MONOLOGUE: A Tool for Negotiating Exchanges of Private Information in E-Commerce

Scott Buffett*, Luc Comeau†, Michael W. Fleming†, Bruce Spencer*

*National Research Council, IIT - E-business, Fredericton NB, E3B 9W4, {scott.buffett, bruce.spencer}@nrc.gc.ca
†University of New Brunswick, Fredericton NB, E3B 5A3, {luc.comeau, mwf}@unb.ca

***Abstract*— The MONOLOGUE system (Multi-Object Negotiator for On-Line Offers Guided by Utility Elicitation) for privacy negotiation is described. It integrates a number of innovative components. Negotiations are alternating offers, where each offer may have multiple attributes which can be scalar-, discrete- or set-valued, and so are general enough to allow P3P statements to be negotiated. It relies on the PrivacyPact protocol, which ensures that a mutually acceptable deal, if present, will be found. The utility of private information is elicited from the user by posing questions, bearing in mind the bother cost of asking. Several negotiation strategies are available. MONOLOGUE analyses the opponent's responses to classify the opponent; better knowledge of the opponent can help to find mutually acceptable agreements. Experiments with simple strategies show that negotiations can converge to an agreement quite quickly.**

I. Introduction

People are often reluctant to engage in electronic commerce interaction because they fear loss of private information. Often a website will request personal information such as name, home address, browser type, etc., from a user accessing the site. This information can be used to improve the website's content or services offered. However, occasionally a website's policy might include less desirable practices, such as the transmission of private information to third parties.

Recent work in privacy economics research has shown that concern over privacy and security is the number one reason why people do not make purchases [23]. Culnan and Armstrong [13] argue that concerns over Internet privacy have a negative influence on likelihood of electronic exchange. Contrary to popular belief, experience does not tend to breed wariness. In fact, increased Web usage has been shown to decrease concern over privacy [22]. One reason for this is that, with experience, people tend to see the benefits of giving away information, such as Web site personalization, customer profiling or lower prices. Another factor that has a positive relation with reduced concern over privacy is perceived control over one's private information [22]. Culnan and Armstrong argue that consumers are more willing to share their private information if they believe that fair information practices are in place. Fair information practices are those that 1) reveal why the information is being collected and how it will be used, and 2) give consumers control over its possible uses. Empowering users with knowledge of the advantages and disadvantages of releasing private data, and also with control over such data after its release, is thus a vital step toward overcoming fears associated with privacy and achieving growth and prosperity in the area of electronic commerce.

Privacy policies help businesses to inform users of their data-collecting practices, ideally putting visitors at ease so they feel uninhibited in their participation. However, many users do not read such privacy policies, believing them to be too time-consuming to read or too difficult to understand. The Platform for Privacy Preferences Project (P3P) [11] enables Web sites to express their privacy policies in a machine readable format, allowing P3P user agents to read and "understand" policies on behalf of the user. Cranor et al. have worked extensively on user interfaces for these agents, including the AT&T Privacy Bird [10].

Even if an information sender is well-informed of the privacy policy, he/she might still find the data collection too intrusive. In this case it may be important for the receiver to either offer some form of incentive, or at least make the sender understand the benefits of transmitting their data. Recent work in privacy economics research shows that people are typically willing to share their private information if they foresee a sufficient reward in return. Chellappa and Sin [6] and Culnan and Bies [14] argue that, when attempting to collect user data in order to personalize Web sites, consumers are willing to share preference information in exchange for benefits such as convenience if the quantified value of services outweighs the quantified loss of privacy. Hann et al. [18] show that economic incentives affect users' willingness to share information, and derive consumers' monetary worth of secondary use of personal information. Cheskin Research [7] found that many expert users (younger males especially) are willing to sacrifice more private information if it leads to better prices.

