Supporting Information Retrieval via Matchmaking

Daniel Kuokka Larry Harada
Lockheed Research Labs, O/96-20, B/254F
3251 Hanover Street, Palo Alto, CA 94304 USA
TEL: 415-354-5291 FAX: 415-354-5235
kuokka@aic.lockheed.com, harada@aic.lockheed.com

Abstract

The massive increase in information available via electronic networks is placing severe burdens on traditional methods of information sharing and retrieval. Matchmaking proposes an intelligent facilitation agent that accepts machine-readable requests and advertisements from information consumers and providers, and determines potential information sharing paths. Matchmaking permits large numbers of dynamic consumers and providers, operating on rapidly-changing data, to locate and share information effectively. This paper introduces matchmaking, as enabled by knowledge sharing standards like KQML, and gives a brief description of the the SHADE and COINS matchmakers. In addition, several applications are described to illustrate the utility of matchmaking for information retrieval.

Introduction

There has been a phenomenal explosion of information on electronic bitways such as corporate networks, the Internet, personal computer networks, and even television cable networks. This has led not only to a vast increase in information, but also to a vast increase in the number of information sources. These phenomena offer great promise for obtaining and sharing diverse information conveniently, but they also present a serious challenge. The sheer multitude, diversity, and dynamic nature of on-line information sources makes accessing any specific piece of information extremely difficult.

To address this problem, several exciting new technologies have been developed. The standards and protocols of the World Wide Web, as well as its associated browsers, have provided a hugely successful dissemination framework for previously disassociated information. Furthermore, integration frameworks from CAD vendors and telecommunications companies provide information connectivity where there was none before. However, both of these employ address-based messaging or browsing paradigms—the users must know where the information exists. Unfortunately, as users try to make the transition from adventurous explorers to goal-driven information seekers, it becomes very difficult to find desired information. The browsing paradigm employed by the Web has been overwhelmed by its own success.

In response to this problem, two common solutions have appeared: clearinghouses and exploration services. Clearinghouses, such as Commercenet and MCC's EINet Galaxy, are central servers at which individual information providers can register. Since there are relatively few clearinghouses, and they are organized in some fashion (and are usually searchable), consumers are able to effectively locate desired information. Exploration services, such as Lycos (Mauldin & Leavitt 1994) and the World Wide Web Worm (McBryan 1994), "crawl" the network compiling a master index. The index can then be used as the basis for keyword searches much like a manually-created clearinghouse.

These approaches provide very useful solutions to the overflow of information, but several problems remain. First, as the number and size of clearinghouses grow, they degenerate into a duplication of the network, itself (an interesting phenomenon is that many clearinghouses are becoming cross-indexed, allowing each to benefit from the knowledge-base of the others). Thus, inefficiencies and difficulties in locating a specific piece of information return. Also, exploration is a computationally inefficient approach (in terms of bandwidth, processor, and memory utilization), so it is usually applied sparingly, and therefore provides a limited index of the subject network.

More fundamentally, the above approaches make the assumption that information producers are (mostly) passive, so consumers alone drive the process. This necessarily imposes several handicaps:

- Information consumers must know of or arduously locate all relevant providers. However, today's networks are composed of millions of potential information sources, each of which may provide information dynamically. Thus, discovering all sources is very difficult.
- Information providers have no way to contribute their efforts. Even though producers often have a stake in delivering their information, and would therefore be willing to assist in the process, this potential goes unutilized.
- Once a connection is made, there is no means by which a provider can notify a consumer of new information or updates to past queries. Thus, in contexts where information is updated frequently and dynamically, approaches where the provider is passive simply can't work.

Matchmaking

A different approach to addressing this problem is called matchmaking. Matchmaking is based on a cooperative partnership between information providers and consumers, assisted by an intelligent facilitator utilizing a knowledge sharing infrastructure (Genesereth 1992; Patil *et al.* 1992). Information providers take an active role in finding specific consumers by *advertising* their information capabilities to a matchmaker. Conversely, consumers send *requests* for desired information to the matchmaker. The matchmaker attempts to identify any advertisements that are relevant to the requests and notifies the providers and consumers as appropriate.

Matchmaking is an automated process depending on machine-readable *messaging* and *content* languages. The main advantage of this approach is that the providers and consumers can continuously issue and retract advertisements and requests, so information does not tend to become stale. This is particularly critical in situations where information changes rapidly, as in product development and crisis management, and in situations where the shear magnitude of providers and consumers would cause the clearinghouse to be updated nearly continuously.

