input parsing grammars depending upon the sorts of questions they are likely to receive from users.

For the Tiger example in Fig. 2, the input processor received the query:

```
Tell me about Dennis Sidharta
```

and translated it into the instantiated communicative goal:

```
DESCRIBE <Dennis Sidharta> TO <current user>
```

This goal is then passed to the VDP to drive the information retrieval and delivery process.

3.2 Virtual Document Planner

The VDP is the central component of the Myriad architecture. Its implementation is based on Moore and Paris's [16] generic planning mechanism and the Tiddler document planner [18], both of which used plan operators to drive both the selection of appropriate content and the construction of an appropriate presentation of that content. This sort of planner has frequently been used in the language engineering community [16, 19] to drive information presentation. Its resources are represented as libraries of declarative plan operators. Myriad supports the construction of multiple plan operator libraries for each information application.

For example, the Tiddler system included two plan operator libraries: discourse operators, which built a content plan (structured in terms of Rhetorical Structure Theory [15]), and presentation operators, which structured it in a manner appropriate for the user's current context.]. The Skil prototype adds a third library of interaction operators, which specifies interaction goals for training lessons.

In the Tiger example, the communicative goal passed to the VDP (see above), matches the following discourse plan operator (specified in XML):

```
<operator>
  …
  <effect>(describe ?person to ?user)</effect>
  <constraint>(set ?grading (retrieval:JDBCAgent ?per-
son))</constraint>
  <constraint>(set ?file "results.html")</constraint>
  …
  <satellite>
    <type>optional</type><relation>elaboration</relation>
    <value>(CalvinPersonAgent ?file ?person)</value>
  </satellite>
  <satellite>
    <type>optional</type><relation>elaboration</relation>
    <value>(IMAPAgent ?file ?person)</value>
  </satellite>
  <nucleus>
    <value>(delivery:htmllister ?file "Grades given" (re-
trieval:JDBCAgent ?person))</value>
  </nucleus>
  …
</operator>
```

Here, we see that the `effect` of the operator can be unified with the communicative goal (i.e., if we bind `?person` to <dennis sidharta> and `?user` to <current user>). This operator is constrained to work only when the JDBC retrieval agent can find grades for the student in the user's database (see the first `constraint`). Otherwise, another plan operator must be found. The `nucleus` and two `satellite`s insert the "calvin results", the "emails" and the "grades given", as shown in Fig. 2.

3.3 Context Models

Myriad's approach to adaptivity is to capture the user's context by monitoring their interactions with the system and reflecting them in the context models. The five basic context models are discussed here. All of the models are represented using a KL-ONE styled representation scheme [2] and can be configured for each new information application as appropriate:

- **User Model** – The user model represents the user's long-term (or stereotypical) profile and their short-term information goal. The long-term profile is application dependent, but could include a user's job description or the user's role in an organization. The short-term goal is a user's current information need, which is usually specified in the user's query. For example, Skil relies on the user model to dictate the varying levels of explanation of a particular concept that are required by novices and by experienced users, while PERCY and Tiddler rely on the user's profile and interests to direct the content of their presentations.
- **Domain Model** – The domain model consists of hierarchies of concepts, properties and relationships relevant within the application domain. Myriad does not attempt to develop a definitive ontological model, but rather allows application developers to specify the model that is relevant for them. The Tiger model, for example, distinguishes between people, companies, and other conscious entities.

  Unlike typical adaptive hypermedia systems, Myriad does not require its domain model concepts be related to documents in the knowledge base. The binding between documents and concept instances could happen just as well at retrieval time.
- **Task Model** – The task model is used to describe the tasks commonly performed by users of the application. For each user task, the model specifies the relevant sub-steps, conditions and effects. These tasks are specified using the Diane+ task modeling formalism [20]. For example, DFDMSA bases it retrieval and presentation on detailed models of its users' tasks.
- **Device Model** – The device model specifies the characteristics of a user's current input/output device. In Tiddler, for example, different user devices, such as cell phones or desktop machines, lead to the use of different VDP plan operators and, thus to different output content, structure and presentation.
- **Discourse Model** – The discourse model keeps track of the user's information searching history. These histories may contain the questions that the user has posed to the system and the documents that the user opens for each question. This model can be used to predicate the user's future interest, as in [11, 13], and to generate a search summary toward the end of the user's search session.

