User Profiling with Privacy: A Framework for Adaptive Information Agents

Ian Dickinson[1], Dave Reynolds[1], Dave Banks[1], Steve Cayzer[1], and Poorvi Vora[2]

[1] Hewlett-Packard Laboratories
Filton Road, Stoke Gifford
Bristol BS34 8QZ, UK
{ian.dickinson, dave.reynolds, dbanks, steve.cayzer}@hp.com
[2] Hewlett-Packard Laboratories
1000 NE Circle Blvd Corvallis, OR 97330, USA
poorvi_vora@hp.com

Abstract. This paper presents a framework for personal agents that respect the privacy of the individual. We present some motivations and outline a framework for the use of personal agents and user profiling for information systems designed around web services. A key element of our approach in general is to consider the impact of user-profiling and autonomous agents on the user. One particular aspect, which we explore in this paper, is the need to respect user's privacy. One often-cited benefit of using personal agents is for personalising interaction. However, personalisation and privacy have contradictory goals in disclosing personal information. We explore some elements of our framework that allow the user to control the trade-offs around disclosure of personal information. We conclude with some motivating examples of the use of our framework in information-based tasks.

1 Introduction

A widely recognised problem for users of information systems today is the frustration caused by being unable to find and manage the right information. This is often referred to as *information overload*. The increasing range and sophistication of interesting sources of content on the World Wide Web provides more choices, and, in principle, a greater chance of ultimate success. Such plenitude can be elusive, however, when the costs to the user, in terms of time, effort and attention, are too great. One way that information overload can be addressed, is to use more information about the user's needs and objectives in the process of finding and using information resources – known generically as *personalisation* [1]. There are many ways to achieve personalisation [2], but here we focus on the use of information about the user to *adapt* the user's interaction to their specific needs. The study of *adaptive information agents* [3] addresses the design, use and evaluation of agent-based tools to assist a user to perform information-centric tasks effectively. Here the agent metaphor captures the intuitive concept of a knowledgeable third party, for example, a trained librarian. While the term *agent* is widely used to mean a number of different things [4], our present objective is to build tools to assist the end-user. Hence, we are not referring to mobile or infrastructure agents, but to what are often termed *personal agents* [5].

Applications built around the personal agent metaphor have been widely explored in the literature, and some illustrative examples are outlined below. A common feature of these applications is the use of knowledge about the user in order to provide

M. Klusch et al. (Eds.): Intelligent Information Agents, LNAI 2586, pp. 123-151, 2003.

an adaptive, or otherwise personalised, service. In this context, many authors raise the question of privacy for the end-user as a concern with such user models. A focus for our work is an attempt to build an open, extensible user-profiling platform, but which emphasises privacy and security of the user's personal data. The research we are reporting is still ongoing: this paper presents motivations, results to date, and an overall positioning of our research in the context of related research fields.

The paper is structured as follows: section 2 defines the general need for comprehensive representation of user preferences as a foundation for novel information management tools, including information agents. Section 2 also briefly reviews some prior work in user modelling, and describes extensions to prior approaches that motivate our research. In section 3, we outline the core components of our approach, which we call the *ePerson* approach[1]. We recognise that a fundamental requirement in any system that manages user information is privacy, and we regard our approach to privacy as one of the distinguishing features of our approach. Section 4 reviews the issues for privacy, and some of our research on solutions to these issues. Section 5 describes some application scenarios for the ePerson. We conclude with a review of similar work, and summary of future research issues.

2 Motivation: The Need for General User Models

The use of information about a user to modify that user's interaction with a computing system or information resource is well established. Typically, the stored information about the user is referred to as a *user profile* or *user model*. User-modelling is an established research field in artificial intelligence [6]. Many adaptive or personalised systems contain an embedded user model, and use this local model as the foundation for, e.g., personalised recommendations [7]. It has been observed that this approach requires the application developer to build and maintain their own user model, which has a number of drawbacks. Principle concerns about such an approach are:

- It requires the application developer to spend additional time designing and developing a user-modelling component. This adds to the effort required to build the application. In addition, the developer may not have the time, or expertise, to address all of the relevant concerns relating to the user model, including, for example, privacy protection.
- Information is typically not shared between user models. This has two impacts: first, the subset of user information in any single model gives a less complete, and arguably less helpful, picture of the user's needs and preferences. Secondly, there is a significant cost to the user in having to manage multiple instances of their personal preference data, especially as user model data is known to change frequently [8].

A *generic user model* attempts to alleviate these concerns by centralising user model construction and maintenance in one program (e.g. a user modelling server), which then makes the user data available to client applications. In a recent article,

[1] We note that the terminology "ePerson" was first coined by Roscheisen [85], though in the more specific context of rights contract negotiations.

Alfred Kobsa reviewed a range of generic user modelling tools, ranging from research prototypes to commercial products [9]. A companion paper by Fink and Kobsa compared the features of four classes of currently available commercial personalisation tools [2]. These general-purpose tools attempt to abstract away the problem of constructing and maintaining a user model, providing a *user modelling service* for other application components to utilise. Note that this is an abstract description, and is neutral about the architecture used to integrate a user model into the application[2]. A summary of Kobsa's list of abstract services includes:

- the representation of assumptions about one or more types of user characteristics in models of individual users, formation of those assumptions, and deriving new assumptions based on observations;
- the representation of relevant common characteristics of specific user subgroups of the application domain (often called *stereotypes*), classifying users according to these subgroups, and drawing inferences according to sub-group membership;
- the recording of user's behaviour, particularly their past interactions with the system;
- the generalisation of the interaction histories of many users into stereotypes;
- consistency maintenance in the user model (from Kobsa 1995, quoted in [9] p52).

There is every reason to suppose, going forward, that all of the above capabilities will continue to be useful or essential in the construction of personalised or adaptive user interfaces. However, as the world of networked information services grows in complexity, we can anticipate a number of additional requirements that will need to be addressed by a new generation of user profiling services, as follows:

- *small devices and intermittently connected networks* – adaptive interfaces will be needed on information appliances, characterised by reduced computing power and intermittent connection to the network (see also [10;11;12] and [9] p58,)
- *awareness of the user's current context* – including, but not limited to, the user's current interaction device (PDA, phone, kiosk, desktop computer, etc), the user's location and the availability of location-based services and the user's current role (see also [12;13;14;15])
- *open and extensible* – as more and more user information is available via web-based services, it is clear that the use model can never be specified as a closed system. Instead, the model should be able dynamically to incorporate information from disparate sources, and not duplicate information that is maintained elsewhere. Ideally, the user model should provide uniform, consistent and flexible access to all sources of user information, and support a multiplicity of means of acquiring information about the user;
- *privacy protection* – commensurate with the increasing range and scope of information about individuals stored on computers, concerns about protecting privacy have increased dramatically. We discuss the many dimensions of privacy in more detail in section 0.

---

[2] For example, the user model may be a separate server-based process, or may be a self-contained component that simply plugs into a stand-alone application. We will further discuss architectural issues in section 2.

In essence, it is these additional requirements that provide the motivation for our approach to the user profiling problem, as we discuss further below.

While there are many dimensions along which we can compare generic user modelling systems, two interesting clusters emerge from previous developments in academia and industry. On the one hand, academic approaches have typically favoured deep, sophisticated[3] models, but focussed less on practical issues, whereas commercial systems have generally employed shallower models, but emphasise scalability, security and performance. Kobsa identifies the key features of the academic approach as follows: generality (including domain independence), expressiveness, and strong inference capability. It is also common that academic generic user models employ the "mentalist" paradigm, recording the model's understanding of the user's beliefs, goals, knowledge, etc, often using a modal logic [16]. The alternative approach, typified by the more commercially oriented personalisation servers (and also by machine-learning approaches to user modelling [8]) does not attempt to define the user model in declarative, mentalist terms. For example, the GroupLens product from Net Perceptions [17] provides recommendations by associating patterns of user activity with outcomes, such as reading a news article or buying a book. GroupLens cannot produce a declarative explanation of the user's needs (e.g. an SQL programmer learning Java[4] might be predicted to have a need for information on JDBC), but, given enough data, can recognise a pattern (e.g. "people who bought books on SQL and on learning Java also bought books on JDBC). The reasoning in such systems is statistical, compared to the knowledge-based inferencing schemes in the mentalist academic user modelling systems. However, this has a strong benefit, in that such systems can be, and typically are, much more robust and predictable in their behaviour, and scalable to moderate size data sets. Consequently, such commercial systems are often better at addressing practical deployment issues, including scalability, privacy, and, to a lesser extent, integrating with existing or legacy sources of user data in the organisation (see review in [2]).

A user model may be extended by the addition of new user profile categories and individual data. This presents a difficulty in terms of the design goals we outlined above. For instance, we would like to empower users to express rich policies about when and how their private data may be used. To do this, the policy must be able to address the contents of the user model by function or category. For example, imagine the user wishes to express the following privacy policy:

```
Any service acting on behalf of my employer can access
my employment details, and preferences for job-related
information systems.
Any member of my photography circle can see details of
my published photographs on photo.com.
```

To carry out this policy, it must be possible, in the user model, to identify employment details and distinguish them from, say, contact details for friends. A common approach to this problem is exemplified by P3P [18]. P3P pre-defines a *standard data model*, containing *well-known names* for elements of the model. These

[3] Some would argue "complicated".