While people understand that their information has value, since different users value their information differently, techniques are needed to help determine these values in order to facilitate effective decision making. Utility elicitation techniques [5] can be used to help senders determine how they privately value their personal information. Once these values have been assessed, since information receivers are not likely to agree on such values, negotiation may take place to determine a suitable exchange. Earlier drafts of P3P included a protocol for multi-round negotiation. However, it was believed that this made P3P too complicated, and it was thus dropped from the specification. Cranor and Resnick [12] show that under assumptions of user anonymity, publicly known Web site strategies, and no negotiation transaction costs for users, take-it-or-leave-it offers yield just as much Web site profit as any negotiation strategy. On the other hand, Buffett et al.[2] show that when rewards are offered these assumptions are no longer valid, and give a protocol for multi-issue automated negotiation [17], [19], [20] where the information receiver can offer a certain level of service (*e.g.*, 10% discount, free delivery) in exchange for private information. Multi-attribute utility theory [21] is used to rank each party's preferences.

In this paper, we present a tool for determining and conducting fair private information exchanges. The MONOLOGUE (Multi-Object Negotiator for On-Line Offers Guided by Utility Elicitation) system provides a negotiation engine that works autonomously on behalf of the user towards finding an acceptable exchange. To facilitate effective negotiation, the MONOLOGUE implementation includes special utility elicitation capabilities that allow it to learn the user's specific privacy preferences with minimal interaction with the user. In addition, MONOLOGUE uses special statistical inference techniques to learn preferences and goals of the opposing negotiator (i.e. the website) during negotiation. This not only accelerates the negotiation process, but also aids in strategy computation and thus helps the negotiator achieve better deals on behalf of the user.

II. An Example Private Information Exchange Scenario

Consider an online shopper searching for a new book at a (fictitious) website, *Amazin.com*. After a short search, the book is found to be for sale at a cost of $34. Carefully assessing all pertinent factors such as current bank balance, bills, future income and personal interest in obtaining the book, the buyer concludes that $34 is a reasonable price to pay, and agrees to the transaction.

While a buyer might not actually do a full analysis of his financial position each time a purchase decision such as this is made, factors such as these will typically have some sort of indirect influence on whether a candidate purchase is made. The point is that one can fairly easily determine the costs and benefits of the purchase. However, such costs and benefits quickly become unclear when private information exposure is part of the deal.

Consider again the above buyer. After accepting the offered price, the buyer is told that some personal information is needed in order to set up an account. In particular, the website requires his name, address, credit card number and phone number. The buyer is also informed that, as a new member, he can sign up for "The Amazin Club". The club entitles the member to benefits such as a $30 discount on his next purchase, automatic updates on sales and website personalization. The cost is free; all that is needed is the buyer's e-mail address, birthday, and some information on his interests and hobbies. Finally, the buyer is informed that he can join the "Amazin Community", where members can suggest books to each other. Here the buyer only needs to provide the names and e-mail addresses of ten friends and/or family members.

Determining how to proceed in this case is clearly not as simple. To start, it is much more difficult to determine the possible costs associated with such exposure. Will the information be used only as promised in the website's privacy policy? Even if the information is used properly, what are the possible consequences? Will the buyer receive junk mail at home, phone calls or spam? Are more serious consequences possible, such as those that could jeopardize his professional career? Economic concerns are important as well. The information has obvious value for the requestor, but are they offering enough in return? Would they be willing to offer more benefit, or perhaps settle for less information for the same reward?

The objective of MONOLOGUE is to help the user to determine the fair value of his private information, given market values as well as personal preferences, and to negotiate with the website to determine an exchange such that the costs and benefits are fair to both sides. MONOLOGUE has the added goal of carrying out this preference elicitation and negotiation as quickly as possible, and with minimal interaction with the user. The remainder of this paper discusses the MONOLOGUE system model and some of the theory used in the implementation to achieve these goals.

III. Automated Negotiation

Several issues arise in the definition of an automated negotiation system. Here we assume that negotiations are done by alternating offers - no other information is exchanged between participants. We partition the set of such negotiations according to the content of offers: one issue or several issues, and for each issue whether it is scalar-, discrete- or set-valued. Note that some issues have a natural orientation; customers want to pay less money, and information holders want to divulge less data. This imposes a mutually known partial order over the set of values of that attribute (which our protocol will exploit in Section IV B). An example of a single-issue negotiation with a scalar-valued issue would be haggling over the price of a single item, where price is a scalar and the owner of the object seeks to increase the price. When money is exchanged, it is (almost) always a scalar value. Sometimes discrete values are used, where the discussion is over the choice of one of several possible colours, for example. Another type of discrete-valued attribute would be a binary one that represents the inclusion or exclusion of a special feature, such as an extended warranty. Often set-valued attributes are used in offers. For example, a carpenter might negotiate to perform a set of tasks for a mechanic in exchange for a set of the mechanic's services on the carpenter's car. In this multi-issue case each offer is a pair of sets: the carpenter's services and the mechanic's services. For a different example, if a set of carpentry services and a set of mechanic's services are exchanged for money, each offer would be composed of two sets of services and a scalar price.

