CHAPTER 6 *Related Work*

6.1 Introduction

We have presented a general architecture and a sample application, and some evaluation of them, designed to promote a particular sociopolitical agenda and to demonstrate that starting with such an agenda can lead to technological advances. Let us now turn our attention to related work in some relevant fields.

This research touches on a large number of possible topics. We shall restrict ourselves here to examining:

- Other types of matchmaking systems — Section 6.2
- Other decentralized systems — Section 6.3
- Other systems and software that have been designed for political purposes — Section 6.4

6.2 Matchmakers

In general, systems that perform any sort of matchmaking task are *centralized* systems. Such an organization has several useful advantages, especially to the implementors of such systems, if they do not also have to deal with personal information:

Why centralization is a popular approach

- They are easier to administer—all, or almost all, of the relevant software can run on hosts directly under the administrative control of the implementors
- If they are being used for a business, it is often obvious how to structure the system such that users may be charged fees, or have advertising delivered to them as they use the system
- If the business model of the matchmaker *also* requires that personal information be reused for other purposes—such as marketing—then centralizing all data on the company's own servers makes this easy.

Collaborative filtering

Webhound/Webdoggie [103] and HOMR/Ringo/Firefly [112], for example, are typical examples of centralized matchmakers. A central server maintains information about user interests, and users connect to the server (in both cases, with web browsers) to discover whether they have a match. Both systems require the user to be proactive in establishing and maintaining an interest profile, although Webhound/Webdoggie also obtained leverage by using a data source the user already kept updated, namely his or her hotlist.

Brokering services

Kuokka and Harada [99][100] describe a system that matches advertisements and requests from users and hence serves as a brokering service. Also a centralized server, their system assumes a highly-structured representation of user interests.

Sixdegrees

Sixdegrees [110] is an interesting idea in matchmaking, generally for professional reasons; this site keeps track of *who you already know* and uses this information to find minimal spanning trees to others who you would like to know. It does this by asking for email addresses corresponding to others that you know, and also for their relationships to you (as well as other information, such as their profession), and then contacts those people to see if they agree. If they do not repudiate the relationship, the system records the correspondence. Users are always identified; unlike most other systems, there are no pseudonyms. Users can then ask queries such as, "Who do I know who knows a lawyer?"

The system is somewhat cumbersome because of the need to involve everyone explicitly (anyone you name must take the effort to become a member themselves), but its narrow targeting of social relationships makes it likely to find interesting contacts. It is, of course, another centralized system, although it takes certain efforts both to reassure its users that their information will remain private—although, of course, they make no assurances about either crackers or subpoenas— and that the system cannot easily be *gamed* to expose large numbers of relationships—for example, you can only find out about the relationships of other people to people you already know, out to a very limited diameter, and can only spam those you already know, which is presumably not very productive.

PlanetAll

PlanetAll [150] takes a somewhat different approach. It concentrates on *finding people you once knew*, rather than on finding new people you might like to know. Like Sixdegrees, it is a centralized, web-based service, and everyone using the service is identified by their real name. Unlike Sixdegrees, the primary organizing principle behind PlanetAll is *affinity groups.* Such groups are prespecified, named entities corresponding to organizations in the real world—not online—of which the user was at one time a member. They are typically schools, clubs, or religious organizations, and PlanetAll allows one to search for them by keyword. When registering, the user specifies affinity groups, and is then notified when others join the group. He or she can send messages into the group or to particular individuals.

Spamming is prohibited by the rules of service, and, since individuals are always strongly identified, tracking them down and barring them is easy. On the other hand, it is not clear what would happen if someone who was never part of some affinity group in real life were to join one anyway—such a party crasher would probably simply be tolerated, as least if he or she was not obnoxious, because everyone else in the group might assume that *someone* knew them.

One can also tell PlanetAll about particular individuals in the system and ask it to send mail when that individual's information (such as work address) changes. It is presumed that individuals already know each other when they receive notification of one joining the group—thus, PlanetAll concentrates on finding people after one has lost track of them, rather than on describing unknowns to each other. PlanetAll also has a number of other interesting features. For example, it allows users to enter their travel itineraries, and will notify them when their paths cross in foreign cities.