[4] Java is a registered trademark of Sun Microsystems.

well-known names are agreed in advance by users of the P3P standard to have a particular form and meaning. An example such name is "`user.home-info.online.email`". This approach, however, does not support extensibility well. Even if a mechanism is provided for propagating new public names, their meaning will be hard to convey. What is needed is a means to be able to name or identify different parts of the user model, without identifying symbols necessarily being defined in a pre-existing standard. Beyond the difficulties of agreeing the meaning of shared symbols, we anticipate that significant research, and innovative design, will be necessary to make privacy policy specification amenable and attractive to the individual. The agent should give a significant amount of support to the average home or office user, ensuring that the goal of placing privacy under the user's control is becomes real.

Another aspect of this problem concerns *open access* to the contents of the user model. It must be possible for services using the data from the user model to provide suitable adaptations, without necessarily knowing *a priori* what the elements of the model are. Even if tools like GroupMinds are able to recognise very robust patterns of behaviour, other services cannot integrate those patterns into their personalisation strategy unless there can be a meta-description of the affinities that have been detected.

Our goals for extending the user-profiling paradigm to include these features suggest a number of additional design goals that must be adopted. In summary, these are:

- Place the user in control of their own data;
- Make user profile data available more widely, including shareable between applications;
- Ensure that access to all profile information is governed by a privacy policy that the user controls;
- Do not assume a particular network topology; especially do not require all data to be centralised in one place;
- Allow the data model for the profile to be extended in a meaningful and open way.

3 An Outline Framework for Open Extensible User Profiles

Given the motivation, and in particular the design goals, outlined above, we are developing an open platform to provide a uniform means for applications to access and share user profile and user model information. This platform provides one basis for representing of the user in the shared information space. We use the term *ePerson* for this representation. Fig. 1, below, illustrates the key features of the platform. It provides a common, policy controlled, access mechanism for stored and indirectly obtained profile information that a variety of client agents can access.

For the user, such a common shared platform offers the advantages of convenience and control. Conceptually, the user profile is in one place and the user has one interaction point for checking and updating information, and adjusting the access control policies. In implementation, the profile might be distributed across several devices – for example, parts of it might be cached on the user's PDA, phone or PC. However, it is important to maintain the illusion of a single control point both to simplify the user's conceptual understanding of the profile (and how to control it), and to simplify the design of profile-using web services.

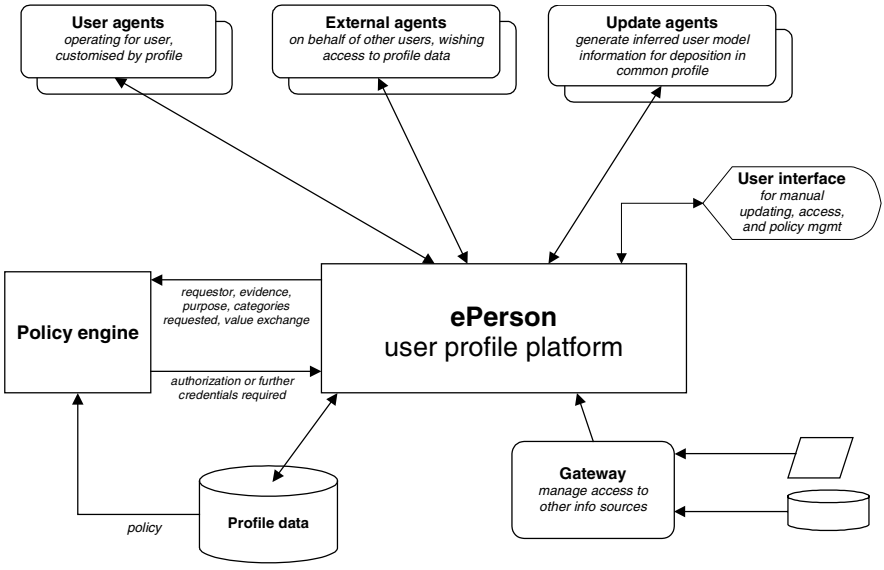

Fig. 1. Outline ePerson system framework

For applications, including information agents, wishing to access the user profile in order to carry out their tasks, the shared profile service provides uniform access to a richer set of user information than the application itself could have generated. It also relocates the problem of checking access requests against a user's privacy policy from the individual agents to a shared policy engine. By pooling this resource, the information agents can access better information more flexibly, while the user experiences greater control of the profile data.

Design Principles

This platform outlined above raises several design challenges, including:

- *Open platform*. The platform must support a dynamic, open-ended set of client agents and services.
- *Semantic extensibility*. The profile information itself must also be openly extensible. New agents may be able to generate and exploit new categories of profile information that the platform must accommodate. Even for a fixed class of information agents the precise structure and semantics of the profile information may evolve over time as the agents grow in sophistication.
- *User control*. The ePerson profile store is a store of information on behalf of one user, not a population profile. It is critical that the profile be trusted by the user. We propose to use policy-based access control, which is discussed further in section 4, below. It also suggests that solutions in which the information is directly under the user's control and pushed to the network edges are preferable which leads us to the following design principle:

- *Decentralization*. In addition to putting the data under clearer control of the user who owns it, decentralization of the platform is a core method for addressing issues of scalability and openness.
- *Partial network connections*. If the user data is held on the network edges, indeed sometimes in individual client devices, then the ability of external software agents to access that data will be compromised by any connectivity limitations. Conversely, client devices wishing to use the profile information to tailor their functions or interactions require access to parts of the profile even at times of intermittent connectivity. Thus, appropriate caching and replication strategies are required.
- *Legacy and external data support*. The ePerson profile is not a universal store. Rather, it attempts to provide an integrated view of data about the user in part by storing the data, and in part by referencing data stored elsewhere. The framework must support integrated access to data in such legacy or third-party services.

We now discuss these challenges, and the design principles they engender, in more detail.

Open Platform

One key motivation for the ePerson user profile is to provide an open-ended service that a variety of software agents and other services can use to better adapt to the user's preferences and needs. The consequence of supporting an such open platform is that it is not possible, or desirable, to fix a pre-specified set of services that will interact with the ePerson.

Our goal of building an open platform stems, in part, from the wider vision of the future of web services. Many commentators and researchers have suggested that IT applications of the future should be constructed from component, globally available, electronic services. HP was an early proponent of this vision of *e-Services* [19]. Other companies have articulated similar visions [20;21], leading to the recently industry moves towards an XML Protocol for web services [22] and associated directory and description solutions [23].

This move towards a common web services infrastructure is relevant to the development of a common ePerson user profile store in two ways. Firstly, general web services are as much in need of access to personal profile information to tailor their functionality and interface as are information agents. Secondly, in our framework, access to profile information is itself regarded as a web service, and specified in such a way that different implementations can interoperate using published web-services standards.

Semantic Extensibility

The role of the ePerson user profile platform is to allow an open-ended set of software agents and web services to access and exchange user profile information. This provides few constraints on the nature of the user profile itself. How can we provide useful structuring for such unconstrained data, without the problem reducing to a trivial lowest common denominator?[5] How can the ePerson define the structure and semantics for the profile information it is offering, in such a way that other agents can determine whether it meets their needs?

[5] It is not uncommon to see profile information only constrained as "keyword-value pairs".

We propose to satisfy these needs by building upon the *semantic web* technology platform. The semantic web is a vision for a future evolution of the World Wide Web to support the exchange of machine processable data [24]. In particular, it is envisioned as a key component of the information fabric through which personal agents exchange requirements and negotiate solutions [25]. The semantic web is based upon a tiered architecture, shown in Fig. 2:

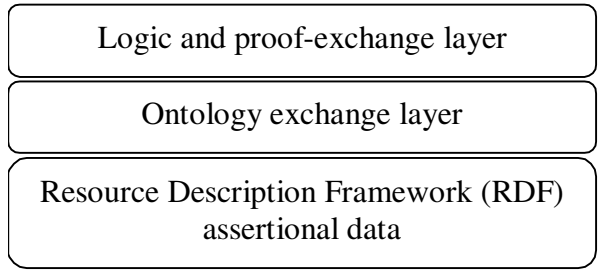

Fig. 2. Semantic web technology platform

The foundational layer, RDF, provides a common format for the exchange of data. It originated as metadata for describing web-addressable resources [26], but is now used more widely. RDF builds upon the XML standard for document formatting by providing a generic data model for graph-structured data, and a means of encoding that model using XML syntax. The data model is designed to support positive ground assertions of the form:

```
<subject, predicate, object>.
```

The `subject` and `predicate` values are described by RDF as resources, which are identified using uniform resource identifiers (URI's) [27]. The `object` value may also be a resource, or literal whose value is specified as a character string. A special case of resources is the *bNode*: a resource node that has no identifying URI. The bNode is equivalent to an existentially quantified variable. While this data model, frequently referred to in the RDF literature as *triples*, only directly encodes binary relations, it is straightforward to encode relations of any arity using the bNode construct.

RDF provides us with a common basis for the exchange of assertions that make up the user profile. The use of URI's as identifiers in the RDF data model allows it to support a weak form of openness: a guarantee that the set of terms in the profile can be extended without names clashing. For example, if one community of agents wishes to express information on, say, sports preferences, it could do so using resource URI's defined within a namespace under that community's control. A second group of agents wishing to record information on, say, news preferences, might then define their own resource URI's within their namespace. The distinctness of these namespaces then ensures that profile information referring to both news and sports will not suffer accidental duplication of identifiers.