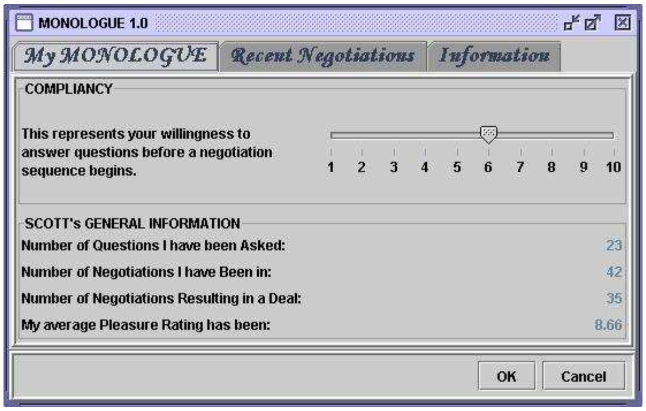

Fig. 1. The initial screen of the MONOLOGUE user interface.

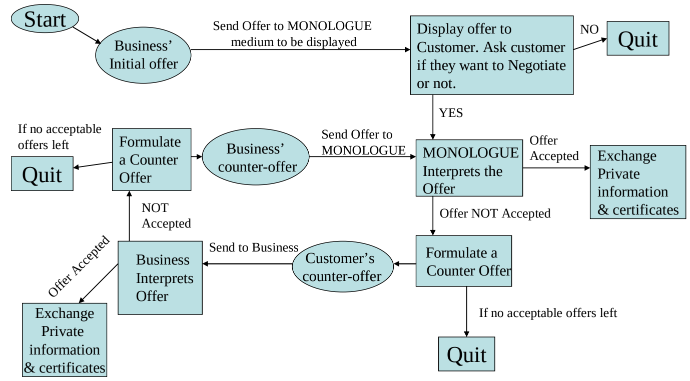

Fig. 2. Flow diagram of the MONOLOGUE system.

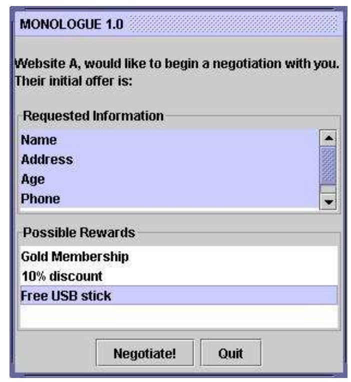

Fig. 3. The MONOLOGUE screen displaying the details of an information request initiated by a website.

IV. The MONOLOGUE System

A. System Overview

The MONOLOGUE system has a two-tiered architecture where the client-side application interacts with the user to determine the user preferences, and also informs the user of the current negotiation progress as well as histories of previous negotiations. The client also devises the negotiation strategies, sends and receives offers, and performs decision-making. The server-side maintains a database consisting of utilities of other (anonymous) registered users for various private items as they pertain to various websites. This database is used to help determine the specific user's preferences, using a process described in more detail later in the paper. Figure 1 depicts the initial MONOLOGUE screen where the user can specify his settings and preferences off-line, and examine various results and statistics of previous negotiations.

The flow diagram in Figure 2 shows the MONOLOGUE system at work. The process is initiated by a request for private information by a website (see Figure 3). This request, along with the benefits that would be obtained by the user by consenting to such a release, is considered to be the initial offer. After this offer is displayed, the user may choose to either negotiate, or terminate the interaction with the website altogether. The negotiation process then consists of one or more cycles. In the first step of the cycle, the website's offer is considered. Using a given negotiation strategy, the MONOLOGUE negotiation engine determines whether to accept the offer or decline and submit a counteroffer. If no valid counteroffer exists under the rules of the negotiation protocol, the negotiation may be terminated. A message is then composed according to the protocol that either indicates the decision to accept or quit, or proposes a counteroffer. This message is transmitted to the website by the communication engine, which then awaits a response. The process continues until either party either accepts an offer or quits. If an offer is accepted it is considered binding, and the exchange of private information, reward tokens, certificates, and any other components of the deal takes place.

