

Supporting Privacy in E-learning with Semantic Streams

Lori Kettel, Christopher Brooks, Jim Greer

Abstract— The goal of the semantic web is to facilitate the exchange of meaningful information in a form that is easy for machines to process. The goal of an e-learning system is to support the learning process by providing meaningful information for students in a form that best suits them. Natural synergies between these two areas have led to a surge in the feasibility of user modelling and intelligent tutoring systems, where computer systems can understand a learner, and help guide them through the learning process. The question arises then; with deep meaningful information being shared about learners, who is protecting the privacy rights and desires of the learner? This paper provides an investigation of this question, and outlines approaches that are being undertaken to integrate privacy into our current semantic web e-learning framework.

Index Terms—Semantic Web, Privacy, User Modelling

I. INTRODUCTION

IN an interactive e-learning environment, learners, educators, researchers, and administrators need to know about one another. Individuals often have the ability to act under a number of different roles (e.g. student, marker, tutorial assistant, and part time researcher), making the environment complex. As more features and functions are added to the environment, the amount of user information, the detail of that information, and the number of parties interested in that information increases. This then begs the question -- in a complex, multi-user, multi-role environment, how do we impart personal information to others while protecting privacy?

Information about learners is diverse and may include a wide range of activities and assessments such as quiz results, assignment marks, submission times, how often and when the

learner accessed course materials, postings made and read on web-based discussion boards, ratings of posting quality by participants, chat interaction logs, and opinions held about the learner by others. Even though most e-learning environments require that learners implicitly relinquish control of some of this information (often for both system functionality as well as pedagogical evaluation purposes), there are privacy concerns when some of this information begins to be shared with others. The rise in popularity of Computer Supported Collaborative Learning (CSCL) environments as well as increased concerns over privacy legislation are fuelling a push for next generation solutions for privacy management. CSCL environments typically encourage learners to share personal information with one another to encourage group building, self-reflection, social navigation and finding an appropriate helper for students having problems. [3] By allowing learners to see one another's activities online, awareness is promoted within the learning community. This awareness can serve as a motivator for learners, making them feel less isolated and more a part of the learning community, and encouraging them to participate actively within groups helping to fuel the learning process.

In addition to learners and teachers, the expected set of participants, e-learning environments often include a diverse set of institutional stakeholders. Interested institutions may include school systems, government agencies, non-profit corporations, private corporations, professional organizations, and special interest groups. The range of interests that these groups bring to the teaching and learning process also make this an ideal domain for studying privacy technologies.

A significant problem to supporting privacy in e-learning is in determining how much information the environment should allow others to see. Evidence suggests that certain people will feel more comfortable releasing greater amounts of information about their online activities than others. Releasing personal information is strongly linked to the trust individuals put into the roles that are adopted by others around them. For instance, learners are more likely to release private information to certain groups (instructors, tutorial assistants, and friends) than to others (classmates and strangers). As a general rule, each person must feel as though his or her desired level of privacy is respected and will not be violated. [11]

Educational technology researchers typically take information gathered about individuals in an e-learning environment and draw it together into a representation known as a user model. User models can then be used to provide for

Manuscript received August 20, 2004. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada through the LORNET Network Centres of Excellence project.

L. Kettel is with the Laboratory for Advanced Research in Intelligent Educational Systems (ARIES) in the Department of Computer Science at the University of Saskatchewan, Saskatoon, SK, Canada (306-966-4740; e-mail: kettel@cs.usask.ca).

C. Brooks is with the Laboratory for Advanced Research in Intelligent Educational Systems (ARIES) in the Department of Computer Science at the University of Saskatchewan, Saskatoon, SK, Canada (e-mail: cab938@mail.usask.ca).

J. Greer is with the Laboratory for Advanced Research in Intelligent Educational Systems (ARIES) in the Department of Computer Science at the University of Saskatchewan, Saskatoon, SK, Canada (e-mail: greer@cs.usask.ca).

adaptive instruction, tailoring of user interfaces, locating expert learners for peer help, and a number of other functions. While there are no widely used learner modelling standards, most current implementations partition information into three categories:

- Private information, to be shared only with those that the learner indicates
- Public information, to be shared freely with most other users in the system
- Hidden information, to be kept by the system to provide functions to the user, but not shared with the user explicitly

Privacy and access rights of certain groups or individuals, individual privacy preferences, and the system's over-riding need for good quality information about users are all considerations that determine the kinds of information that go into a user model.

The semantic web is becoming a key technology for building modern adaptive e-learning environments. The semantic web is often referred to as the web of meaning where software components (often called agents, for their ability to act autonomously on behalf of a user) can share information to accomplish a variety of tasks that would be difficult or impossible to do on the traditional web. Fundamental to this view is the need to easily represent and share information about users, objects, usage and context. Because such environments are dynamic, one can consider constant streams of new information being created. New content, new inferences, new user actions all constitute events to be supported by the semantic web.