However, ensuring distinct identifiers, while necessary, is not by itself sufficient. We also need to be able to discover and re-use extensions to the profile data model. Suppose the second group of agents, above, also wished to record specific sports-news preference information. It should be possible for them to discover that an appropriate namespace for sports information already exists, and build on that to

create a rich profile that is compatible with other sports data. Even if the conceptual model of sports underlying the existing namespace is not powerful enough to support all of the needs of the second (news) agent group, it could still be possible to identify points of correspondence between the two conceptualisations of "sports". This requires each agent group to provide an explicit, formal conceptual model for the vocabulary underlying the profile data they are using. This explicit conceptualisation is called an *ontology* [28].

The semantic web technology framework calls for a standard means of encoding and exchanging ontologies. Once this ontology support is in place, groups of agents and applications can use it to discover conceptual models they can reuse, to formalize (at least partial) mappings between such models, and to extend their domain vocabulary over time. At the time of writing[6], a W3C working group to define this standards layer is still being formed. However, a *de facto* standard already exists in the form of the DAML+OIL language [29]. DAML+OIL is the result of merging the OIL ontology standard, which resulted from the European OntoBroker project, with the early results of the DARPA Agent Markup Language (DAML) project.

DAML+OIL builds upon RDF, both in the sense that RDF ground facts can be used to provide instance data for the ontology, and in the sense that the ontology itself is encoded in RDF. This makes the ontology language openly extensible as well. DAML+OIL supports the declaration of schema information including class and property hierarchies, axiomatic relationships between class expressions, and restrictions on properties. DAML+OIL ontologies can refer to other ontologies permitting the modular construction of extended conceptualisations. The semantics of the language are precisely defined, using description logics as a formal foundation [29].

Our ePerson profile store supports storage of, and access to, profile information in RDF format. It associates that RDF encoding with a corresponding DAML+OIL ontology. An agent can query the store to determine if information is directly available in an ontology that it understands. In the case that a direct match is not available, we anticipate being able to use the emerging semantic web infrastructure to discover ontology mappings. Such mappings can provide at least partial transformations from the native ontology of the profile source to an ontology understood by the profile requestor.

We should add one qualifier here, which we see as an important direction for future research. The ontology approach to agent communication is well suited to reasoning with user information expressed in symbolic terms. For example, a taxonomy of sports, against which to express preference information, is easily expressed in this way. However, non-symbolic or sub-symbolic information will also be required. For example: a numerical score corresponding to the users interest in the given sports category, as extracted by a probabilistic learning algorithm. Existing ontology languages, including DAML+OIL, cannot express the semantics of such scoring schemes. Furthermore, in some cases, the categories themselves may not fit the symbolic, logic-based approach. For example, class-membership might be a numerically weighted membership property, rather than a crisp symbolic assertion. In the longer term, bridging symbolic declarative models with sub-symbolic and probabilistic models will be necessary to achieve a complete and useful user profile representation scheme.

[6] November, 2001

Decentralisation

Our framework places the ePerson profile store for a given user directly under the control of that user, and stored towards the network edge. This approach is agnostic as to whether the user runs a profile service directly on a personal machine (or distributed across a set of personal machines), whether they share use of a work-group or community-level server, or whether users subscribe to an independently managed web service offering profile management to many people. However, from the point of view of client agents, they cannot assume that all profiles are stored in the same place and need to treat each user's profile as if it were stored separately.

Network-edge services offer strong advantages. For the user, it gives them more flexibility, allowing them to take as much or as little control of their ePerson data as they wish. In particular, users with a strongly-felt need for maintaining personal privacy may prefer to host the profile service, whereas others would prefer to trust a third party to manage the service on their behalf. A decentralized storage system may also present a less attractive target for hacker attacks. For the system overall, decentralization increases scalability. There is no single point of storage, or access, that might become a bottleneck. It also helps to ensure openness: if anyone can create and run a conformant profile service, then the whole system is not dependent upon the actions of a single supplier (whether commercial or open source).

There are, however, drawbacks to a decentralised approach that must be overcome.

Firstly, in the absence of a single access point for a directory of profile services, there is the problem of service discovery. We can envisage many potential mechanisms for dissemination of profile server addresses. For example: they can be embedded in personal home pages, discovery by web robots, direct transmission via personal email, vCard-like encoding in electronic business cards [30], or embedded as extension attributes in X509v3 certificates. However, as with all such network technologies, we would expect an initial period of uncertainty before the successful formats and mechanisms emerge.

A second difficulty is that some uses of profile information currently expect to have access to centralized population statistics. For example, consider a collaborative filtering approach to recommending information sources [31]. Given a centralised population database from which recommendations are drawn, a recommendation agent could consult the ePerson profile, match that against the population data and make suggestions to the user. However, collecting data centrally is contrary to our goal of keeping users' profile data distinct and under privacy policy control. One of our long-term goals in this research is to develop decentralized solutions to services based on population data. For example, a recommendation agent might traverse a community of distributed profile stores, incrementally building up a statistical picture of the population without duplicating the raw data. Each interaction with the ePerson profile would negotiate access to the profile data under the terms of the user's privacy policy. Ultimately, the entire clustering and analysis process might be fully distributed using appropriate algorithms. See section 0 for more details of ongoing research into distributed social filtering algorithms.

Partial Network Connection

Increasingly, users interact with the information on the web (and their organisational intranets) through a variety of means. These include multiple personal computers (e.g.

one in the office and one at home), PDA's, mobile phones and public Internet cafés. Maintaining continuity and consistency across these different touch points is a frustrating experience for many users [32]. We can identify two core problems for the ePerson framework to address. The first is that as they user moves through their day, performing different tasks in a variety of contexts, they will use different client devices to access the information space on the Internet. We cannot rely, therefore, on storing user information in one client device. Further, we can assume that user profile information will be generated in multiple places, generating a problem of consistency. This scenario will be familiar to anyone who has attempted to maintain consistent sets of web bookmarks on multiple computers. The second problem is that a proportion of the client devices a user has access to will have intermittent connectivity to the network. This will vary by each user and their own situation, but, in general, we cannot assume that we can simply store all profile information in some Internet-based store, and expect the user always to be able to access it.

One approach to ensuring consistency of access to profile data is under investigation in HP Labs by Stephanie Riche et al [10]. In this project, consistency of user profile across multiple client devices is seen as analogous to the general problem of cache consistency. By investigating different usage patterns for profile data (such as whether it tends to group within the profile or be diffuse, or whether profile elements tend to migrate to the user's current interaction point), Riche et al have identified the characteristics of suitable cache consistency and coherence algorithms. Ongoing work is testing the performance of different algorithms under different operating conditions.

The ePerson platform will incorporate results for maintaining profile consistent across different access points as they become available. One other characteristic of the ePerson approach, shared with some other user agent designs, provides additional user benefit when network connections are intermittent. Given enough information, there are scenarios in which the user agent can respond to incoming requests on behalf of the user, when the user is unavailable or busy elsewhere. For example, a calendar-aware agent can answer queries about the user's ability to attend meetings on a give date, based on information from the user profile.

Legacy and External Data Support

The framework as discussed so far calls for all profile data to be accessed as positive ground assertions using the RDF data model with the underlying conceptual models being made explicit by encoding using DAML+OIL. We do not propose, however, that it is appropriate to store all user data within the ePerson profile, nor to store all data natively in RDF. A useful metaphor for the user profile is as a *meta-service*, part of the role of which is to provide an integrated view of user profile data, which may include pointers to data services residing elsewhere, or a transliteration of data stored in other formats.

Many existing data sources could be usefully regarded as part of the user profile, but do not natively use the RDF/DAML+OIL format. Our approach to this is to provide mediators [33], that map those legacy or external data sources to the semantic web format. In our current design, we see this primarily being a one-way access: client agents can request information from the legacy sources via the RDF mediator. However, in some cases, the legacy sources will continue to be updated. In these cases, the ePerson could provide the web service descriptors for the external data

sources, that would allow the user profile to be updated directly by the client agent. For example, consider calendar or diary management. One approach would be for the ePerson profile service to provide an ontology for calendar entries, and provide mediators between RDF queries and any native calendar information stores. However, this might not always be practically feasible. Alternatively, the profile service might provide the address and description of the native information stores directly, using a standard web-services interface.

Summary of Design Principles

The ePerson profile store provides a common access protocol for user profile information in RDF format. It allows agents to query for profile information against an explicit ontology, and can potentially use ontology mappings to transform natively stored information into the requested ontology.

The store can be accessed using as a web service, using either remote procedure call (RPC) semantics (e.g. using SOAP [22]), or using stateless protocols, such as HTTP. The choice will be application and environment specific, and will be based on factors such as the expected latency and reliability of the network connection. In either case, each user may be using a different physical store and all of the access agents need to negotiate the terms of access and update.

Issues of how the stores are identified, how the access policies and expressed and enforced and how user authentication is handled without violation of privacy are key issues in the overall framework and are reviewed in the next section.

4 Privacy Protection and Controlled Disclosure

Introduction

Increasingly, individuals, corporations, governments and other commentators are raising concerns about the protection of personal information held online. Many researchers into personal agents and information agents cite user privacy as a concern, yet, to date, little research has been applied to ensuring privacy protection for these technologies. In society generally, and the Internet industry in particular, new technologies and legislative frameworks are being advanced that can enhance privacy (though sometimes the reverse is true). The legal and moral debate about the trade-off between strong privacy and the general well-being and security of society at large will continue. Our goal with the ePerson is more modest: the protection of personal information from unwarranted intrusion, and from the unwanted consequences of such intrusion – for example unsolicited advertising (“spam”).