Taken together, these models represent the context of the user's information need. Generally, the relevant context elements from the domain, task, device and discourse models are linked from the user's instance in the user model. The domain model is primarily used to define a set of concepts (and their relationships) used by the other models and by the VDP. In the Tiger example, part of the user model entry for person asking for information is as follows (again, specified in XML):

```
<instance>
  …
  <type>Professor</type>
  <link>
    <slot>name</slot>
    <value>Keith Vander Linden</value>
  </link>
  <link>
    <slot>email</slot>
    <value>kvlinden@calvin.edu</value>
  </link>
  <link>
    <slot>imapagent:hostname</slot>
    <value>mailhost.calvin.edu</value>
  </link>
  <link>
    <slot>imapagent:username</slot>
    <value>kvlinden</value>
  </link>
  …
</instance>
```

Here, the user is a `professor`, as indicated by the `type` specification. The user's name is given, as is his email address and his IMAP mail hostname and username. Addition information (not shown) is specified for how to query his personal grading database and how to access his personal account on the Calvin information server.

3.4 Retrieval Agents and Information Access Tools

Retrieval agents receive a specification from the VDP and then use appropriate information access tools and information sources to find information according to the specification. For the Tiger application, we have implemented a number of retrieval agents within the Myriad architecture. They include:

- **GoogleAgent** – This agent queries the Google search engine for information specified by the VDP. It is able to take advantage of any of the features exposed by Google's search API and to return those results to the VDP.
- **IMAPAgent** – This agent provides a generalized mechanism for searching the current user's IMAP mail account. In the Tiger example, this agent retrieved the emails shown in Fig. 2.
- **JDBCAgent** – This agent provides a generalized mechanism for querying databases via JDBC. Users can customize this agent to query either personal or public database sources for any type of information to which they have access. In the Tiger example, this agent retrieved the student grades shown in Fig. 2.

- **LocalPersonAgent** – This agent accesses a web form to search a local database of students, faculty, and staff, and returns that person's picture, email address, position, department, and phone number. In the Tiger example, this agent retrieved the "Calvin Results" shown in Fig. 2.

Here, we can see that retrieval agents range in degree of scope, from the very general (e.g., the GoogleAgent) to the very localized (e.g., the LocalPersonAgent). This allows application developers to use Myriad to support tools that provide access to heterogeneous data sources, retrieving private data when available, and reverting to more public data when necessary. In the Tiger example, for instance, if the grading agent had failed to find grades for the given person, then the plan operator discussed above would have failed and another more general operator would have been run unstead. The more general operator would have called the GoogleAgent.

3.5 Delivery Agents

Delivery agents take content from the retrieval agents and present that content in the appropriate manner as directed by the VDP. For example, a delivery agent could organize the retrieved information and present it in a manner that suits a user's current interaction platform. Another delivery agent could take a set of retrieved documents and call a clustering function to group the documents into topic related clusters [22]. In the Tiger example, the output was delivered in HTML form to a web browser.

4 Discussion and Future Work

The Myriad architecture has been evolving from our earlier applications (Tiddler and PERCY) that supported user, domain and device models, to applications supporting task models (DFDMSA), and discourse models (Skil and Tiger). The adaptation is also evolving from adaptability, based on a profile including user preferences and device capabilities, to adaptivity, based on not only user profile but also the domain model, task model and discourse model.

Future work on the Myriad architecture primarily centers on further development of the current Myriad applications and on the features Myriad provides to support this development. In particular, a greater variety of retrieval and delivery agents are required as well as a more general set of VDP plan operator libraries that can more easily be adapted to new information application domains.

References

1. Aroyo, L., De Bra, P. and Houben, G. J. Embedding Information Retrieval in Adaptive-Hypermedia: IR meets AHA! In Proceedings of the AH2003 Workshop, pp.63-76, Budapest, Hungary, May 2003.