As with Sixdegrees, PlanetAll users must trust that the central site will protect their personal information. Since such information could be valuable to a number of commercial interests, and also to those contemplating identity theft, this could be a major exposure.

Romantic matchmakers

Although romantic matchmaking is *not* an explicit goal of Yenta, there are a large number of matchmaking systems specialized for this application, and they are worth studying. Such systems appear to be invariably centralized. For example, Match.Com [41] is a straightforward romantic-matchmaking service. Users fill out a form detailing their own characteristics and those of people they would like to meet (sex, age, geographic location, etc.), which are used in a simple match/filter algorithm; they also post personal ads to supply more detail once a user's filter has selected some ads. Similarly, the Jewish Matchmaker [43] (unfortunately also called Yenta, for obvious reasons) is one of several more-specialized systems that function similarly: surveys for filtering, personals for secondary selection, and a centralized server, all backed up by a web-based interface.

A rare decentralized example

Kautz, Milewski, and Selman [91] are one group, of very few, to have taken a more distributed approach to matchmaking. They report work on a prototype system for expertise location in a large company. Their prototype assumes that users can identify who else might be a suitable contact, and use agents to automate the referral-chaining process. They include simulated results showing how the length and accuracy of the resulting referral chains are affected by the number of simulated users and the accuracy and helpfulness of their recommendations. Yenta differs from this approach in using ubiquitous user data to infer interests, rather than explicitly asking about expertise. In addition, Yenta assumes that the individuals involved probably don't already know each other, and may have interests that they wish to keep private from at least some subset of other users.

6.3 Decentralized systems

There are a variety of other decentralized systems that bear consideration here. For the most part, these systems may be divided by their underlying metaphors: *biological*, *market-based*, or *other*. We shall discuss all three below.

Both biological and market-based systems are often used in the allocation of scarce resources, although with a difference in emphasis. For example, biological systems often model individual actors or agents through their births, lives, and deaths. It is commonly assumed that the characteristics of agents change relatively slowly over their lifetimes, but that an entire population may change through evolution. Individual agents generally have very limited models of the world and sometimes vanishingly small reasoning abilities. Market-based systems, on the other hand, tend to assume agents which exist for indefinite spans of time, but can change their behavior relatively quickly due to learning within an agent. In addition, information flows—as opposed to flows of matter—are often considered to dominate the interaction, and explicit negotiation between agents with high levels of reasoning are common.

Biological metaphors

The *artificial life* approach is explicitly informed by a biological metaphor [94]. This discipline tends to model systems as small collections of local state that have generally been mapped into a simulation of some physical space. Within this space, these bundles of state may interact solely through local interactions—there is no action at a distance. Systems modeled often tend also to simulate real biological systems, albeit simplified versions—ant and termite colonies [142], predator/prey systems and various simulations of Darwinian or Lamarckian evolution [19][102], learning [57][62], immune systems [95], and many more. Some simulate decidedly nonbiological systems using biological metaphors—for example, many problems in optimization are often effective solved using genetic algorithms [96]; for example, producing optimal sorting networks [79].

The choice of self-contained bundles of state, and strictly local communication, stems naturally from systems which either simulate or are inspired by the natural world, where nonlocal effects tend to be rare. Most such systems run on uniprocessors, but there are exceptions. For example, many learning [62] or simulated-evolution [165]

systems have been implemented on SIMD or MIMD architectures such as the CM-2 or CM-5 Connection Machines from Thinking Machines. Others have been distributed to collections of uniprocessors connected via the Internet. One example is *NetTierra* [139], a network-based implementation of the original *Tierra* [138], a system originally written to explore the evolution of RNA-based life via an easy-to-mutate machine language.

Market-based metaphors

Market-based approaches tends to use negotiation, barter, and intermediate representations of value—such as money—to enable a collection of actors to decide on individual strategies [25][111]. One example of such a system is *Harvest* [74], which uses a decentralized collection of *gatherer*, *broker*, *collector*, and *cache* elements to greatly improve the performance of, e.g., web servers. Element use market-based ideas to decide how to allocate various resources such as storage or bandwidth.