This paper provides a look into how we are implementing privacy control in our next generation e-learning environment. Section II outlines the core framework of the environment, built around semantic web and web service technologies. Section III describes both anecdotal and empirical motivations for supporting privacy in e-learning systems. Section IV presents some of our preliminary work on privacy servers and section V outlines techniques being used to support privacy in our new e-learning infrastructure. Finally, section VI concludes the paper with a look at the contributions and limitations of this approach and avenues for future research.

II. AN ARCHITECTURE FOR SEMANTIC STREAMING

To support the capture and dissemination of user modelling information on the semantic web, we have built and deployed a system dubbed the Massive User Modelling System (MUMS) [2]. This system acts as a piece of middleware between those software components in an e-learning system that produce information about users (e.g. a learning management system quiz tool), and those that need to consume that information (e.g. an intelligent tutoring application). MUMS is aimed specifically at providing access to user modelling information in real-time, using a publish/subscribe interaction model. The middleware identifies four primary software entities in an e-learning system, evidence producers, user modelers, the

broker, and various filters. Each of these will be discussed in turn.

Evidence producers observe user interaction with an application and publish information about the user. This information can range from direct observations of the user, to models about the users mental state. Evidence producers can range from very simple software components to more complex agents, who can understand the learner at a deeper level. For example, a testing application may generate just simple events about a learner indicating which questions they got correct or incorrect, while a web browser may generate events about which websites a user visited, how long they visit, what actions they took while they were at that website, and perhaps even an idea of why the user went to that website.

User modellers are interested in acting on information about a user, usually by reasoning over events to create a user model. The modeller then interacts with the user (or the other aspects of the system, such as learning materials) to provide adaptation or instruction. Modellers may be interested in modelling more than one user, and may receive user information from any number of evidence producers. For example, an intelligent tutoring system may register to receive user information events from both web browsers and testing tools. The tutor could then make correlations between websites that were effective for certain tests, and then recommend these websites to other users in the system.

The broker acts as a logically centralized router between evidence producers and modellers. The broker matches the information coming from evidence producers to the modellers that are interested in that information. Matching is done based on the semantics of the information being passed as opposed to the evidence producer that the information is coming from. This supports the ad hoc addition and deletion of both evidence producers and modellers to the system.

Finally, filters are components which can masquerade as broker, modeller, and an evidence producer. By registering for and reasoning over user information, a filter can create higher level annotations about what has happened. This offloads the amount of work done by a modeller to form a user model, but maintains the more flexible decentralized environment. Filters can be chained together to provide any amount of value-added reasoning desired. Finally, filters can be specialized within a particular instance of the MUMS framework by providing domain specific rules that govern the registration, processing, and creation of user modelling information.

The format of information being passed between these entities is known as an *opinion*. Opinions are packages of statements about a user in the form of triples (subject, predicate, object) corresponding to the Resource Description Framework (RDF) data model [5]. This data model provides flexibility in that opinions can be marked up in any number of different ontologies, with as much or as little detail that is required. This is important, as a number of prominent e-learning standards and specifications, such as the IEEE Learning Object Metadata standard [12] and the IEEE PAPI

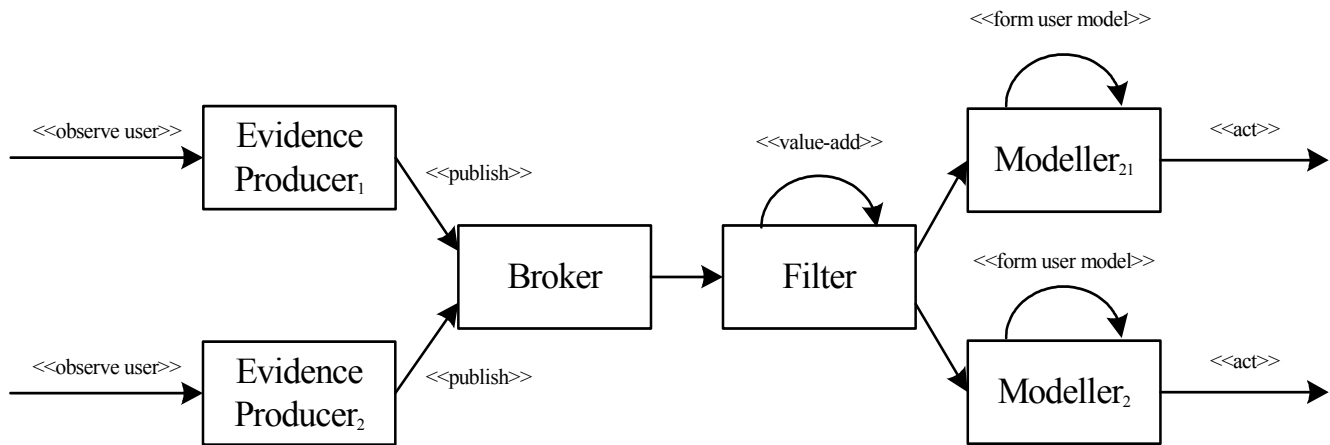


Fig. 1. Flow of information through the MUMS system. Evidence producers create opinions which they publish to a broker for distribution to filters and modellers. Filters take opinions, modify them using domain knowledge, and provide them to modellers who then create user models and interact with the learner.