2. Brachman, R. J., & Schmolze, J. An Overview of the KL-ONE Knowledge Representation System, Cognitive Science, 9, 171-216, 1985.
3. Brusilovsky, P. Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction, 6 (2-3), pp. 87-129, 1996.
4. Brusilovsky, P. Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110, 2001.
5. Colineau, N., Lampert, A. and Paris, C. Task-Sensitive User Interfaces: grouping information provision within the context of the user's activity. To appear in Proceedings of Advanced Visual Interfaces (AI 2004), May 25-28, 2004, Gallipoli, Italy.
6. Colineau, N. and Paris, C. Task-Driven Information Presentation. In Proc. of the Annual Conference of the Computer-Human Interaction (OZCHI'03), Special Interest Group of the Ergonomics Society of Australia, Brisbane, Australia, Nov 25-28.
7. De Bra, P. and Calvi, L. AHA! An Open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, 115-139, Taylor Graham Publishers, 1998
8. De Bra, P., Adaptive Hypermedia on the Web: Methods, techniques and applications, Proc. of the AACE WebNet'98 Conference, pp. 220-225, Orlando, Fl., 1998.
9. De Bra, P., Houben, G. J., and Wu, H. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In Proceedings of ACM Hypertext, 1999, pp.115-139.
10. Frasincar, G., Houben, G. J., Vdovjak, R. An RMM-Based Methodology for Hypermedia Presentation Design. In Proc. Advances in Databases and Information Systems, LNCS 2151, Springer (2001), pp.323-337.
11. Gates, K. F., Lawhead, P. B. and Wilkins, D. E., Toward an adaptive WWW: a case study in customized hyperemia. New Review of Multimedia and Hypermedia 4, 89-113.
12. Halasz, F., and Schwartz, M. The Dexter Hypertext Reference Model. CACM, 37(2), 1994, pp.30-39.
13. Joachims, T., Freitag, D. and Mitchell, T., WebWatcher: A Tour Guide for the World Wide Web. Proceedings of 15th International Joint Conference on Artificial Intelligence, IJCAI;97, pp.770-775.
14. Jurafsky, D. and Martin, J. (2000). Speech and Language Processing, Prentice Hall.
15. Mann, W. C. and Thompson, S. A. Rhetorical Structure Theory: toward a Functional Theory of Text Organisation. In Text 8(3): 243-281, 1988.
16. Moore, J. D. and Paris, C. L. Planning Text for Advisory Dialogues: Capturing Internatinal and Rhetorical Information. In Computational Linguistics, Cambridge, MA, Vol 19(4), 651-694, 1993.
17. Paris, C., Wu, M., Vercoustre, A., Wan, S., Wilkins, P. and Wilkinson, R. An Empirical Study of the Effect of Coherent and Tailored Document Delivery as an Interface to Organizational Websites. In The Proceedings of the Adaptive Hypermedia Workshop at the 2003 User Modelling Conference, Pittsburgh, USA, June 22, 2003. pp 133 - 144.
18. Paris, C., Wan, S., Wilkinson, R. and Wu, M. Generating Personal Travel Guides – and who wants them? In Proceedings of the International Conference on User Modelling (UM2001); M. Bauer, P. Gmytrasiewicz and J. Vassileva (eds). Sonthofen, Germany, July 13-18, 2001. pp 251 - 253.
19. Reiter, E. and Dale, R. Building Natural Language Generations Systems. Cambridge University Press 2000.
20. Tarby, J. C. and Barthet, M. F. The Diane+ method. In Computer-Aided Design of User Interfaces, in Proceedings of the Second International Workshop on Computer-Aided Design of User Interfaces (CADUI'96). Namur, Belgium, 5-7 June, 1996.
21. Terveen, L. Overview of Human-Computer Collaboration. Knowledge-based Systems, 8 April-June 1995, pp.67-81
22. Wu, M., Fuller, M. and Wilkinson, R. Using clustering and classification approaches in interactive retrieval. Information Processing and Management, vol. 37, pp.459-484, 2001.