Recent innovations around e-commerce in digital content (e.g. streaming video on-demand, electronic books, on-line music) compound the privacy problem, thanks to electronic tracking and user authentication for the purposes of copyright protection. These make the task of gathering of personally identifiable information easier, and hence increase the extent of potential privacy violations. Users must have trust that their personal information will be protected by the agent, in order to undertake to disclose it in the first place. For example, if agents are given detailed information to then autonomously carry out tasks on the user’s behalf, the user should be confident

when, where and to whom such information will be disclosed – or, at least, that it will only be disclosed appropriately.

In this section, we outline some of the general issues for protecting the privacy of the user of adaptive agents, and discuss some of our solutions in the context of the ePerson.

Privacy and Minimal Information

For excellent material on privacy and its importance for the consumer, the corporation and the state, see the publications of Ann Cavoukian [34;35;36;37]. In particular, a simple and reasonably complete definition of privacy may be found in [38]:

> *"Personal control over the collection, use and disclosure of any recorded information about an identifiable individual"*

For our purposes, we consider privacy to consist of protection of (a) personal identifiers and (b) profiles (recorded information) associated with personal identifiers. The protection of personal identifiers presents a dilemma in the context of personalised services: how to get personalisation, while restricting the disclosure of who you are. Below, we describe various anonymity and other pseudo-anonymity techniques to address this problem. Protecting profile information amounts to controlled revelation of information, with conditions on future use of the disclosed data. In both cases, it is instructive to consider the *minimal information* that is necessary in any given transaction.

The notion of minimal information transfer has been an important principle for privacy in the non-digital world [39]. We consider that it will continue to be important in the digital age. The principle itself is simple to state: any information requested that is not *necessary* to complete the explicit purposes of an interaction with an agent or a service, needs to be called out as such by the requestor while making the request.

However, conforming to a minimal information principle is a difficult goal even in the cases when the requestor is well intentioned. For example, a transaction for the purchase and delivery of a physical good has very well defined minimal required information: (a) electronic payment information (b) delivery destination. On the other hand, minimal information for a gift recommendation is not well defined, and greater information transfer may or may not result in an improved recommendation. As the use of minimal information has been very successful as a good business practice in the non-digital world, it is worth trying to formalize the return to the user for revealing more information, in cases where minimal information cannot be easily determined.

Overall Privacy Approach

In our overall approach, the end user gains direct control over their private information through the application of the following principles:

- *Control of credentials and identifiers for user authentication*
  The degree of user anonymity is determined by the manner in which a user is deemed qualified to participate in an interaction. The qualification process itself reveals information about the individual, and determines the extent to which the information exchanged in a particular interaction can be linked to information produced by the same user in another interaction.

- *Control of which elements of the user profile are revealed and to whom*
  Following the principle of minimal information, the requestor should be able to identify which profile elements are necessary for a transaction, and which are optional to provide an enhanced service. For example if an interaction will result in physical delivery of items, the user's address would be part of the minimal information. Further, a user may be agreeable to the release of the minimal information, or may opt not to participate in the interaction, and ought to be able to make that decision. Lastly, he or she may be willing to release more than the minimal information, and that, too, ought to be possible.
- *Control over what the privacy policy for later use of this data is*
  Privacy is enhanced if the user has control over the conditions under which the credential and profile elements are revealed. For example, may the requestor sell the information to all types of buyers without restriction? To some types of buyers? May the requestor share the information with other business associates? Should the requestor destroy the data once it has been used for the current purpose? For what uses may the information ever be used? Should the information be used only when aggregated with similar information from a certain minimum number of other users? Note that, technically, we can only reconcile the *declared intent* of the requestor to the user's wishes. A separate physical auditing process [40]or technical auditing mechanism will be necessary to determine compliance with stated intent.
- *Control over the accuracy of data revealed*
  Conventional privacy techniques typically involve making a simple decision to either reveal a piece of profile data, or not, or reducing the precision of the data[7]. *Variable privacy* provides an alternative: the data is revealed, but it may or may not be accurate. By subjecting the data to deliberate corruption, according to a defined distribution, the data has reduced utility for making assertions about a specific user. However, it regains value when aggregated with similar data across a large population of users.

The next section describes the privacy-related elements of the ePerson framework, showing how we put these principles into practice.

EPerson Privacy Framework

In this section, we discuss more specifically about the ePerson framework for privacy: identity and authentication, policy engine, trusted aggregators and variable privacy.

Protecting the Identity of the User

In order to gain access to a particular user's profile, a third party (say a web site) needs to indicate to the profile store the *identity* of the user in which they are interested. Thus, the user needs to present some form of *identifier* to the web site, which can be passed on to the profile store and used to locate the user's profile.

In the simplest case, this identifier may be global in scope and persistent over a long period. An example of such an identifier is a username, or an email address. This scheme affords the user little in the way of privacy protection, since the same

[7] The statement "I earn between $20K and $40K per year" is less precise, and better protects privacy, than "I earn $25,600 per year".

identifier is given to each site, allowing long term tracking of the user, and correlation of the user's activities between different sites.

Privacy protection can be improved in several ways.

The user may be permitted to create and manage multiple separate identities, which may or may not reference the same user profile. They now have the choice as to which identity to present to which site. This could be used, for example, to separate work activities from non-work activities, by explicitly creating a different pseudonym for each. This behaviour is visible today in data available on the users of another.com [41], of which 80% are under 25s. This data shows, on average, that males have two active email addresses and females have four.

In general, there is no need for the identifier to be global in scope. The identifier also need not even be persistent over time. As long as the profile store is able to map it back to a specific user, then that is sufficient. However, unless the identifier is persistent over time at a given site, that site will not be able to recognise the user returning, and so personalization options will be more limited. Depending on the privacy preferences of the user, this behaviour may or may not be desirable. The following scenarios expand on these ideas, and afford the user better privacy protection than a persistent global identity.

1. At the start of a session, a user would authenticate them self with the profile store. This results in a random session identifier being allocated to the user, which only the profile store is able to map back to the real identity. Because the identifier changes each session, it cannot be used by the web site to track the user over long periods.
2. The user may have a single, global identity, but that is never used directly. Instead, a client side process generates a different identifier for each site visited. This would prevent users activities being linked between sites. This site-specific identifier would have to have the property that the profile store, and only the profile store, can map it back to the global identity. This is straightforward to achieve using public key cryptography.
3. The combination of (1) and (2).

In the above scenarios, the user authenticates with the profile store directly. However, identity services are starting to emerge which enable people to use a common identity (username and password) across multiple web sites. This avoids the need to register with each site visited, and in particular avoids the need to remember multiple usernames and passwords. These services are also referred to as single sign-on or unified sign-on services. Microsoft's Passport [42] service is the most prevalent example today. However, similar services include AOL's Magic Carpet, currently begin promoted at the screen name service, and in the near future a similar offering from the Liberty Alliance [43].

These third-party identity services also aspire to own the user's profile. However, we feel that a separation between the identity service (responsible for authentication) and the profile service (responsible for managing the user's profile) is logical, and beneficial to the privacy of the user. Thus, our intent is that the profile store not become an identity service in its own right, but instead be flexible enough to work with these third party identity services as they evolve, particularly ones that place more emphasis on privacy protection.

Policy Engine

The policy engine (see Fig. 1) implements fine-grain policy-driven access control over the user profile data. The user can establish a set of policies controlling who may access each part of the profile, and for what purpose. The profile store is organised hierarchically, and policies may be attached at different points in the hierarchy. In general, any given request will be subject to the set of policies encountered as a path is traced from the request point back to the root node. The policies at each level of hierarchy need to be aggregated, and any conflicts that arise must be resolved.

The policies will generally specify that some form of evidence needs to be presented by the requestor before access is granted. The evidence will constitute proof of who, or what, the requestor is. More specifically, the evidence may demonstrate the actual identity of the requestor, or it may demonstrate just that the requestor posses certain relevant attributes. The policy engine compares the evidence provided by the requestor against the policy covering the requested data. If the evidence satisfies all the policy pre-conditions, then access is granted. If insufficient evidence is provided, the request will be denied, and a reason stated. The requestor can then decide whether to supply the additional evidence, or simply abort the request.

Requests to access data from the profile store, and the policies to control that access, will be based on a policy language, some elements of which are outlined below.

- *Request path:* identifies the sub-set of information within the profile to which this policy applies by describing a sub-graph of the profile structure.
- *Request permissions:* specifies the access permissions to the profile data under this policy.
- *Usage policy:* this (machine-readable) policy dictates the terms-and-conditions covering how the requestor may use the profile data disclosed. It is returned to the requestor with the profile data.
- *Identity pre-conditions:* specifies the type of identity disclosure required. Restrictions may be placed on the technology used, and, once validated, the identity may be compared to a list of authorised parties or roles.
- *Attribute pre-conditions:* specifies the type of attribute disclosure required. Restrictions may be placed on the technology used, the specific attributes required and the range of acceptable values.
- *Post processing*. This section specifies what, if any, post processing should be performed on the profile data before returning it. Post-processing allows the profile data to be modified on the fly, for example to implement systematic lying needed to support variable privacy.
- *Audit and logging*. This section specifies what type of audit trail should be left when this policy is used. For example, failure of a requestor to meet all the pre-conditions may result in the user being notified by email.

