In: Proc. PAAM '96, London, U.K., April 22-24, 1996, pp. 45–59

Constraint-based Information Gathering for a Network Publication System

Uwe M. Borghoff and Remo Pareschi
Rank Xerox Research Centre, Grenoble Laboratory
6, chemin de Maupertuis. F-38240 Meylan, France
Tel.: +33 7661-5076. Fax: +33 7661-5099
E-mail: {borghoff,pareschi}@grenoble.rxrc.xerox.com

Harald Karch
Rank Xerox Germany Systems Operations
Werftstraße 37. D-40549 Düsseldorf (Heerdt), Germany
Tel.: +49 511 6269-38. Fax: +49 211 990-7570
E-mail: karch@braunschweig.netsurf.de

Martina Nöhmeier and Johann H. Schlichter*
Institut für Informatik
Technische Universität München. D-80290 München, Germany
Tel.: +49 89 4505-5220. Fax: +49 89 4505-5222
E-mail: {noehmeie,schlicht}@informatik.tu-muenchen.de

Abstract

The Internet and the World-Wide Web (WWW) are revolutionizing knowledge exchange by linking heterogeneous information repositories into a kind of gigantic world-wide digital library. Yet up until now, knowledge management on the WWW has mainly been provided by navigation tools like Mosaic and Netscape, and by engines like Alta Vista, Lycos and Yahoo which support navigation by automating the search for user-relevant WWW sites. The simplicity of this paradigm has been the key to the initial success of the Web infrastructure but now falls short of more complex applications needed by an ever-growing community of users. Prominent among these needs is flexible information gathering from multiple knowledge sources to ad-hocratically serve the requests of specific user groups. For instance, Network Publication Systems (NPS) for large organizations need flexible integration of enquiry information like Who's Who services and tables of contents of journals with E-print archival material, as well as flexible adaptation of local query services. Agent technology can provide the right answer to these demands. In this paper, we describe agent-based information gathering on the WWW in the context of a NPS for the European Physicist Society. In our approach, we exploit *constraints* to implement information gathering with maximal flexibility.

Key words. Internet, WWW, network publication systems, information gathering, constraints.

* Work performed while visiting the Rank Xerox Research Centre in Grenoble

1 Introduction

In spite of the enormous amount of information available, the World-Wide Web has so far been accessible essentially through simple navigation tools like Mosaic and Netscape. The simplicity of this paradigm has been the key to the initial success of the Web infrastructure but now falls short of more complex applications needed by an ever-growing community of users. Prominent among these needs is flexible information gathering from multiple knowledge sources to ad-hocratically serve the requests of specific user groups. Agent technology is an obvious candidate to fulfill this demand. Through the use of remote programming tools like Java and Telescript, autonomous software agents can access heterogeneous information repositories to select and merge appropriate knowledge to satisfy user requests. In doing so, they can leverage basic yet powerful indexing facilities such as Lycos, Alta Vista, Yahoo and other retrieval engines.

However, aside from such infrastructural support, the issue on how to design and implement agents of this kind remains open. In this paper, we show how *constraints*, a long-known construct from artificial intelligence and computer science, can give the right answer. Constraints have in the past been exploited essentially for the combinatorial optimization of computationally hard problems. These capabilities have been embedded into full-blown programming environments, either of the object-oriented (Freeman-Benson 1990) or of the logic programming (Hentenryck and Saraswat 1995) breed. More recently, constraints have been used to capture partial information in a world of concurrent communicating agents (Henz *et al.* 1995, Saraswat 1989). The possibility of exploiting such a view of concurrency in the context of distributed knowledge management was made explicit by Andreoli *et al.* (1994–1996). For the practical purpose of knowledge management on the Web, the main advantages of using constraints can be summarized as follows:

1. from the users point of view, constraints can be used to flexibly specify *partial requests*, namely requests which may leave underspecified, certain aspects of the requested information.
2. from the point of view of the agent platform, constraints can be used to create *dependencies* among concurrent subrequests into which an initial request is decomposed. Constraints also provide *concurrency control* among the agents managing the queries.
3. for information providers, constraints can be used to dynamically augment local query interfaces by *filtering* result items on a per constraint basis.