[8] have bindings to RDF, and can be transparently supported by the framework. Fig. 1 provides a flow diagram describing how entities in the system interact with one another.

Opinions are transferred from evidence producers to the broker, where they are stored and indexed for quick retrieval. Modellers register with the broker for a specific set of opinions by providing an *event query*. This event query identifies the kind of information a modeller is interested in receiving. Brokers then forward only those opinions that match a given modellers' query to those modellers – there is no channel or producer based routing.

For example, a software component interested in providing visualizations of peer networks could register with a broker to receive all events involving email, instant messaging, and real-time chatting. Assuming that all mail clients, instant messaging applications, and chat engine implementations can agree on a shared ontology, the visualization client can register for usage data and remain ignorant of client specific implementation details. Further, as new evidence producers come online no negotiation process is required to start receiving new opinions – those that match the registered event query will automatically be forwarded to the visualization component.

The MUMS system enables us to build simple hooks into e-learning applications and to gather models of users. The semantic streams of events that MUMS produces and filters can be used for purposes of internal system reasoning, assessment of learners, teachers or other actors, and for data gathering and mining for many other purposes. The semantic streams also require regulation as the information that flows may be private or sensitive and individuals may wish certain levels of privacy protection of their information. This would include what information is released, to whom it is released, and for what purpose.

III. MOTIVATIONS FOR PRIVACY IN E-LEARNING

Users of an e-learning environment usually have differing

views on how much information they are willing to share with others. To determine the willingness of learners in sharing their online activities, we conducted several surveys of users of a web-based discussion forum called I-Help Public Discussions [3]. Participants were asked about whom they would be willing to reveal various pieces of sensitive information to, and what information they would be interested in knowing about others. One of the surveys was divided into two parts. In the first part, users were asked about their willingness to reveal information in four different categories:

- General information: willingness to help, areas of interest, descriptive personal information
- Value judgments: grades obtained, endorsements by fellow learners, achievement scores
- Activity records: times when they log on or when they post or read questions
- Profile attribute modification: such as setting knowledge levels or changing areas of interest

For each of the four categories, respondents were asked to whom they would be willing to release certain information (everyone, classmates, friends, no one) and with what identifying label (real name, alias, or as part of a global summary). In the second part of this survey, users were asked how much they wanted to know about other people using the same four categories.

One hundred and thirty-three people responded to this survey. The results showed that people were more willing to reveal information if using an alias rather than if they were to use their own name. Fig. 2 shows the reluctance of people to disclose personal information based on how they would be identified. For all types of information, users are more reluctant to release the information that identifies themselves by name than that which includes them as an anonymous part of a summary.

The survey showed that people are almost as unwilling to disclose their online activity records, as they are reluctant to disclose grades earned in a class. In addition, they were

considerably more reluctant to disclose details of their

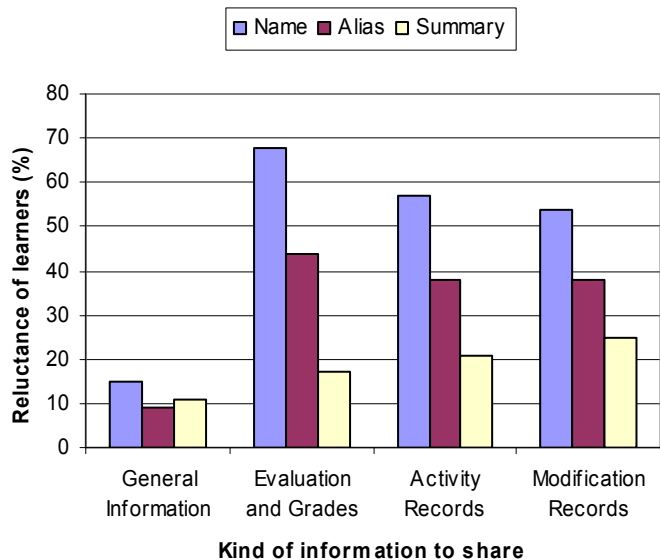


Fig. 2. Percentage of learners reluctant to share personal information. The trend in most cases indicates learners will be more reluctant to share information if it becomes easier to identify them by it.

activities than general information about themselves such as willingness to help or areas of interest. This indicates that people have concerns about revealing their detailed set of actions while online, as well as concerns about what inferences others could potentially make about them because of these actions. Whatever the reasons for the unwillingness, it is necessary to create a mechanism that allows people to control who has access to their online activity, allowing those the user trusts to view the information while hiding it from others.