The representation must allow broad classes of information (i.e., many different documents) to be conveyed succinctly; otherwise, very many highly-specific messages, essentially duplicating the clients' database, would be required. Whereas this provides useful representational economy and efficiency, it dictates that advertisements and requests are only approximate versions of the actual information. Thus, false positive and false negative matches (depending on whether the advertisements and requests are over- or under-general) may occur.

As variations on this general theme, matchmaking can take many different specific forms. For example, the consumer might simply ask the matchmaker to *recommend* a provider that can likely satisfy the request. The actual queries then take place directly between the provider and consumer. The consumer might ask the matchmaker to forward the request to a capable provider with the stipulation that subsequent replies are to be sent directly to the consumer (called *recruiting*). Or, the consumer might ask the matchmaker to act as an intermediary, forwarding the request and forwarding the reply (called *brokering*).

An implicit form of the last case, called *content-based routing*, is also possible. In this approach, consumers *subscribe* to information as if the matchmaker were the source (thus, from the consumers' perspective, this is essentially an implicit version of the brokering case above). Providers, rather than advertising their capabilities, send updates (e.g., *tells*) to the matchmaker as changes to their knowledge-base occur. The matchmaker then routes the updates to the subscriber. This approach has the obvious problem of requiring providers to "blab" their results regardless of expressed interest, which may be infeasible given efficiency, bandwidth, and remuneration constraints. However, in simple cases, it has proven to be very useful and convenient, since it eases some of the representational and processing overhead of advertising, as imposed by brokering.

The SHADE and COINS Matchmakers

To evaluate and test the matchmaking approach, we have built two prototype matchmakers, the SHADE and COINS matchmakers, implemented as KQML-speaking agents. KQML, the Knowledge Query and Manipulation Language (Finin *et al.* 1993), is an emerging standard that defines a number of performatives (message types) for information exchange, such as *tell*, *broker*, and *subscribe*. The term agent is used to refer to a semi-autonomous tool or program, possibly under the guidance of a human, that interacts with other agents. Other researchers are also working on communicating agents that perform many matchmaking services, such as the ABSI facilitator (Singh 1993).

The SHADE matchmaker was designed and prototyped as part of the SHADE system (Kuokka *et al.* 1994; McGuire *et al.* 1993), an effort to define a knowledge-level communication infrastructure for engineering. The SHADE matchmaker handles a variety of KQML performatives. Advertisements are sent using the `advertise` performative. Requests are sent using the `recommend`, `recruit`, and `broker` performatives. The matchmaker also supports content-based routing, where `tells` from providers are forwarded according to `subscribes` sent by consumers.

As it's content language, the SHADE matchmaker supports two logic-based representations: a subset of KIF (Genesereth & Fikes 1992) and a structured logic representation called MAX (Kuokka 1990) augmented to support string patterns as terms. KIF is supported since it provides an expressive, standardized shared language with well-defined semantics. MAX is supported since it is convenient for representing highly structured data such as objects and frames. Furthermore, with its string matching augmentation, it provides a convenient means for advertising and requesting semi-structured text, such as outlines.

The SHADE matchmaker is implemented entirely as a declarative rule-based program within the MAX forward-chaining agent architecture. This allows features of the matchmaker (e.g., support for additional KQML performatives) to be added as additional rules. The actual matching of advertised and subscribed content fields is performed by a Prolog-like unification algorithm. If strings are present in the logic forms, a regular expression pattern matcher is used for term unification.

Motivated by the utility of the SHADE matchmaker on structured information and by the need for similar functionality over the huge amount of text available on-line, a second matchmaker has been created that operates on free-text as its content language. This matchmaker was initially conceived as the central part of a system called COINS (COmmon INterest Seeker), which allows users to easily advertise and request information about their interests. However, since COINS is architected as a set of agents, the COINS matchmaker is also useful as a general purpose facilitator.

As with the SHADE matchmaker, the COINS matchmaker is accessed via the standard KQML messages advertise and broker. However, the content language is unconstrained free-text (or optionally, preprocessed concept vectors to reduce the message size). To determine if a request matches an advertisement, the content of each message is converted into a concept vector (a weighed list of stemmed words in the document) using the SMART (Salton 1989) information retrieval system. The SMART matching algorithm is then used to determine the degree of match. Finally, an adjustable cutoff measure is used to make the match binary. Thus, other than supporting a different content language, the COINS matchmaker works much like the SHADE matchmaker.