There is a large variety and number of multiagent applications for knowledge management on the Web where these capabilities can be exploited: among others, bargain finding, dynamic assemblage of virtual catalogs, data warehousing from backend repositories, agent-based document construction and customization can all be supported through this paradigm. In this paper, we describe a case from the domain of network publication systems (NPS), specifically a system embedded in a European distributed document database for physics which will be extended to other learned fields like mathematics, computer science and chemistry. This project is being developed collaborative between Rank Xerox Germany System Operations, the Grenoble Lab of the Rank Xerox Research Centre and several university departments.

The paper is organized as follows. After a brief discussion of related work in Sect. 2, we present in Sect. 3 the application environment, namely the Physicists Network Publishing System (PNPS). PNPS serves as a testbed for the knowledge broker framework, our agent-based approach for information gathering using constraints. Sect. 4 describes, in detail, the

architectural framework and illustrates the four basic components, user interface, broker hierarchy, wrappers, and external archives. Sect. 5 concludes the paper.

2 Related Work

Well-established publishing systems, like Gopher and the World-Wide Web, provide a seamless information space in the Internet, at least as far as graphical browsing is concerned. Index and search subsystems appeared hand in hand with the rapid growth in the amount of information and in the number of users having specific needs. Obraczka *et al.* (1993) and Schwartz *et al.* (1992) give an overview of resource discovery approaches.

One of the earliest Internet indexing approaches were the Wide-Area Information Servers (WAIS) (Kahle and Medlar 1991), providing a Z39.50-based search and retrieval interface, and Archie (Emtage and Deutsch 1992). Archie periodically contacts a set of registered servers to gather a file index. Similar to that, Aliweb contains user-written summaries of server contents that are displayed on request.

More recently, with Glimpse (GLobal IMPlicit SEarch) (Manber and Wu 1994) an index/search subsystem has been installed that allows sophisticated searches over entire file systems. Among others, it allows misspelling and regular expression searches over non-uniform information including many types of documents. At the University of Karlsruhe, a prominent application has been realized on top of Glimpse, namely the sophisticated search facility for a large collection of computer science bibliographies.

Although multi-source index/search subsystems have already been built for Gopher, with Veronica, and for WWW, with Alta Vista, Lycos, and the World-Wide Web Worm (WWWW), retrieval engines or retrieval support systems for heterogeneous information are still open research fields (Barbara 1993). Early prototypes have however got an airing. The system Inquery, currently being developed at Amherst University by Callan *et al.* (1992, 1995), calculates the appropriateness of heterogeneous information sources with respect to a given query. It chooses the best fitting sources and conducts the search processes. At Stanford University gGLOSS (generalized Glossary-Of-Servers Server) addresses a similar idea. Following Gravano and Garcia-Molina (1995), gGLOSS keeps sophisticated statistics on available databases to determine an estimate of which databases are most appropriate for a given query. The search process is performed in a ranked list of databases. In contrast to Archie, which gathers an index without having a particular query in mind, Inquery and gGLOSS provide their indexes dynamically and are tailored to individual needs, viz a single query. The indexes then guide individual searches across the set of servers.

As soon as appropriate index/search prototypes were implemented, intelligent agents (CACM 1994, Wooldridge and Jennings 1995) or knowledge brokers (Barbara and Clifton 1992) started to exploit these subsystems. Harvest (Bowman *et al.* 1994a), for example, exploits as an index/search subsystem, both Glimpse and Nebula (Bowman *et al.* 1994b). Knowledge brokers are autonomous entities that may collaborate, negotiate, and coordinate, but which by no means can be coerced into activities such as searching information or answering a query whose scope does not conform with the broker's ability in query handling (Andreoli *et al.* 1995). Thus, knowledge brokers are generally used in combination with index/search subsystems.

In the Constraint-Based Knowledge Broker model (CBKB), constraints have been introduced to flexibly manage the search space of broker agents, as well as to flexibly adapt user requests and answers from information providers. Andreoli *et al.* (1996) present the

theoretical background of CBKB. Protocol issues within CBKB are addressed by Arcelli *et al.* (1995) and Borghoff *et al.* (to appear). Fikes *et al.* (1995) also use logic-based models to capture the domain of expertise of information brokers. Rather than using constraints, their modeling language is based on a predicate logic with contexts. The Tsimmis project (Chawathe *et al.* 1994) takes a different approach using a self-describing object model for the internal representation of information and queries.