Fig. 3 describes to whom users would be willing to release their information and whether they would prefer to release it by name, alias or anonymously as part of a summary. As shown previously, learners are more willing to release general information than grades or activity records. More importantly, they are more willing to release information under an alias than under their real name and even more willing to release information if it is part of a summary that does not reveal any identifying information. For this reason, online learning environments should generally make use of aliases to protect the privacy of the users. In addition, learners were less interested in pseudonymising or anonymising data when the information was destined for others in their own personal roles (such as their friends).

Based on this and other surveys conducted by us as well as others (e.g. [1] [6]), there is definitely a concern about privacy in online communities of learners. Given sufficient reasons and enough confidence in the system, users are willing to disclose information about themselves but would often prefer to do so in a pseudonymous or anonymous fashion when possible.

IV. TECHNIQUES FOR SUPPORTING PRIVACY IN E-LEARNING

In order to allow users to control their own level of privacy there needs to be a system in place to control information streams for different groups and individuals. The Privacy Event Stream Traffic (PEST) Server is the name given to the application agent that we designed to control access to the stream of events and information that flow through learning environments [11]. Systems like I-Help or other learning environments, learning management systems, or collaboration tools can be configured to share their information with PEST. Every time users interact with the learning environment, an event is triggered within the system and passes through the system's Event Stream. PEST was designed to work in a multi-agent software environment with a system of personal agents that represent human users. It is assumed that each user has their own personal agent and that users configure or program their agent with their own individual privacy preferences. The user describes which kinds of information can be passed on to different types of users (e.g. grades to my teachers and interest indicators to my friends), and in which format (e.g. identified by name, alias, or anonymously). When someone wishes to know the actions of another person in the system, their personal agent attempts to view the event stream to retrieve the information. To protect each person's privacy, each personal agent tells PEST who should be allowed access to view its owner's actions and which actions may be viewed by which people. PEST then creates a restricted personal event stream for each agent to access, which contains only those events others permit them to see.

For our initial prototype implementation of PEST in the context of the I-Help system, seven categories of events were considered:

- Logging in/out of the system
- Changing preferences
- Viewing postings or content pages
- Responding to a help request/posting messages
- Requesting information
- Updating System attributes (providing evaluations, prestige ratings, etc.)
- Updating privacy settings/viewing others' information

For each of the seven categories, each user could advise his/her personal agent the level of privacy to be assigned. The privacy levels are:

- Visible to all system groups (courses) to which the user belongs identifying the user by name.
- Visible to all system groups (courses) to which the user belongs identifying the user by "alias" or "pseudonym".
- Visible to all system groups (courses) to which the user belongs as part of a summary containing no identifying information.
- Visible to members of selected groups the user has created (often friends or study groups), identifying the user by name.
- Visible to members of selected groups the user has created (often friends or study groups), identifying the