This policy language shares some design goals with P3P [44]. However, there are some differences: P3P policies do not specify post-processing, auditing, or different ways in which to identify or authenticate requestors. P3P also does not have a means of identifying a hierarchical data structure and using it for specifying access/usage policy. P3P assumes that the data collection happens as a result of a user approaching a web site, while we need a policy language for the purposes of controlling access to data for a variety of purposes.

Attribute Based Evidence from the Requestor

When creating policies controlling access to the profile store, the presence of certain attributes may be more important than the actual identity of the requestor. For example, a user may want to limit access to part of his profile to web sites carrying either the TrustE privacy seal [40] or an acceptable P3P [18;44] encoded privacy policy. To gain access to the profile, the requestor must disclose these attributes, rather than just their identity.

One way of demonstrating attributes is to first demonstrate identity and then show a secure mapping from that identity to a set of attributes. This might be done through a secure directory service. Alternatively, additional X509 certificates could be issued by a certificate authority to the same effect, since X509 certificate extensions can be used to carry attributes. However, we are particularly interested in techniques that allow attributes to be demonstrated without first disclosing identity, since this improves the privacy protection of the requestor by allowing them to remain essentially anonymous. We propose using some form of anonymous or pseudo-anonymous attribute credential.

One scheme developed by members of our group involves user identity comprising a large pool of public/private key pairs, rather than just a single key pair [45]. Each attribute certificate is bound to a different public key drawn from the pool. If the pool is large enough, it then becomes difficult to link different attribute certificates back to the same user.

A more complex, but more powerful, scheme of attribute certificates has been developed by Stefan Brands [46]. This is a generalisation of Brands' research into anonymous digital cash. It is based on protocols that enable the blind issuing of attribute certificates and the selective disclosure of their attributes. Blind issuing prevents the certificate authority (CA) from being able to track the use of the credential, by ensuring nothing in the final credential can be linked back to the CA's view of the issuing protocol. Selective disclosure allows the owner to control which of attributes within the credential are disclosed. This is analogous to being able to erase fields in a paper credential before showing it. Further, it is possible to include additional attributes in the credentials, which act as a strong disincentive to sharing credentials.

Trusted Aggregators

The profile store allows trusted components to act as aggregators, or summarisers, of profile information. These components are authorised to read raw data from the profile, process it in some way, and write back the results to the profile. An example (described in more detail in section 0) is an aggregator that takes raw browser click stream data from the profile, and attempts to map this to a set of "current interests" for the user. The "current interests" part of the profile could be used, for example, by news web sites to provide the user with personalised news. It is unlikely that the user would have permitted these sites access to the raw click stream data.

Variable Privacy through Systematic Lying

For the exchange of usage information at a particular aggregation level in return for discounts and services, one needs a means of ensuring that information collected is indeed averaged. One way of doing this is to reveal incorrect information with a

secured probability of truth for binary or multiple-valued information, and a secured variance for continuous-valued information [47]. Because the information is incorrect, it is not worth much by itself, and is valuable only in aggregation. This forces data collectors to aggregate the information. The accuracy of the calculated (i.e. aggregated) statistical information depends on the probability of individual answers being true when they are multiple-valued or binary-valued, or on the variance of the answers provided if they are continuous-valued. More uncertainty in the answers implies that more information has to be aggregated to obtain a particular level of accuracy.

The degree of accuracy affects the value of recommendations and ratings determined using this data. It also affects the value of this data to the data collector. Thus, it provides a manner of making explicit the degree of privacy violation as well as the returns (recommendations, other services) obtained through the privacy violation. It may be used as a substitute in cases where minimal information is not determinable.

5 Application Examples

Here we consider some of the applications that are enabled by the ePerson framework. Some of these we have working prototypes of, while others are in development or at the conceptual design stage.

History Store

The first application example we will describe is called the *personal history store*, and is in use as an experimental prototype in our Bristol research lab. The personal history store collects data from the web pages visited by the user through one or more web browser sessions. It takes its name from the history capability provided by most major browsers, but extends the capability of the standard history listing by collecting additional metadata, including full-text keyword indexes on the pages visited. This enables the user to pose requests such as "I recall that sometime last week, I visited a web site about Java implementations of collaborative filtering algorithms and I want to go back there". It thus provides a powerful search capability, analogous to search engines such as Google [48], but personalised to the user's own interaction history. The personal history store comprises a number of functional elements: data collection, history search, and data aggregator, which we outline further below.

Data collection Data collection in our current prototype occurs through inserting a profile-capturing service between the user and the wider web. This is implemented as a web proxy, that is pointed to by the user's web browser proxy settings. The history store then keeps a record of each web page visited. Many users are quite sensitive about their web browsing patterns. Thus, even though the history store web-proxy is a shared resource, the standard behaviour is to partition the store by user. In principle, the partitions could be encrypted with a user-controlled key, but we are not doing this at present. For each page visited, the title, URL, date, and raw page content are stored. In addition, the page content is converted to plain text and passed to a keyword-indexing package[8].

[8] We used Lucene [86], now part of the Apache Jakarta project [87]. Lucene is a high-performance, full-featured text-indexing engine written entirely in Java.

Due to the volume of data maintained, we chose to implement the history store as a profiling service, outside the actual profile itself. Data is available from this service in RDF, but RDF is not used as the storage mechanism.

History search service A web interface provides a search capability, similar to standard web search engines. The search algebra supports access to the meta-data captured by the data collector, Boolean expressions, phrase queries (including wildcard terms), and temporal constraints. Using this interface, users can easily return to pages they have browsed in the past. As the history store keeps cached copies of pages; the original content is still available if the original site is down, or has changed. These cached copies can be aged as necessary to reduce storage overheads.

Trusted aggregator Within the user profile framework we have created a trusted aggregator whose purpose is to mine the history store and attempt to determine a set of current-interests for each user, based on their web-browsing history. The classifier tries to match the page URL against a local copy of the DMOZ [49] database. This database contains 2.7 million URLs organised into over 370,000 topic categories. If a match of the complete URL fails, then the path part of the URL is gradually truncated. If there is still no match, we apply some simple re-writing rules to the host part of the URL. This approach has proven to have a high probability of generating a match. We maintain counts of how many URLs map to each DMOZ topic category. After all of the user's URLs are classified, the trusted aggregator writes any non-empty topic categories and weights to the current-interests section of the ePerson profile. These aggregated categories may then be governed by different privacy rules than the raw data itself. Since its functionality is well defined, the aggregator is trusted by the user, and hence granted privileged access to the make changes to the profile.

In summary, this example illustrates the use of a trusted aggregator to distil highly sensitive private click-stream data into a set of current interests that may be more openly shared.

Web Assistants

The remaining application examples in this section describe illustrative scenarios using the ePerson platform. The intent is to ground the abstract platform descriptions, above, in terms of use cases and scenarios that illustrate how users could benefit from the use of ePerson-based information agents.

One category of information agents that has been widely investigated is the web-browsing assistant. Some examples (among many) include Letizia [50], WebWatcher [51], Personal WebWatcher [52] and WebMate [53]. Generically, the web-browsing assistant can play two roles in relation to the user: a *mediator* for the user's web-browsing experience, or a *benign observer*. The difference being, in essence, whether the user's interaction requests go direct to the agent, which then interacts with resources on the web to satisfy the request, or whether the agent "looks over the user's shoulder" [54] and makes suggestions in a parallel user dialogue. Partly this choice comes down to available technology. Usability studies on personal web-browsing agents have faced difficult choices in the past between using custom, instrumented software, delivering more accurate data, and standard browsers that the users are comfortable with and contain their personal bookmarks. Both the mediator and assistant design concepts have serious design challenges to address, and would make use of the profile store in different ways. For brevity, we consider only the assistant further in this paper.

Assume that the user has browsed to a particular web page. There are various ways in which the web assistant can support the user's information gathering needs. First, we assume a minimum of context is available for the current task from the user profile. The tasks that the agent could perform include:

- Looking ahead to the links on the page, and identifying those that match well against the user's profile of interests, or which score highly in social-filtering measures based on the user's interests. If the agent is able to collaborate with a network-edge proxy service, through which the user's browser retrieves web pages, this identification could be done in-line by highlighting words, inserting additional mark-up or generating links in a browser side-bar. Alternatively, the agent can pick out the hyperlinks in a separate window, perhaps displaying them in rank order [50].
- Identifying sub-sections of the page that match well with the user's interest profile, again using page modification as a means of feedback [55] or generating a section index in a parallel window.
- Auto-generating a site map for the current web site, using profile information to highlight regions of the site that might be of high interest.
- Use feedback from the user, either explicitly or through inferences made on observations such as time spent on a page, to refine the user profile.

With additional information about the user's context, more assistance can be given. For example, the user may choose to indicate to the agent a current goal or role. There are clearly usability issues here: users typically will not want to interrupt the flow of their interaction to engage in extensive consultations with an agent. Our design goal must be to capture as much information as possible for as little cost to the user as possible, and attempt iterative refinement to improve the resolution on the user's need. A good starting point is that users do attempt to express an approximation of their goal to a search engine during an information discovery or retrieval task. Observing or mediating this dialogue would give a good approximation to an initial goal for the context. Given this context, the agent can do a better job at the above tasks, and can provide additional refinements –such as thesaurus or ontology search on terms from the context [56].

An advantage to the user in having their information stored in a network-edge service, such as the ePerson, is that it allows them to move between different web interaction points and maintain a degree of continuity. Many people experience the frustration of attempting to keep sets of bookmarks consistent between a PC in their home and one in their place of work. As the information in the user profiles grows in size and value to the user, maintaining this consistency will only grow in importance to the user, along with the need to keep that information secure and private.