It should be pointed out, however, that our approach differs from other frameworks for agent-based information gathering on the Internet not only in the technology, but also in the assumptions we make with respect to the development of a "cyber-economy." Indeed, we differ from those approaches which view the Internet as a kind of global market where agents – roaming over open electronic domains – will meet and gather information, possibly leading to business transactions. On the contrary, we see the Internet as evolving into a galaxy of *intranets*, linking together information providers and users around common interests. On the basis of these social and economic considerations, technological choices can be consequently specialized to optimally fit the requirements of specific intranets and user communities. Tools like Java, that provide capabilities for easy customizations of both client and server sides, appear particularly well-suited for this purpose.

This paper documents one such case of specialization, namely the adaptation of an agent infrastructure for constraint-based information gathering to the requirements of a network publication system for research and education.

3 Physicists Network Publishing System

In this section we describe the application environment, namely the Physicists Network Publishing System (PNPS), that serves as a testbed for the knowledge broker framework.

PNPS is embedded in a European distributed document database for physics (DDD-Physics). The DDD-Physics is a coordinated effort to organize, and to some degree standardize, the document-servers of physics departments and related research institutions and combine them with services of commercial providers such as publishers, database hosts, or libraries using common search interfaces. It will allow searches in a somewhat unified way over all diverse distributed document servers. This effort will be extended into other learned fields in Germany like mathematics, computer science and chemistry.

The PNPS will serve the remote user from an html browser to order a document from several document databases for printing-on-demand on local commercial copy centers.

3.1 Basic Architecture of the PNPS

There are three major functional subsystems:

1. *archive management.* There may be different archive systems, each managed locally by their providers (e.g. commercial publishers, scientific archives like E-print servers LANL, SISSA, or local department servers). The archives may be based on different management software and are integrated by the knowledge brokering system. This aspect of the subsystem will be discussed in detail in Sect. 4.
2. *production management and controlling.* The production management subsystem handles incoming jobs from network clients. A job describes a workflow, which in the simplest case is a printing task of a network document.

3. *clients and communication infrastructure.* The complete production process must be managed including accounting, authorization, billing, and logistics information.

3.2 Production Management and Controlling

Rank Xerox Germany System Operations (RXG-SO) has developed a printing-on-demand (POD) system based on Xerox printer technology in a local area network.

Special focus was placed on implementing the specific functions and requirements of the customers, and designing a highly generic rescalable system. In contrast to *network printing* application classes, where the emphasis is to have a highly universal network print server with strongly varying print jobs, the POD-system addresses an application class which may be called *archive printing*. The characteristics of archive printing are quite similar to the design model of local POD-systems:

- the system works in a nearly static and well-defined production environment. This means that the formats of the printable data are specified in advance, since they are to be stored in local archives.
- the system is interfaced to external control systems, for example, jobs are input from external systems, and information (accounting, document-keys, etc.) has to be returned to these external systems.
- the local archives are managed by system administrators, and there is a validation process for all new documents.
- documents are described by both page description languages (PDL) and in raster format.
- the print jobs are mixed with logistic information (delivery sheets, DP-data with forms, etc.) and archive documents.
- the print jobs may be huge (100.000 pages) requiring a focus on crash recovery.
- jobs enter the system anonymously, where they have to be identified, classified, and controlled.
- all printers serve as one printer pool, which can be controlled centrally.

3.3 Clients and Communication Infrastructure

Network clients use a general purpose interface (e.g. WWW) to input jobs into the production system. The communication infrastructure provides functions for authentication, authorization and accounting of services. Today, the first version of an accounting service exists. It is installed at the University of Oldenburg, to authenticate client orders by password.

Billing servers use the accounting information and other user related data (e.g. rating) to produce the invoices for the printing-on-demand service. The status of a job may be reviewed by the user client after it has passed the production stage.

3.4 Current Status and Futures

The dedicated POD-system was implemented for commercial applications in a LAN, following the operational model given above. In a second step it was extended to distributed

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.