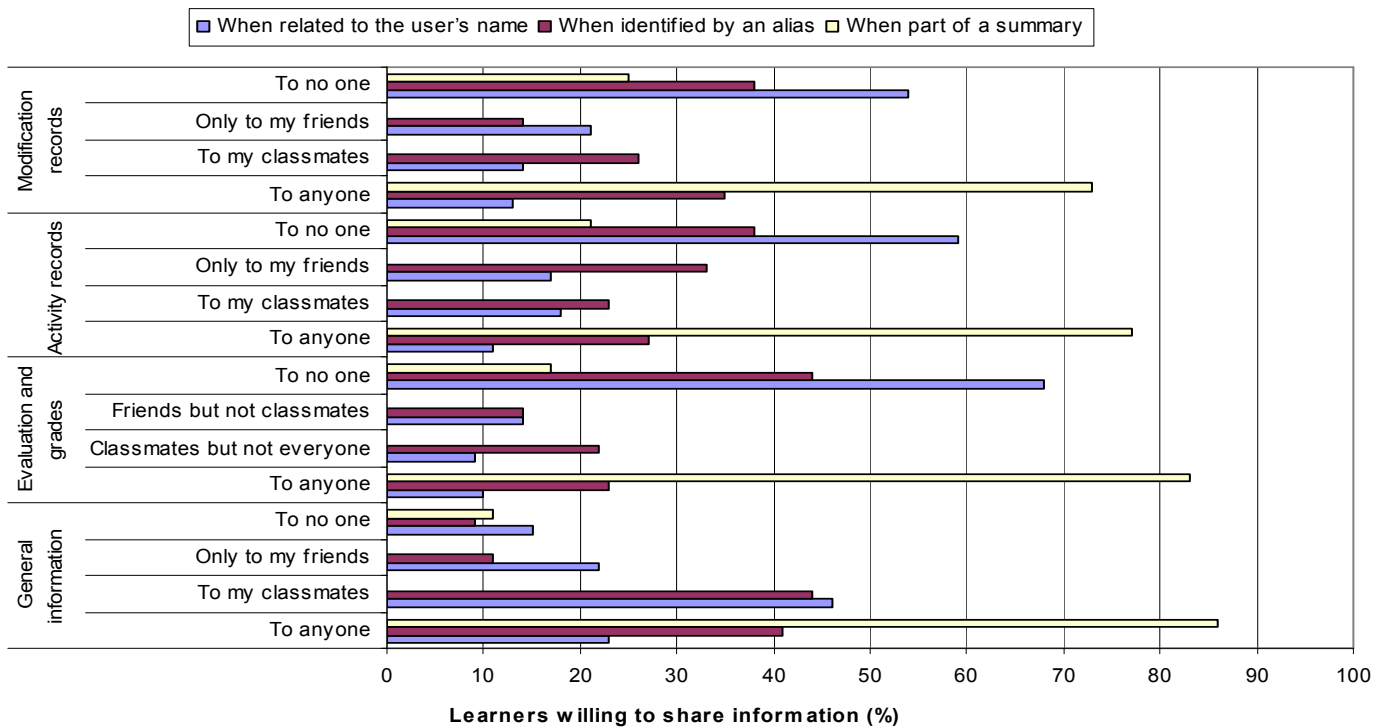


Fig. 3. Percentage of learners willing to share information to different roles within an e-learning system. Data suggests that that the more personal the role, the more likely a learner is to share information without pseudonymising or anonymising it first.

user by “alias” or “pseudonym”

- Visible to members of selected groups the user has created (often friends or study groups) as part of a summary containing no identifying information.
- Visible only to particular individuals specified by the user, identifying the user by name.
- Visible only to particular individuals specified by the user, identifying the user by “alias” or “pseudonym”.
- Visible to no one

Each category of events begins with default levels of privacy including:

- Information about the user’s login times will be available to all people in the same course as the user (identifying the user by alias).
- Information about viewing pages and requesting information will be available to members of groups the user creates (identifying the user by alias).
- No other interactions are visible by anyone.

The user can change the default settings at anytime. When a person wishes to know the activities of another person, the personal agent checks the event stream created by PEST for that agent. PEST can see all events, but PEST makes available to others’ personal agents only those events that the individual has explicitly allowed. Users are able to set a permission level for each type of event that occurs. The permission level represents the degree restrictions that are placed on the release of the events and are as follows:

- events are not released

- events are released as part of a summary only
- events are released by alias
- events are released by name.

While there are pre-defined groups (classmates, teachers, students, tutors, etc.), users can also create their own groups (e.g. friends, project team members, banned people, etc.) Users can set more restrictive or less restrictive permission levels for each group. Therefore, users can release more information to friends than classmates or can create their own list of banned users who are not permitted to see any information about them.

A study using PEST was designed that allowed students to set their privacy preferences. The study was intended to see what effect allowing students to hide as much or as little about themselves would have on other students’ opinions of them. It also gave students the opportunity to reset their privacy preferences after using these inspection tools, after having a chance to live with the consequences of their and others’ privacy settings. Rather than having the study participants use the I-Help system directly, the participants were given access to the profiles of simulated students within the system and then given a set of scenarios that required the participants to view the activities of the simulated students. The simulated students revealed differing amounts of information in differing formats, therefore allowing the participants to see how the system would reveal or hide information as requested. Using simulated students and situations for the study had two benefits. The first benefit was that there were no risks to privacy invasion of the participants, as any information

revealed did not belong to a real student. Second, running the simulation required participants to use the system for only an hour or two rather than needing to use it in a real class setting for weeks or months.

Two groups of undergraduate students were selected to determine if they would find it useful to be given the opportunity to view model information about other users in an online learning environment. One group consisted of students taking computer science courses and the other group was non-science majors.

Sample inspection/awareness tools were developed to allow the participants to view information of other student's use of the I-Help system. These inspection/awareness tools allowed participants to view general information and usage patterns of a student based on whether the students allowed information to be released by name, alias, as a summary or not at all. Seven types of information were released: name and identifying information, notifications, browsing pattern, searches conducted, assignment submission dates, frequency of communication and frequency of access. The inspection/awareness tools processed the data received through the Privacy Server and displayed the information to the users in a human readable form. The participants were then asked to make assessments of the simulated students in the system based on these inspection tools.