B. Negotiation Protocol

We consider a two-participant bilateral negotiation where each participant is self-interested and has incomplete information about the opponent. Information is incomplete in that a participant is unsure not only about the opponent's reserve limits and deadlines, but also about its preference ranking of possible offers. The PrivacyPact protocol [2] is a protocol for alternating-offers bilateral negotiation of private information exchanges. Each offer under the protocol consists of two components: a P3P statement and a reward. A P3P statement specifies the contents and terms pertaining to a private information exchange. In particular, a P3P statement dictates the data to be exchanged, the recipients, the purposes and the time for which the data will be retained. Rewards could include discounts on merchandise, free software or document downloads or air miles.

While utilities for statements and rewards are privately known, a partial order of each negotiator's preferences is mutually known. Specifically, we assume that the website necessarily values a statement s no more than another statement s' if the data, recipients and purposes specified in s are subsets of those specified in s' and the retention time specified in s is no longer than that in s'. For the user, s' is valued no more than s. We denote this as $s \preceq s'$, meaning that "the information specified in s is necessarily less than or equal to that specified in s'". Similarly, since some rewards are mutually agreeable to be "more" or "better" than others (e.g. items with monetary value), we assume that a partial order exists over the set of reward tokens. If a token t is no larger than a token t' (denoted by $t \preceq t'$), then the website values t' no more than t and the user values t no more.

These partial orders are used to ensure that each negotiator attempts to make progress. In particular, a negotiator cannot make an offer that is necessarily worse to the opponent than a previous offer. Thus the website cannot offer $o = \langle s, t \rangle$ if it previously made an offer $o' = \langle s', t' \rangle$ such that $s' \preceq s$ and $t \preceq t'$. Similarly the user cannot offer $o = \langle s, t \rangle$ if it previously made an offer $o' = \langle s', t' \rangle$ such that $s \preceq s'$ and $t' \preceq t$. If a negotiator makes an offer that is necessarily as good or better according to this partial order than an offer previously made by the opponent, the negotiation ends with this last offer being the agreement.

To illustrate the point, we demonstrate a simple example negotiation, where the business is interested in obtaining the customer's salary, name, phone number and email address. In practice, at least one purpose and one recipient as well as a retention time are needed in a P3P statement, but for simplicity we omit these in the example. After establishing these desired data elements, the customer then responds with a subset of these items, perhaps omitting the salary element. This indicates that he is not interested in divulging any information on salary, no matter what the business is offering, and effectively takes it off the bargaining table. Also, consider the set T of reward tokens presented to the customer to contain three levels of service: Silver, Gold and Platinum, where Silver $\preceq$ Gold $\preceq$ Platinum. This forms the initial domain of negotiation and concludes the initial phase.

In order to negotiate, each participant $p \in \{b, c\}$ determines his own utility u_p for each potential offer. Since utility is typically normalized from 0 to 1, we set the utility of the best option to 1 and the worst to 0, and the utilities for all other offers to sensible values in between. In the customer's utility table (Table I(a)), we see that giving out the name, phone number, and email in exchange for Gold service has a utility value of .3. On the other hand, the business values this offer with utility .65 (Table I(b)). Note that each participant's utility table is not visible to the other participant in a negotiation. Also, let the utility acceptability threshold be $\alpha_c = .55$ for the customer and $\alpha_b = .5$ for the business. The areas that are outlined by dark black are the acceptable offers. For instance, the customer may accept an offer from the business asking for the phone number in exchange for Gold service. The cells with darker gray background are the (unknown) mutually acceptable deals. For example, one such deal involves the customer divulging his name and the business offering gold service in return.

A sample negotiation session is listed in Figure 4. Private utility values for each participant are given for each offer. In step 7 the business's offer indicates acceptance since it is at least as good for the customer as one of the customer's previous offers (in step 6). Finally, the certification phase is reached and the transaction is completed.