The decision to implement two distinct matchmakers rather than a single, fully capable matchmaker was initially motivated by non-technical issues. However, it turns out that the resulting modularity is appropriate and beneficial in the agent-based world. Such an architecture allows many matchmakers, each created by researchers with specific technical expertise, to be specialized for specific classes of languages. If a single, multi-language matchmaker were needed, a simple dispatching agent could be developed that farms out requests to the appropriate subcontracting agent. Such a distributed approach may also address pragmatic issues of scalability, but little effort has been applied in this area to date.

Applications

The SHADE/COINS matchmakers are being used as a central component of several research projects. The SHADE project, itself, has developed a testbed for collaborative engineering to motivate and test infrastructure components such as the matchmaker. The testbed supports several engineers working together on the design of complex mechanical structures, including a systems engineer, a component designer, a rigid body dynamics analyst, and a controls engineer. These participants use a variety of tools that consume and produce complex engineering information, such as the SDRC I-DEAS solid modeler, Matlab, Mathematica, and the Lockheed Parameter Manager (Kuokka & Livezey 1994) and Project Coordination Assistant (PCA) (Kuokka 1994).

Rather than attempting to hardwire into these tools all the potential transfer paths for information (which would be impossible in general due to the dynamic nature of engineering teams), these tools use the SHADE matchmaker to advertise and subscribe their information capabilities and needs. For example, it is likely that many engineers would be using the Parameter Manager to state their constraints on the parameters of specific interest to them. When any one engineer decides to add a constraint, he has no way of knowing exactly which other engineers are impacted, and therefore whom should be notified. This is solved by each Parameter Manager sending advertisements and subscriptions to the matchmaker for the specific parameters of concern, allowing all agents to locate the new sources and sinks of information for this specific, unforeseeable engineering need.

The matchmaker is also vital to the operation of collaboration tools like the PCA. For example, a Systems Engineer might use the PCA to create a monitor for problem reports. The monitor is realized by sending to the matchmaker a KQML subscribe message that matches on specific content keywords. As engineers make reports, they are also forwarded to the SHADE matchmaker as KQML tells. This allows the matchmaker to identify those reports among the extensive message traffic that declare problems, and send them to the Systems Engineer. PCA reports are also sent to the COINS matchmaker to locate other relevant information. For example, another project (potentially in a completely separate organization) may have already addressed the problem being reported. As long as that organization is also using the matchmaker, their reports can be matched against those of the local project. In this case, if a similar problem report had been logged, the matchmaker would send a pointer to the local PCA. Thus, highly relevant information, which might otherwise never have been discovered, would be brought to the attention of the Systems Engineer.

The matchmaker has been used by several other engineering-related projects as well. The Cosmos project (Mark & Dukes-Schlossberg 1994), which is creating a knowledge-based commitment reasoner to determine impacts of engineering changes, uses the matchmaker to provide indirection between a set of dynamic clients and the server. The ARPA Simulation Based Design project uses the matchmaker to provide change subscription and notification services over its large, object-oriented product model. In this application, if an object for which a subscription has been issued changes, the user will receive automatic notification. Other applications of the matchmaker, such as its use to locate relevant pages in a large distributed engineering notebook, are in earlier stages of development.

The functionality of matchmaking goes beyond engineering teams. For example, the matchmaker is an integral part of a prototype information retrieval system being developed to support Lockheed's SII (Space Imaging, Inc.) project, an effort to sell high-resolution remote sensing imagery on the commercial market. A key element of this task is to locate data available from multiple dynamic satellite image providers in response to specific queries. The SII prototype uses the SHADE matchmaker to perform this task.

The system works as follows. As new classes of images become available, the data sources issue advertisements in terms of the image attributes (e.g., geographic area, resolution, spectral bands, and cloud cover). When a user needs a specific kind of image, she uses a front-end agent to formulate and issue a query to the matchmaker that describes the desired attributes. The matchmaker compares the advertisements to the query, and sends any potential matches back to the front-end agent. The servers located by this first-pass match are then issued further, more specific, queries. In addition, when a source database is not appropriate, the matchmaker returns a failure reason.

The matchmaker is important to the SII application not only because there are multiple sources of data, but also because the data is constantly being updated as satellites circle the earth. The matchmaker allows each data source to advertise and retract its image capabilities dynamically, permitting the matchmaker to suggest sources even if the specific image hasn't yet been collected. Only an automated system like the matchmaker can offer the up-to-theminute location of data required by SII.