When the participants were asked whether they felt it is beneficial to see other people's activities, 53% said yes and only 12.5% said no. The main reason students felt the inspection/awareness tools were beneficial was that the tools allowed them to see that other students are having similar problems and hence would feel more comfortable discussing problems with other students than with the professor. When asked if they felt the awareness tools were an invasion of privacy, most of the participants said there was a risk of an invasion of their privacy, only 12.5% felt there would be no risk of privacy invasion. The last question asked was whether participants felt the potential benefits from using such a system were greater than the possible risk to their privacy. Overwhelmingly, people did feel it was worth the risk. 87.5% felt the benefits were greater than the risk to privacy and 50% felt that the benefits were very significant.

PEST was successful in demonstrating that by allowing learners fine-grained control over their information they were more comfortable in sharing information with other learners. This sharing of information allows learners to become aware of the other learners in an educational environment, fostering a greater sense of community. This makes the environment more useful to the learners and therefore can encourage learners to become more active in the environment. Unfortunately PEST was limited to the I-Help system and required a full ontological knowledge of the domain. In order to be most useful it would need to be developed into a framework that allows access by many different applications that provide their own knowledge. PEST was also limited in scope in assuming that agents would be programmed by users

to not only protect the learner's privacy but also consume, process, and draw conclusions from personal information gathered about others. The job of consumer and protector should be separated into two or more entities. The consumers of the information could then be not only other learners but also instructors, intelligent agents, or outside entities such as professional licensing bodies or public or private interest groups. Information obtained by instructors for assessment purposes may necessarily be more revealing than users might prefer, while outside groups may be even further restricted than by the privacy server than some learners might allow. Also, it would be beneficial if the server could reason about the information being released to ensure it cannot be combined to reveal more about the learner than desired.

V. EXPANDING PRIVACY CONTROL OVER SEMANTIC STREAMS WITH MUMS FILTERS

While PEST was able to give control of personal information to the user, it only worked within the confines of the I-Help system. PEST needed to know not only what the events were that were happening within the system but also need to know how events related to each other and the meaning behind the events. This is difficult in larger heterogeneous environments, where expansion of end user applications happens frequently. In these environments it is often undesirable to integrate privacy controls directly into the e-learning applications themselves, as a number of daunting consistency issues arise – consistent look and feel of privacy controls, consistent use of language to ensure semantic clarity for the learner, and a consistent inter-application protocol to describe and share privacy rules. In addition, applications would need to have their privacy rules rapidly updated as designers share ontological information, forcing an increase in revision cycles. Instead, it is useful to integrate privacy features directly within a user modelling framework, such as MUMS, and provide clear interfaces for e-learning applications to make use of these features.

The MUMS architecture naturally supports the pruning and censoring of data by the addition of a filter between the broker and modellers¹. Filters take opinions from a number of evidence producers, apply domain logic to these opinions (reasoning over them), and provide the derivative opinions to registered modellers. With this framework, it seems natural to integrate PEST style privacy agents into a MUMS filter, and connected it as a component in our distributed modelling framework.

While PEST agents required deep knowledge of the domain being modelled, filter agents instead rely on transformation rules to restrict (or extend!) user models. There exists one agent for each user of the system, in charge of enforcing privacy for that user. As opinions are passed to the privacy filter, it inspects the opinion using a small shared ontology to

¹ It should be noted that both the filter and the broker can be thought of as logically centralized entities, but are actually implemented in a distributed fashion to provide for higher reliability.

determine which users are involved. The opinion is then passed onwards to the appropriate filter agents.

Each filter agent contains both an ordered set of institutional rules, and an ordered set of user defined rules. Each rule is a triple $R = (G, U, T)$ where:

- G: A pattern within the opinion to match (e.g. a set of RDF nodes/arcs)
- U: The user or role for which that rule applies
- T: An ordered set of transformations to apply to opinion

While the agent's goal is to protect the privacy of a given user, it is autonomous and interacts within the role structure of the institution that it is involved with. Certain roles within an educational environment must be given full access to user modelling information. For instance, an instructor must be given access to student grades in a course with assurances that the student had not restricted or modified information within the model. Because of this, institutional rules are evaluated separately from user defined rules.

In addition to separating institutional rules from user

defined rules, both rule sets are ordered as a priority queue. This supports a more flexible method of control, as rules can be chained such that the results of one are applied to the results of another. For instance, a user may define a rule to allow their friends to see their online status, and another rule to disallow other classmates from seeing their online status. A third rule could be put into a lower priority in the rule set which extends the opinion by adding user location data, if there is an online status present. Based on which of the first rules were fired, the third rule would either extend the opinion, or do nothing.

As opinions are represented using RDF, they can be thought of as graph based data structures. Thus, a rule that results in applying a transformation set to the opinion is similar to inserting or deleting nodes within a graph. Because a given user may be in many roles, each agent must apply all of those rules which apply to those roles to the opinion. The results of applying a given rule are always passed down to the next rule in the rule set.