Note that in this example the protocol would not allow the business to make an offer at step 7 of, say, $\langle \{n, e\}, Silver \rangle$, since this is necessarily worse to the customer than a previous offer (given in step 5). This prevents a participant from purposely making offers that wear on the patience of the opponent, thus possibly persuading him to accept the last reasonable offer. However, keeping in the spirit of negotiation, a participant is in no way bound to accept the first acceptable offer received. For example, the business could have countered in step 7 with $\langle \{n, p\}, Gold \rangle$, and possibly reached a better deal. So our protocol does not restrict any reasonable bargaining strategies.

	{n}	{p}	{n,p}	{e}	{p,e}	{n,e}	{n,p,e}
Silver	.4	.35	.3	.25	.1	.05	0
Gold	**.7**	.65	**.6**	.55	.4	.35	.3
Platinum	1	.85	.8	.75	.6	.55	.5

(a) Customer utility table (n = name, p = phone, e = email, α=.55)

	{n,p,e}	{n,p}	{n,e}	{n}	{p,e}	{p}	{e}
Platinum	.5	.45	.35	.3	.2	.15	0
Gold	.7	**.65**	.55	**.5**	.4	.35	.3
Silver	1	.95	.85	.8	.7	.65	.6

(b) Business utility table (n = name, p = phone, e = email, α=.5)

TABLE I

UTILITY VALUES FOR EACH PARTICIPANT IN THE EXAMPLE NEGOTIATION.

Business	u_b	u_c	Customer	u_b	u_c
1. $\langle \{n, p, e\}, Silver \rangle$	1	0	2. $\langle \{n\}, Platinum \rangle$	.3	1
3. $\langle \{n, p\}, Silver \rangle$	.95	.3	4. $\langle \{p\}, Platinum \rangle$	.15	.85
5. $\langle \{n\}, Silver \rangle$	.8	.4	6. $\langle \{n\}, Gold \rangle$	.5	.7
7. $\langle \{n\}, Gold \rangle$ DEAL!	.5	.7			

Fig. 4. A sample negotiation session using the PrivacyPact protocol after the initial phase

C. Utility Elicitation

In order to make intelligent decisions during privacy negotiations, a system must understand the user's utilities for the various consequences that could arise as a result of an agreement that is negotiated with a website. Different users might have very different preferences when it comes to giving up their private information, and a particular user might have very different opinions about releasing a specific piece of information, depending on the particular website involved.

As part of the utility elicitation process, the system should inform users of the possible consequences of releasing certain pieces of private information. For example, suppose a website is seeking the following information from the user: name (n), address (a), e-mail address (e), company (c), academic institution (ai), student number (s) and phone number (p). Thus the entire set of information units being sought is denoted by $IU = \{n, a, e, c, ai, s, p\}$. The set A of actions that could occur as a result of releasing the information in IU is

Spam	Sp	:- e
Junk mail	J	:- n, a
Telemarketing	T	:- n, p
Get grades	G	:- ai, s
Notice to boss	B	:- n, c
Visit from salespeople	Sa	:- a

where J :- n, a denotes that junk mail is a possible consequence when name and address are given up.

We also estimate the probability that each of these actions would be performed by the party receiving the information. The utility (cost) of giving away a set of information units is calculated based on the expected utility of the actions that could be performed as a result.

It can be very difficult to obtain complete information about a user's utilities for all possible outcomes. We follow the approach of Chajewska et al. [5], in which utilities are treated as random variables drawn from known distributions, based on data gathered from other users. Our goal is to ask the user questions about his privacy preferences that will reduce the uncertainty in these distributions. By gaining new information and reducing our uncertainty, we can improve the expected utility of following our chosen negotiation strategy.

Given the distributions D and a strategy π, which is the sequence of decisions to be made in the current problem (and the criteria used for making those decisions), the expected utility $Eu[\pi|D]$ of the strategy given the distributions is computed. The goal is then to determine the question to ask the decision-maker that will provide the most valuable information. Let q be such a question with n possible answers. If the user gives the ith answer with probability $p(i)$ and the resulting distributions given this new information are D_i, then the posterior expected utility after asking q is

$$\sum_{i=1}^{n} p(i) Eu[\pi|D_i]$$

At any given time, we ask the question q that yields the greatest increase in expected utility.