Conclusions

The growth of information available via electronic networks presents both an unprecedented opportunity and a difficult challenge. Rather than relying on traditional techniques that are static and consumer-driven, matchmaking allows both of the stake holders (i.e., information providers and consumers) to contribute to information gathering activities. Thus, information providers can seek specific consumers much like consumers currently find specific providers. In addition, since matchmaking is an automated approach, it better addresses the dynamic nature of electronic information, which is characterized by huge numbers of potential information providers and consumers and rapidly changing information. The need for such an approach is underscored by the rapid adoption of the SHADE and COINS prototype matchmakers by several projects.

However, matchmaking is still an experimental approach, and many questions remain. Additional support is required for formal languages such as object and terminological representations, and a capability to define relevant knowledge bases and ontologies is needed to permit matchmaking based on deeper content, subsumption reasoning, and inference (however, the matchmaker cannot become the reasoning engine to the world). Also, further expansion into free-form human languages and graphics is needed, going beyond the current concept vector abstraction of text. Looking beyond the content language, the experiments with matchmaking to date have already begun to stretch the KQML messaging substrate. Further augmentations are required to support additional modalities and to clarify the semantics of the existing message types. And finally, as applications grow in size and complexity, techniques to distribute the matchmaker load will be required. Yet, in spite of these open issues, matchmaking is a promising approach to supporting information access in heterogeneous and dynamic environments.

Acknowledgments

The initial concept of matchmaking and the KQML formalization stemmed from efforts by the entire SHADE team, including Jim McGuire, Jay Weber, Greg Olsen, Rich Pelavin, Tom Gruber, and Marty Tenenbaum, as well as the efforts of the KQML committee. A longer version of this paper, focusing more on workgroups, is available from the author.

References

Finin, T.; Weber, J.; Wiederhold, G.; Genesereth, M.; Fritzson, R.; McKay, D.; McGuire, J.; Pelavin, R.; Shapiro, S.; and Beck, C. 1993. Draft specification of the KQML agent-communication language. Technical report, The ARPA Knowledge Sharing Initiative External Interfaces Working Group.

Genesereth, M., and Fikes, R. 1992. Knowledge Interchange Format, version 3.0 reference manual. Technical Report Logic-92-1, Computer Science Department, Stanford University.

Genesereth, M. 1992. An agent-based framework for software interoperability. In *Proceedings DARPA Software Technology Conference.*

Kuokka, D., and Livezey, B. 1994. A collaborative parametric design agent. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).*

Kuokka, D.; Harada, L.; Weber, J.; Tenenbaum, J.; Gruber, T.; and Olsen, G. 1994. SHADE: Knowledge-based technology for the re-engineering problem; final report. Technical Report http://hitchhiker.space.lockheed.com/aic/shade, Lockheed Artificial Intelligence Center.

Kuokka, D. 1990. *The Deliberative Integration of Planning, Execution, and Learning.* Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University.

Kuokka, D. 1994. An evolution of collaborative design tools. In *AAAI-94 Workshop on Models of Conflict Management in Cooperative Problem Solving.* AAAI Tech. Report WS-94-04.

Mark, W., and Dukes-Schlossberg, J. 1994. Cosmos: A system for supporting engineering negotiation. *Concurrent Engineering: Research and Applications* 2(3).

Mauldin, M., and Leavitt, J. 1994. Web-agent related research at the CMT. In *Proceedings of the ACM Special Interest Group on Notworked Information Discovery and Retrieval (SIGNIDR-94).*

McBryan, O. 1994. WWWW—the World Wide Web Worm. http://www.cs.colorado.edu/home/mcbryan/WWWW.html.

McGuire, J.; Kuokka, D.; Weber, J.; Tenenbaum, J.; Gruber, T.; and Olsen, G. 1993. SHADE: Technology for knowledge-based collaborative engineering. *Concurrent Engineering: Research and Applications* 1(3).

Patil, R.; Fikes, R.; Patel-Schneider, P.; McKay, D.; Finin, T.; Gruber, T.; and Neches, R. 1992. The DARPA Knowledge Sharing Effort: Progress report. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning.* Morgan Kaufmann.

Salton, G. 1989. *Automatic Text Processing—The Analysis, Transformation and Retrieval of Information by Computer.* Addison-Wesley, Reading, MA.

Singh, N. 1993. A CommonLisp API and facilitator for ABSI (revision 2.0.3). Technical Report Logic-93-4, Stanford University Computer Science Department Logic Group.