As the information in the user profile accumulates, additional services become possible. Consider a computer science researcher, attempting to keep abreast of developments in his or her field of research. Various tools exist for discovering, say, relevant new articles and papers (e.g. the monitoring option at ResearchIndex [57]). However, such tools never have complete coverage, and still require the researcher to evaluate papers that are discovered to see if they are interesting and truly relevant, and to devote effort to keeping the matching filters up to date. If a profile of the researcher's professional interests was maintained in their ePerson profile store, then, assuming this was permissible under the privacy policy, this data could be used to discover a community of researchers with similar interests. References to papers

discovered by one member of the community could be propagated about, sharing the load of the discovery among many peers. Furthermore, the use of *social filtering*[9] could reduce the burden on the user to read and evaluate every resource discovered, by using the ratings assigned by members of the community.

Arguably, it would also be useful if the policy language used to describe the user's preferences regarding the sharing of personal data incorporated constructs that would allow trust networks to be built up. In standard social filtering algorithms [7], every rating has equal weight. In reality, we typically find ourselves placing more trust in particular individuals – so called *opinion leaders* – than we do in others. Social filtering algorithms aim to build robust recommendations from the statistical properties of large populations, but are brittle when the amount of data is limited. A trust network that overlaid the (dynamically discovered) community-of-peers could partly address this limitation, by allowing users to gain robust recommendations from fewer data points.

Distributed Collaborative Filtering

A class of powerful end-user applications, such as collaborative news recommendation, are enabled by the application of social filtering techniques in conjunction with the privacy-enhanced profile store. Social, or collaborative, filtering (henceforth CF) is a term for a broad range of algorithms that use similarity measures between individuals in a population to generate recommendations. The information we collect and store in the user profile provides a rich source of data for such similarity measures. However, as we mentioned in section 0, standard CF algorithms require modification to work with decentralised profile data and privacy protection rules. In this section, we summarise the distributed collaborative filtering problem, and discuss our work on artificial immune system algorithms.

Most CF algorithms work with statistical data. With suitable similarity measures, CF algorithms allow us to directly address the problem introduced earlier, regarding the need to include both mentalist and sub-symbolic models. The simplest approach to social filtering is the *k*-Nearest-Neighbour algorithm, which uses a given similarity method to define a neighbourhood of at most *k* highly similar users. The votes from these users, suitably weighted, are used to make predictions and recommendations. Many improvements on this method have been investigated in the literature (see, among many examples [58] and [31]). Even assuming such improvements however, these methods assume the availability of a centralised collection of profiles to iterate over. In our decentralised framework, we do not make any such assumption.

Viewed abstractly, the task of collaborative filtering can be reformulated as the problem of finding a suitable *neighbourhood* of similar individuals. The metaphor can be further generalised into the task of forming a *community*. The challenge is to provide mechanisms that will allow the build up of an *appropriate* community, the definition of which may be task-dependent. However, we assume no central authority or oracle: it is up to the members of the community to discover themselves.

We have built an experimental recommender prototype to begin to address these issues. More details can be found in [59]. The recommender prototype is based on a novel approach for discovering affinities in a data set called the *artificial immune system* (AIS) [60]. We apply AIS to the problem of community discovery. In brief, we

[9] Sometimes called *collaborative* or *clique-based* filtering.

encode a user profile as an antigen, and potential neighbours as antibodies that bind to this antigen. Neighbourhood formation is dynamic and depends on both antigen-antibody and antibody-antibody interactions. The basic elements are as described below.

Data is collected from individual user profiles to create antigens and antibodies. There is a variety of sources for this data, from publicly accessible preference data [61] to our own experimental bookmark managers or the personal history store (above).

Typically, a neighbourhood will gradually build up, with most antibodies leaving almost immediately and only a few persisting. At this early stage, antibody-antigen interactions are dominant. As the neighbourhood grows, antibody-antibody interactions, based on concentration, will have an increasingly noticeable effect.

To achieve distributed community formation, the antigen can be passed around a population. Each ePerson adds an antibody to the evolving AIS, until a sufficiently large neighbourhood has formed. This idea is similar in spirit to the construction algorithm proposed by Delgado [62], and to the idea of a distributed query, as used in Gnutella [63]. However rather than broadcasting the antigen, each individual propagates the antigen via a local narrowcast, based on some rational criteria.

Thus, we can see that a virtual community can be cached, learnt or recalculated as required. In principle, there is no reason why a single ePerson should not belong to multiple communities at the same time. For example, the use of different metrics (film preferences, mailing lists, homepage analysis) may produce different communities. Adamic and Adar [64] show this effect for a university social web. In addition, we have shown that the parameters used to select a community, e.g. the importance of diversity, will provide different community characteristics [59].

6 Related Work

As discussed above, user modeling, personalisation, and adaptation are well-researched topics with a significant body of prior literature. Rather than attempt any complete review of relevant prior work, we here summarise some of the key projects that share at least some goals with our own. This review is illustrative, rather than necessarily complete or representative.

Stuart Soltysiak and Barry Crabtree report the use of user profiling as a key component of a range of personal agents developed at BT Labs [65;66]. Independent agents including Grapevine (interest-based matchmaking in an enterprise), Pandora (proactive topic suggestions) and Radar (information presentation) all share a common user profile. The profile itself is represented as a TFIDF vector [67], which uses the frequency of occurrence of terms as a measure of the user's interest. Terms that occur more frequently than they do in the domain corpus are picked out as representing the user's interests. The TFIDF representation acts as a common representation formalism between different agents, and is quite flexible in that many different sources of information (favourite web pages, saved emails, documents, etc) can be mapped to a single form. It is limited in expressive power however, and cannot, for example, easily hold demographic data or other symbolic information. Soltysiak and Crabtree do acknowledge the importance of maintaining privacy of the user's profile, though the published papers do not detail their approach to this topic. The success the BT group has had in sharing information among different user agents

using only the TFIDF representation is a testimony to the utility of sub-symbolic representations, and it is clear that any general profiling scheme will have to encompass both styles of representation.

The use of ontologies in adaptive web navigation has been investigated by Jason Chaffee and Susan Gauch [56;68]. In this work, ontology is taken to refer to the hierarchy of categories used by Lycos to organise their web directory, rather than the direct application of an ontology language such as DAML+OIL. The ontology also includes a selection of characterising web pages for each category, so that it is possible to compare other organisation schemes (such as the user's bookmark hierarchy) to the reference hierarchy. The mapping derived from this comparison can be used to assist translations between the user's "personal ontology" and the system ontology. While this approach is interesting, and the use of characteristic documents to aide the translation process is certainly intriguing, we would hope that using an ontology language with a more formal basis will provide a sounder and richer basis for performing such translations, as well as providing the basis for an extensible representation scheme.

Self-contained user -modelling components or services have been investigated by several authors, though few have enjoyed significant use outside the laboratory where they were developed. Examples include BGP-MS [16], Doppelgänger [69], Tagus [70] and UM [71]. All of these systems pre-date recent developments in open representation systems such as RDF and DAML+OIL, and so do not emphasise extensibility as a design goal.

Some basic user-profile ontologies have been published for DAML+OIL, though at the time of writing these are relatively immature. Such ontologies do not compare to the user-modelling tools outlined above, since they define only the terms used and not the reasoning processes and other supporting services that comprise a complete profiling system. Published DAML+OIL ontologies are available from [72]. An alternative approach was taken by the Customer Profile Exchange (CPExchange) consortium [73]. CPExchange is a custom XML format for representing a range of customer data typically held in enterprise databases and customer relationship management (CRM) tools, with the goal of being able to exchange data between such stores. Despite the use of XML, the CPExchange format suffers from inflexibility and limited expressive power, and has not so far gained widespread use. In addition, it is not freely available but must be licensed from the CPExchange consortium, and is therefore unsuitable for our purposes. It would, however, be technically straightforward to map assertions from our DAML-based ontology to the CPExchange format and vice-versa.

A more complete and grounded use of ontologies to assist user interaction is given by Weal *et al* [74]. Here, the emphasis is on the use of ontological descriptions to generate links in a hypermedia space dynamically. Although the system is based on an agent framework, this is not apparent in the user's interface. Instead, agents are used as infrastructure to provide the dynamic behaviour of the system. The ontological information is held in steady state while the links provided to the user are dynamically generated. In our approach, we gather new information related to ontological categories, and new categories, as the user interacts with the system and performs information gathering and management tasks.

Privacy protection in the ePerson framework may be expressed in terms of policy expression, protocols and policy compliance – similar to the use of these terms in [75;76].

The policies that govern the use of collected data and associated credentials need to be expressed in machine-readable form. It is clear that the language used for the privacy policies needs to be open and extensible. At the moment, there are two candidate languages: W3C's Platform for Privacy Preferences (P3P) [18] and Zero Knowledge System's Privacy Rights Management Language (PRML) [77]. P3P is an XML-based specification and is tailored specifically for the collection of regular consumer information by websites. So, for example, it does not include detailed handling of usage profiles. W3C has an associated specification for a language, APPEL [78], meant for specifying client-side privacy preferences incorporated into the browser. When a consumer visits a website, the browser indicates whether the site has a P3P policy, and whether the terms of the policy match the preferences set by the consumer. Zero Knowledge Systems has not made public details on PRML, so not much is known about it. PRML is expected to address the expression of policies associated with data collection laws in various countries. If this is all it addresses, it too will not be sufficient for our needs. If, however, it is extensible, it might be possible to extend it for our purposes.