The result of applying a rule set to the opinion is thus two graphs – one which contains information that can not be

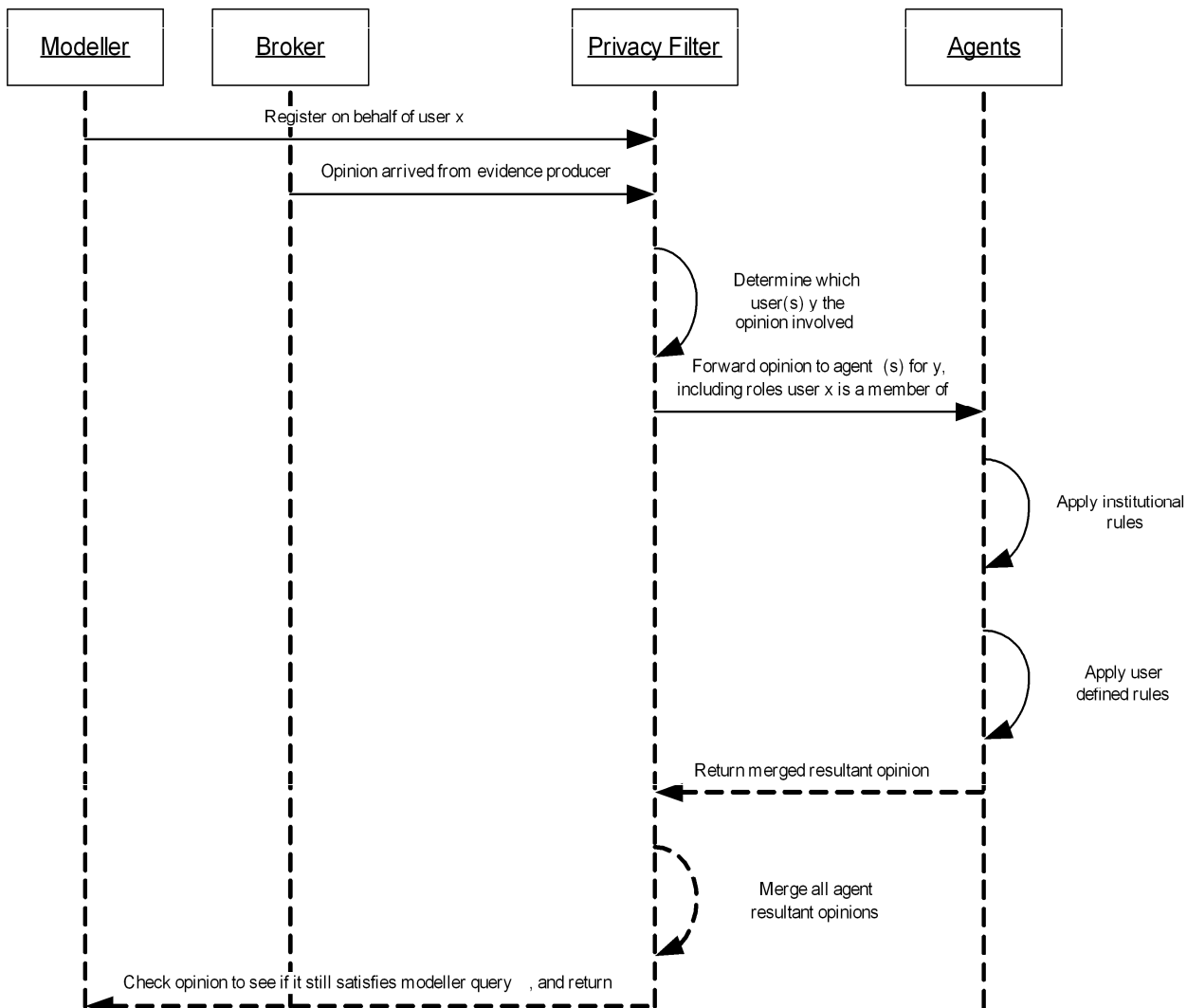


Fig. 4. Flow of information through the MUMS privacy filter.

restricted because of institutional rules, and another containing the minimum set of modelling information an agent will convey based on user defined rules. These graphs are then unioned together and create a *resultant opinion*. This opinion is then unioned with the resultant opinions of the other user agents in the filter (if these agents were given the original opinion). Finally, the results of this union are checked to see if they still satisfy the original query provided when the modeller registered, and if so, are forwarded off to the modeller for consumption.

Fig. 4 describes the flow of information through the MUMS privacy filter.