Consider also a system in which we have a *heap*, such as that found in a Lisp system, where objects point at each other. Reclaiming unused space in a heap is called *garbage collection*, and doing so if the heap spans multiple machines can be quite slow due to communications overhead. Using a market-based approach, in which storage essentially *pays rent* and storage which runs out of money is deallocated [40] can make this problem much more tractable by keeping almost all the computation required local to individual machines.

Other approaches

Not all decentralized systems necessarily require either competition or cooperation between agents—some simply use decentralization to achieve pure parallelism, turning a network of uniprocessor CPU's into an emulation of a MIMD multiprocessor. One common example of this these days is *cryptographic key cracking* [32], in which thousands of CPU's participate in searching the keyspace of a particular encrypted communication. This application is typically *political* in nature—in general, participants take part in order to help demonstrate that ciphers such as 56-bit DES are woefully insecure [15][24][34][42][76][184].

6.4 Political software and systems

Let us now examine various software systems that have been designed with a particular eye towards their political environment. We will concentrate here only on systems which attempt to advance what we believe to be the *socially responsible* position in our political argument—and not, for example, systems such as the centralized Intelligent Transportation Systems described in Section 1.4.

Pretty Good Privacy

By far the most famous example of such software is *Pretty Good Privacy*, or *PGP* [187]. PGP is one of the most widely-used strong-cryptography packages in the world. Recent versions have even been deliberately exported from the United States, even though doing so electronically is illegal. Instead, the First Amendment to the US Constitution was exploited as a loophole—it has already been determined that printed books are not subject to regulation under US export-control law. Thus, source code was printed into a ten-volume book, which *is* legal to export, in a format that was explicitly designed to be easy to scan and convert back into electronic form overseas. (Since then, other important cryptographic efforts have been exported in the same way—for example, all of the VHDL and loader code describing how to build a hardware DES-cracking machine was printed in machine-scannable form expressly to allow this [42].)

PGP's development was motivated by explicitly political aims—its author, Philip Zimmerman, wrote it to make strong cryptography easily available to the masses, or at least to those masses who owned personal computers. And since then, it has become a lightning rod for discussion concerning US cryptographic-export policy.

PGP itself does not depend on any sort of network infrastructure—it encrypts and decrypts files only. However, it is most useful when combined with a network, rather

than when being used to mail encrypted floppies back and forth. Various popular mail-handling programs, such as Eudora for Macs and PC's, and Mailcrypt for GNU Emacs, have incorporated it into their design.

Anonymous remailers

Other political software has made the network a more explicit part of their design. Consider anonymous remailers [10][23][66], which are designed to hide the origin and destination of messages being sent from one computer to another. They work by encrypting messages in transit, and routing them through a large number of computers in various political domains. The assumption is that no single entity could successfully compromise every computer and every network link in the chain, and that this lack of total surveillance will allow truly-anonymous information exchange.

The contents of such messages are varied. Many concern topics which are potentially embarrassing or dangerous to those discussing them, such as unusual lifestyles, or discussion of medical problems such as HIV which might cause the discussant to lose his or her job or social standing. Others are explicitly political in nature, sent by people living in regions where political dissent can lead to imprisonment or execution [11].

anon.penet.fi

One particularly famous remailer was the *anon.penet.fi* remailer [77], run by Johan Helsingius. This service offered single-hop anonymity—messages sent to this remailer had identifying information stripped out, but were then delivered as usual to their destination. This made it particularly easy to use without the special software often required of multihop *Mixmaster* [10][23][66] remailers. It also offered *nyms*—one could have a stable, pseudonymous identity through the use of this service, rather than being completely anonymous. Anyone could reply to a message posted through anon.penet.fi, back to the original author, even though both parties would not know each other's actual identities.

This mechanism also led to a certain amount of insecurity. For example, in one well-publicized case in 1995, the Church of Scientology was able to get the local government in Finland to subpoena the site's operator for the mapping between one particular nym and the real email address of the person behind it. In 1996, the Church tried to determine if a particular individual had ever used the service. The site was eventually shut down by its operator, who cited the increasing load on his time that running it required, and the availability of at least partial substitutes elsewhere on the net.