Changing the accuracy of collected data for the purpose of maintaining privacy has been described in the statistical database [79;80;81;82] and the data mining [83] literature. In these fields, it has largely been used with continuous-valued data, and the experiments reported were performed on accurately collected data which is perturbed after collection. The work on continuous-valued data demonstrates that considerable statistical information can be gleaned with reasonable accuracy from the perturbed individual pieces of information. In our system, we plan to use the technique for both continuous and discrete-valued data, and propose that the data be perturbed by the user at the time of collection, with a user-specified degree of perturbation. The results from the data mining and statistical database literature are consistent with our expectations that our system will enable the user-controlled release of single units of information with varying degrees of privacy, such that the information may be used in different ways, for different purposes, through dynamic aggregation.

In some respects, Microsoft's ".NET My Services" [84] component of the .NET architecture shares similar goals to the ePerson platform. .Net My Services relies on a single, unique user identifier authenticated by Microsoft Passport, a variant of the Kerberos protocol. Users authenticate once to Passport, and are then able to authenticate themselves to Passport enabled web sites. .NET My Services has an extensible data model for profile information, based on XML. However, while it is clear that a given application or service can store information in the .NET My Services profile, it is hard to see how this data can be shared between applications without some additional support for ontology description. The Liberty Alliance [43] has a goal of providing an identity service similar to that provided by Passport, but is positioned to embrace open standards and the federation of independent authentication services. However, at the time of writing the detailed work programme for the Liberty Alliance project is not available.

7 Conclusions

In this paper, we have outlined an approach to providing general, open information agent services to the user based on a collection of key enabling technologies. Many of

these approaches have been investigated in isolation; we propose that together they constitute a novel basis for the development of robust, usable agents. In part, this robustness arises from the use of emerging standards and technologies for next-generation Internet applications: RDF, DAML+OIL and extensible, interoperable web services. In part, it arises from driving the basic requirements for an agent framework from user priorities – hence our emphasis on privacy and user control.

The ePerson platform brings these technologies together as a coherent platform. It is also, however, a key abstraction in the design of larger scale information handling and processing applications. As an element of the design of such applications, the ePerson is the locus of user identity, profile information and preferences. It also provides a platform for autonomous information-based processes, such as web-browsing assistants, that ultimately will assist our users to become more effective in their information handling tasks. Ongoing research work in our laboratory is focussed on generating practical end-user applications based on the ePerson framework.

Acknowledgements. The authors would like to thank colleagues at HP Laboratories, discussions with whom have helped shape the ePerson agenda. Particularly we would like to thank Joe Pato, Gavin Brebner, Stephanie Riche, Martin Griss and Bernard Burg. Thanks also to Martin Sadler and Andy Seaborne for helpful comments on earlier drafts of the paper.

References

1. Pretschner, A and Gauch, S. *Personalization on the Web*. (ITTC-FY2000-TR-13591-01) Information and Telecommunication Technology Center, University of Kansas. 1999. Available from: http://homer.ittc.ukans.edu/website/publications/papers/pers-tr.pdf
2. Fink J. & Kobsa A. "A Review and Analysis of Commercial User Modeling Servers for Personalization on the World Wide Web". *User Modeling and User Adapted Interaction*. Vol. 11. Kluwer, 2001. pp. 209–249.
3. Crabtree I. B., Soltysiak S. J. & Thint M. P. "Adaptive Personal Agents". *Personal Technologies*. Vol. 2:3. 1998. pp. 141–151. Available from: http://www.bt.com/bttj/vol16no3/13.htm
4. Franklin, S. & Graesser, A. "Is It an Agent or Just a Program? A Taxonomy for Autonomous Agents". In: Muller, J. P., Wooldridge, M., & Jennings, N. (eds) *Intelligent Agents III Proceedings of Agent Theories, Architectures and Languages (ATAL 96).* Springer, 1996. pp. 21–36.
5. Dickinson, Ian. *The interface as agent: a comparative review of human-agent interaction*. HP Labs Technical Report. In preparation.
6. User Modelling Inc. *User Modelling Inc Home Page*. 2001. Web site: http://www.um.org
7. Pazzani M. "A Framework for Collaborative, Content-Based and Demographic Filtering". *Artificial Intelligence Review*. Vol. 13:5 - 6. 1999. pp. 393–408.
8. Webb G., Pazzani M. & Billsus D. "Machine Learning for User Modeling". *User Modeling and User Adapted Interaction*. Vol. 11. Kluwer, 2001. pp. 19–29.
9. Kobsa A. "Generic User Modeling Systems". *User Modeling and User Adapted Interaction*. Vol. 11. 2001. pp. 49–63.
10. Riché, Stephanie, Brebner, Gavin, and Gittler, Mickey. *Client-side profile storage: a means to put the user in control*. (in preparation) HP Laboratories. 2001.

11. Kindberg, Tim and Barton, John. *A web-based nomadic computing experience*. (HPL-2000-110) HP Laboratories. 2000. Available from: http://www.hpl.hp.com/techreports/2000/HPL-2000-110.html
12. Chen, Harry and Tolia, Sovrin. *Steps towards creating a context-aware software agent system*. (HPL-2001-231) Hewlett-Packard Laboratories. 2001. Available from: http://www.hpl.hp.com/techreports/2001/HPL-2001-231.html
13. Butler, Mark. DELI: *a delivery context library for CC/PP and UAProf* . (HPL-2001-260) HP Laboratories. 2001. Available from: http://www.hpl.hp.com/techreports/2001/HPL-2001-260.html
14. Butler, Mark. *Current technologies for device independence*. (HPL-2001-93) HP Laboratories. 2001. Available from: http://www.hpl.hp.com/techreports/2001/HPL-2001-83.html
15. Barton, John and Kindberg, Tim. *The Cooltown user experience*. (HPL-2001-22) HP Laboratories. 2001. Available from: http://www.hpl.hp.com/techreports/2001/HPL-2001-22.html
16. Pohl W. "Logic-Based Representation and Reasoning for User Modelling Shell Systems". *User Modeling and User Adapted Interaction*. Vol. 9:3. Kluwer, 1999. pp. 217–282.
17. *Net Perceptions*. 2001. Web site: http://www.netperceptions.com
18. World Wide Web Consortium (W3C). *Platform for Privacy Preferences Project (P3P)*. 2001. Web site: http://www.w3.org/P3P/
19. *HP Web Services*. 2001. Web site: http://www.e-speak.hp.com
20. *IBM Web Services*. 2001. Web site: http://www.ibm.com/software/solutions/webservices/
21. *Microsoft Hailstorm*. 2001. Web site: http://www.microsoft.com/presspass/features/2001/mar01/03-19hailstorm.asp
22. World Wide Web Consortium. *XML Protocol*. 2001. Web site: http://www.w3.org/2000/xp
23. *UDDI – Universal Description Discover and Integration*. 2000. Web site: http://www.uddi.org
24. W3C. *W3C Semantic Web Activity*. 2001. Web site: http://www.w3.org/2001/sw/
25. Berners-Lee, Tim, Hendler, James, and Lassila, Ora "The Semantic Web". *Scientific American.* 2001.
26. World Wide Web Consortium (W3C). *The Resource Description Framework (RDF).* 1999. Web site: http://www.w3.org/TR/REC-rdf-syntax/
27. Berners-Lee, Tim, Fielding, and Masinter, L. *Uniform Resource Identifiers (URI): Generic Syntax*. (RFC2396) Internet Draft Standard. 1998.
28. Chandrasekaran B., Josephson J. R. & Benjamins V. R. "What Are Ontologies, and Why Do We Need Them?". *IEEE Intelligent Systems*. Vol. 14:1. 1999. pp. 20–26.
29. van Harmelen, Frank, Patel-Schneider, Peter, and Horrocks, Ian. *A Model-Theoretic Semantics for DAML+OIL* (March 2001) Revision 4.1. 2001. Available from: http://www.daml.org/2001/03/model-theoretic-semantics.html
30. Internet Mail Consortium. *vCard White Paper*. 1997. Web site: http://www.imc.org/pdi/vcardwhite.html
31. Good, N., Schafer, J. B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. "Combining Collaborative Filtering With Personal Agents for Better Recommendations". In: *Proc. 16th National Conference on AI (AAAI-99)*. AAAI Press, 1999. pp. 439–446.
32. Brown, Barry and Sellen, Abigail. *Exploring users' experiences of the web*. (HPL-2001-262) HP Labs Technical Report. 2001. Available from: http://www.hpl.hp.com/techreports/2001/HPL-2001-262.html
33. Wiederhold G. "Mediators in the Architecture of Future Information Systems". *Computer*. Vol. 25:3. 1992. pp. 38–49.