Work on the MUMS privacy filter is at a relatively early stage. Success of this approach depends heavily on the functionality of personal agents to create and manage privacy rules and to digest the results of privacy-filtered semantic streams. We need to work on an enriched view of personal agents beyond those we have previously used in the I-Help project. It is still not clear whether it is feasible for personal agents to manage and apply both institutional and user-defined privacy rules efficiently. In particular, the current applications in our e-learning environment put a heavy emphasis on roles. The greater the number of roles that exist in the system, the greater the number of rules that a privacy agent may apply. This effectively increases the computational load, and thus the number of resources needed for privacy, for each agent.

In addition to agent complexity, there still exists issues with providing an efficient and natural mechanism for learners to interact with their agents. While some work on user programming of personal agents to request and process awareness information about others has been done in a related project [4], but this is yet to be integrated into a MUMS-style application. In addition to human-computer issues, making the agents ontology aware (but not ontology specific!) is a key issue.

Finally, the agents presented in this section are extremely simple, and have only knowledge of the rules they are given, and the opinion that is to be checked. One can imagine, however, agents with a deeper knowledge of the learner and the various domains the learner is interacting with. These agents would be able to not only restrict the information in resultant opinions, but could include external deductions about the learner to round out the opinions sent to modellers.

At the same time we are encouraged that this direction involving MUMS filtering to support privacy requirements will be useful. MUMS brings the advantage of a privacy filtering infrastructure where deep knowledge of domain ontologies and tight coupling with applications can be avoided. Thus there is a generality in the solution and some hope that in many applications where semantic streams of user information are produced and consumed, a privacy filter can be easily integrated.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

This paper shows an evolution in our work on privacy

filtering – moving from regulating application-specific information as it is passed from one agent to another to creating a more general privacy filter approach for semantic streams. Our studies with users demonstrated the need for such technology in collaborative e-learning environments. One thing that we learned in these user studies was that the initial tendency of many users was to become virtually invisible to others – to opt out of sharing any personal information, and yet to wish to acquire as much personal information about others as possible. This anti-social activity would cause a community to crumble. A simple policy was introduced in which no user could access more information about anyone than they were willing to share with someone. This policy seems to restore some level of social order in the collaborative learning environment.

Our prototypes of PEST demonstrated that privacy filtering can work effectively and that users will react appropriately to the tradeoffs between privacy and awareness. But the limitations of PEST in its difficulty to scale up to other application domains seemed problematic. The move toward MUMS filters for privacy seems to be a promising step forward.

The MUMS architecture is currently deployed within a distributed e-learning environment in the Department of Computer Science at the University of Saskatchewan. This environment is made up of a number of applications designed by various research and instructional support groups over the last decade. Principle applications include:

- A content delivery system, which deploys IMS content packaging [10] formatted learning objects.
- A web based discussion forum system built around the notions of peer help (I-Help Public Discussions [7]).
- A real-time chat application (I-Help Chat) based on the Internet Relay Chat protocol.
- A quizzing system that deploys IMS QTILite [9] formatted quizzes and records evaluations of students.

REFERENCES

- [1] M.S. Ackerman, L.F. Cranor, and J. Reagle, "Privacy in E-Commerce: Examining User Scenarios and Privacy Preferences" *ACM Conference on Electronic Commerce*, pp. 1-8.
- [2] C. Brooks et al., "The Massive User Modelling System" *7th International Conference on Intelligent Tutoring Systems (ITS04)*, Springer-Verlag, 1930.
- [3] S. Bull et al., "User Modelling in I-Help: What, Why, When and How" *User Modelling 2001 (UM01)*, 2001, pp. 117-126.
- [4] Y. Cao, *A case study of agent programmability in an online learning environment*, master's thesis, Saskatoon, SK, Canada, University of Saskatchewan, 2004.
- [5] J.J. Carroll and G. Klyne, "Resource Description Framework (RDF): Concepts and Abstract Syntax", World Wide Web Consortium (W3C) recommendation, February 2004; <http://www.w3c.org/TR/rdf-concepts/>
- [6] Georgia Institute of Technology, "GVU WWW User Survey" 1998; http://www.gvu.gatech.edu/user_surveys.
- [7] Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S., and Kettel, L. Lessons Learned in Deploying a Multi-Agent

- Learning Support System: The I-Help Experience. In *Artificial Intelligence in Education 2001*. San Antonio, TX, USA.
- [8] *IEEE P1484.2. *, Personal and Performance Information (PAPI) Specification*, IEEE, Inc., 2002
 - [9] IMS Global Learning Consortium Inc. IMS Question & Test Interoperability Lite Specification, Version 1.2. 2002.
 - [10] IMS Global Learning Consortium Inc. IMS Content Packaging Specification version 1.1.3. 2003.
 - [11] L. Kettel, *Privacy and Awareness in Agent-Based Learning Environments*, master's thesis, Saskatoon, SK, Canada, University of Saskatchewan, 2003.
 - [12] IEEE P1484.12.1-2002, Draft Standard for Learning Object Metadata, IEEE, Inc., 2002