The Anonymizer

Consider now the *Anonymizer*, which attempts to make it possible to fetch web pages without informing the web server of the identity of the machine doing the fetching—presumably for use in reading pages with controversial content, or to deny marketers the ability to target the reader for profiling. It is a single, centralized server, and simply proxies requests through itself, rewriting HTML links such that following a link on a fetched page will go back through the Anonymizer. While it can effectively hide users from sites, it is useless against traffic analysis attacks—it operates at a single, well-known address and from a single point of presence. This makes its communications easy to tap, either at the site or by looking for requests from a given user to the Anonymizer itself. Even if SSL is in use, thus hiding the actual URL's being requested and the contents of the pages returned, traffic analysis at the user's site can instantly reveal that the Anonymizer is in use at all, and even this is often sufficient to target the user for various unfortunate consequences. Further, sites which offer content may deliberately deny content to the Anonymizer, to force users to come from well-identified IP addresses. Finally, users of the Anonymizer must trust that the site really *is* honoring its stated policies of not keeping logs of the traffic through itself.

Crowds

A more-sophisticated system, developed after the Anonymizer, is the *Crowds* system [141]. This system is also an attempt to strip identifying information from web surfers, and uses decentralization to foil traffic analysis. Participating users join a

crowd—a collection of other machines, all of which participate in the system, and which randomly reforward HTTP requests and responses among themselves before sending them to their final destinations. This means that any particular web page fetched by a user could come from any of the participating machines at random, hence denying the web server the ability to know precisely who is fetching which pages.

This system is explicitly aware of the problems of traffic analysis, both at the web server itself and in the intervening links between that server and the user, and takes steps to foil it. It also reduces the problems of trusting the privacy policy of a single site.

Web filters

Web-filtering programs grew directly out of political concerns—they are software packages which are deliberately designed to block content from particular users, generally minors and anyone else who might be coerced into using them, such as library patrons in some cases. Some of them, such as RSACi [140], rely on self-ratings by sites. Others, such as PICS [177], rely on third-party ratings. These third-party ratings may be either public, and possibly distributed, or provided by the manufacturer of the filtering software, and often private.

Since *someone* must choose which sites are acceptable and which are not, there is an implicit political agenda to using such software. Even systems which claim to allow the user to select any other third party's recommendations may be abused given enough control of the network infrastructure. For example, China carefully controls traffic across its borders, and could insist that all web surfers use only government-approved PICS sites for their filter lists. In addition, those systems in which the vendor of the filtering software choose are often extremely heavy-handed about what sorts of sites are deemed unacceptable. In response to this, Bennett Haselton [75] has spent considerable time and effort exposing the antics of filter manufacturers who claim to be blocking "sexual content" but are also blocking a wide variety of nonsexual web sites that happen to have politics that the filter vendors find unacceptable. The list of sites blocked by these packages are secret, ostensibly for reasons of competitive commercial advantage, but this means that there is virtually no oversight for what often turns into an appalling censorial exercise.

FSF and Open Source

Finally, let us consider an *intellectual property methodology,* as opposed to particular systems or programs. The methodology of interest is the union of the *Free Software Foundation* and the more-recent *Open Source* movement. Both of these approaches view *freely-redistributable software* as a social good. While they differ on the details of what this means and how to achieve it, they are in substantial agreement that the *freedom to examine and modify source code* is the cornerstone of building high-quality software. Many famous examples of their effort exist, such as the GNU collection of hundreds of utility programs—Emacs, autoconf, automake, gtar, gmake, and all the rest—and other projects which use their licensing terms but were not written by the FSF—such as Linux, SCM, and so forth.

Both the FSF and the Open Source group have an explicit political agenda, which they enforce through the technology of copyright and contract law. Thus, their technology is that of intellectual property per se, rather than that of software itself. Their efforts have had an enormous effect on the way that software is currently developed, especially—but not exclusively—that which runs under various varieties of UNIX, and is likely to leave a considerable legacy.

6.5 Summary

In this chapter, we have touched briefly upon matchmakers, decentralized systems, and politics. All three of these fields are assuming increasing importance as the Internet continues to expand and its user base continues to grow. The research that led to

Yenta and its underlying architecture did not arise from the vacuum. Instead, it is explicitly informed from—and, in some cases, in reaction to—some of the existing systems and methods of practice currently popular in the field.