34. Cavoukian A, Tapscott D. *Who knows: safeguarding your privacy in a networked world* McGraw-Hill, 1996.
35. Cavoukian, Ann. *Privacy: The Key to Electronic Commerce*. The Information and Privacy Commissioner/Ontario . Available from: http://www.ipc.on.ca/english/pubpres/papers/e-comm.htm
36. Cavoukian, Ann. *Data Mining: Staking a Claim on Your Privacy*. Information and Privacy Commisioner/Ontario. 1998. Available from: http://www.ipc.on.ca/english/pubpres/papers/datamine.htm
37. *Privacy as a Fundamental Human Right vs. an Economic Right: an Attempt at Conciliation* . Information and Privacy Commissioner/Ontario. 1999. Available from: http://www.ipc.on.ca/english/pubpres/papers/pr-right.htm
38. Cavoukian, Ann and Gurski, Mike "Managing Privacy: a Challenge in Designing Today's Systems". *Municipal Interface*. 2000. Available from: http://www.ipc.on.ca/english/pubpres/ext-pub/misa-00.pdf
39. Wright, Tom. *Privacy Protection Makes Good Business Sense*. 1994. Available from: http://www.ipc.on.ca/english/pubpres/papers/busi-e.htm
40. TRUSTe. *Building a framework for global trust*. 2001. Web site: http://www.truste.org
41. Bowbrick, S. "Multiple Identities in the Online World". In: *Proc. First Digital Identity Workshop*. hyperion.co.uk, 2000. Available from: http://www.consult.hyperion.co.uk/PDFlibrary/y2000/bowbrick.pdf
42. *Microsoft .NET Passport*. 2001 . Web site: http://www.passport.com
43. *The Liberty Alliance Project*. 2001. Web site: http://www.projectliberty.org
44. World Wide Web Consortium (W3C). *The Platform for Privacy Preferences Project (P3P) 1.0 Specificatio*n. 2001. Web site: http://www.w3.org/TR/p3p
45. Banks, David and Reynolds, David. *Method and System for Controlling the On-Line Supply of Digital Products or the Access to On-Line Services*. Patent application (UK 0103984.1). Filed 17-Feb.-2001.
46. Brands S. *Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy* ISBN 0-262-02491-8. MIT Press, 2000.
47. Vora, Poorvi , Vazirani, Umesh, and Knapp, Verna. *Probabilistic Privacy Protection*. Patent applied for. (USA 10004309-1). Filed Dec.-2000.
48. Google.com. *Google search engine*. 2001. Web site: http://www.google.com
49. ODP - *The Open Directory Project*. Web site: http://www.dmoz.org
50. Lieberman, H. "Letizia: an Agent That Assists Web Browsing". In: *Proc. Internation Joint Conference on Artificial Intelligence (IJCAI '95)*. Morgan Kaufmann, 1995. pp. 924–929.
51. Joachims, T., Freitag, D., & Mitchell, T. "WebWatcher: a Tour Guide for the World Wide Web". In: *Proc. 15th International Joint Conference on AI (IJCAI 97*). Morgan-Kaufmann, 1997. pp. 770–775.
52. Mladenic D. "Text-Learning and Related Intelligent Agents: a Survey". *IEEE Intelligent Systems*. Vol. 14:4. IEEE, 1999. pp. 44–54.
53. Chen, L. & Sycara, K. "WebMate - a Personal Agent for Browsing and Searching". In: *Proc. 2nd International Conf on Autonomous Agents*. ACM, 1998. pp. 132–139.
54. Maes P. "Agents That Reduce Work and Information Overload". *Communications of the ACM*. Vol. 37:7. ACM, 1994. pp. 31–40.
55. Anderson, C., Domingos, P., & Weld, D. "Web Site Personalizers for Mobile Devices". In: Anand, S. S. & Mobasher, B. (Co-chairs) Proc. *IJCAI 2001 Workshop on Intelligent Techniques for Web Personalization*. IJCAI Inc, 2001. pp. 47–52.
56. Pretschner, A. & Gauch, S. "Ontology Based Personalized Search". In*: Proceedings 11th International Conference on Tools with Artificial Intelligence. TAI 99*. IEEE Computer Society. pp. 391-398.

57. NEC Research Institute. *ResearchIndex - the NECI Scientific Literature Digital Library.* 2001. Web site: http://www.researchindex.com
58. Aggarwal, C. C. & Yu, P. S. "On Text Mining Techniques for Personalization ". In*: Proc. 7th Internaltional Workshop on New Directions in Rough Sets, Data Mining and Granular Soft Computing RSDMGrC'99*. Springer-Verlag, 1999. pp. 12–18.
59. Cayzer, Steve and Aickelin, Uwe. *A Recommender System based on the Immune Network.* (in preparation) 2001.
60. Dasgupta D. *Artificial Immune Systems and Their Applications* Springer-Verlag, 1998.
61. Compaq Systems Research Centre. *EachMovie Collaborative Filtering Data Set*. Web site: http://www.research.compaq.com/SRC/eachmovie/
62. Delgado J. & Ishii N. "Multi-Agent Learning in Recommender Systems For Information Filtering on the Internet ". *International Journal of Cooperative Information Systems*. Vol. 10:1-2. 2001. pp. 81–100.
63. Gnutella. Web site: http://gnutella.wego.com
64. Adamic, Lada and Adar, Eytan. "Friends and Neighbors on the Web " *unpublished manuscript.* 2000 Available from: http://www.hpl.hp.com/shl/people/eytan/fandn.html
65. Soltysiak S. J. & Crabtree I. B. "Automatic Learning of User Profiles - Towards the Personalisation of Agent Services". *BT Technology Journal*. Vol. 16:3. 1998. pp. 110–117. Available from: http://www.bt.com/bttj/vol16no3/13.htm
66. Soltysiak, S. J. & Crabtree, B. "Knowing Me, Knowing You: Practical Issues in the Personalisation of Agent Technology". In: *Proc. Third International Conf. on the Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM98).* 1998. pp. 467–484.
67. Salton T. & Buckley C. "Term-Weighting Approaches in Automated Text Retrieval". *Information Processing & Management*. Vol. 24:5. 1988. pp. 513–23.
68. Chaffee, J. & Gauch, S. "Personal Ontologies for Web Navigation". In*: Proc. 9th International Conf. on Information and Knowledge Management (CIKM'00).* 2000. pp. 227–234. Available from: http://homer.ittc.ukans.edu/website/publications/papers/chaffeeCIKM2000.pdf
69. Orwant J. "Heterogenous Learning in the Dopplegänger User Modeling System". *User modeling and user adapted interaction*. Vol. 4:2. 1994-1995. pp. 107–130.
70. Paiva A. & Self J. "TAGUS - a User and Learner Modeling Workbench". *User Modeling and User Adapted Interaction*. Vol. 4:3. 1995. pp. 197–226.
71. Kay, J. "Vive La Difference! Individualised Interaction With Users". In*: Proc. International Joint Conference on AI (IJCAI)* 1995. Morgan-Kaufmann, 1995. pp. 978-84.
72. *The DARPA Agent Markup Language Homepage*. 2001. Web site: http://www.daml.org
73. *CPExchange home page.* 2000. Web site: http://www.cpexchange.org
74. Weal, Mark, Hughes, Gareth, Millard, David, and Moreau, Luc. *Open Hypermedia as a Navigational Interface to Ontological Information Spaces*. (ECSTR-IAM01-004) University of Southampton. 2001. Available from: http://www.bib.ecs.soton.ac.uk/records/5161
75. Erickson, J. , Williamson, M., Reynolds, D., Vora, P., & Rodgers, P. "Principles for Standardization and Interoperability in Web-Based Digital Rights Management*". In: Proc. W3C Workshop on Digital Rights Management (DRM).* Jan 2001: Available from: http://www.w3.org/2000/12/drm-ws/pp/hp-erickson.html
76. Vora, P., Reynolds, D., Dickinson, I., Erickson, J., & Banks, D. "Privacy and Digital Rights Management". In: *Proc. W3C Workshop on Digital Rights Management (DRM).* 2001. Available from: http://www.w3.org/2000/12/drm-ws/pp/hp-poorvi2.html

77. Zero Knowledge Systems. *Enterprise Privacy Manager.* 2001. Web site: http://www.zeroknowledge.com/business/privacyrights.asp
78. World Wide Web Consortium (W3C*). A P3P Prefenence Exchange Language (APPEL) 1.0 Working Draft.* 2001. Web site: http://www.w3.org/TR/P3P-preferences.html
79. Duncan G. T. & Mukherjee S. "Optimal Disclosure Limitation Strategy in Statistical Databases: Deterring Tracker Attacks Through Additive Noise". *Journal of the American Statistical Association*. Vol. 95:451. 1998. pp. 720–729. Available from: http://duncan.heinz.cmu.edu/GeorgeWeb/OptimalDisclosure Limitation 98 June 16.htm
80. Beck L. L. "A Security Mechanism for Statistical Databases". *ACM Transactions On Database Systems*. Vol. 5:3. 1980. pp. 316–318.
81. Schlorer J. "Security of Statistical Databases: Multidimensional Transformation". *ACM Transactions On Database Systems.* Vol. 6:1. 1981. pp. 95–112. Available from: http://www.acm.org/pubs/articles/journals/tods/1981-6-1/p95-schlorer/p95-schlorer.pdf
82. Adam N. R. & Wortmann J. C. "Security-Control Methods for Statistical Databases: a Comparative Study". *ACM Computing Surveys*. Vol. 21:4. 1989. pp. 515–556.
83. Agrawal, R. & Srikant, R. "Privacy-Preserving Data Mining". In: *Proc. of the 2000 ACM SIGMOD Conference on Management of Data*. ACM, 2000. pp. 439–450.
84. Microsoft. *Introducing .NET My Services*. 2001. Web site: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/Dndotnet/html/ Myservintro.asp?frame=true
85. Rosheisen, R. M. *A Network-Centric Design for Relationship-Based Rights Management (PhD Thesis).* Stanford University. 1997.
86. *Jakarta Lucene*. 2001. Web site: http://jakarta.apache.org/lucene
87. *The Apache Jakarta Project*. 2001. Web site: http://jakarta.apache.org