Typical questions follow the standard gamble pattern [21]. For example, a user might be asked whether he would prefer a situation in which he would receive junk mail for certain or a lottery in which he would receive nothing with probability 0.5 and would be phoned by telemarketers with probability 0.5. If we already know the user's utilities for receiving nothing and for being phoned by telemarketers (perhaps 1 and 0, respectively), then this question improves our estimate of the user's utility for receiving junk mail, essentially by telling us whether it is above or below 0.5. A more detailed description is given by Buffett et al. [3].

One of the most challenging aspects of the utility elicitation process is the task of composing prior distributions – in particular, the task of modeling the dependencies between utilities. For example, a user who is particularly unhappy with the prospect of receiving future offers from a company by phone is also likely to be averse to the idea of receiving spam e-mail. To account for these dependencies, we model the system of outcomes as a Bayesian network. Figure 5 gives a simple example of such a network, where five consequences are shown:

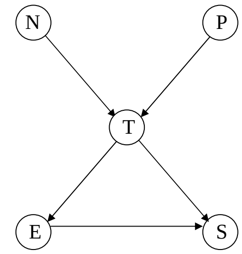

Fig. 5. An example Bayesian Network modeling dependencies in utilities for privacy consequences

N: Company A gets name, keeps in database
P: Company A gets phone number, keeps in database
T: Receive unsolicited phone calls from company A for 1 year
E: Company B will be provided with e-mail address
S: Receive spam

The network in Figure 5 indicates that the user's utility for telemarketing is dependent on his utilities for giving up his name and his phone number, and his utility for spam is dependent on his utility for telemarketing and for having a third party receive his e-mail address. While some of these dependencies may be moderate, such as spam's dependency on telemarketing, there is likely a high interdependency between two very similar consequences, such as third party e-mail and spam. It is likely that if we can ascertain the user's utility for having a third party receive his e-mail address, we can be quite certain about her utility for receiving spam.

Judging these conditional probabilities can be difficult. Utilities for telemarketing almost certainly depend on those for name and phone number, but the degree of these dependencies can be difficult to estimate. Very low utilities for name and phone number would certainly imply low utility for telemarketing, but high utilities for name and phone number (indicating that the user would not mind very much giving either piece of information away separately) certainly does not necessarily imply that the utility for telemarketing is high. We utilize the approach proposed by Cooper and Herskovits [9], which builds a Bayesian Network completely from data. In this way, starting with some initial sample utilities, the network can essentially be learned from the true utilities that are derived from each user. Each time a new user's utilities are elicited, these can be added to the database (perhaps maintained by a web service), and the Bayes' net and corresponding conditional probability distributions can then be updated.

As discussed earlier, the task of choosing the next question to ask is a matter of determining the question that will yield the greatest increase in the expected utility of our chosen negotiation strategy. However, we want to ensure that our model respects the fact that a constant barrage of questions from the system is likely to annoy the user to the point that he is unlikely to want to use the system at all. Taking this into consideration, we have developed a preliminary formula for the *bother cost* [3] associated with asking the user a question at a particular time. This bother cost is modeled as a function that depends on the user's self-reported willingness to interact with the system and also on the degree of bother to which the user has been subjected in recent sessions. The system then uses this bother cost to decide when to terminate a question period with the user: it quits as soon as the bother cost exceeds the expected gain in utility from asking the best available question.

D. Negotiation Strategy

The MONOLOGUE negotiation strategy mechanism is based on the idea of similarity maximization when making tradeoffs [16], and is described in more detail by Buffett et al. [2]. In single-issue negotiation, there is typically a clear understanding of the preferences for each negotiation party. For example, when the issue is money, each party knows the opponent's preference: the buyer prefers less and the seller prefers more. In multi-issue negotiation, the opponent's preferences for some issues may be unclear (e.g. colour of a car). Even if preferences over attribute values are clear, preferences over combinations of attribute values might not be obvious. It may be known that someone likes red cars and Corvettes, but they may not like red Corvettes. MONOLOGUE currently employs a strategy that attempts to make offers that are similar to those previously made by the opponent, with the hope that such offers are likely to be more preferred than less similar offers. This co-operative approach is likely to help guide the negotiation process to a faster convergence. Only offers that fall into a target utility range for the user are considered. This means that the user should be virtually indifferent over the set of candidate offers, since satisfactory tradeoffs are made in the attribute values. Thus the offer from this set with highest similarity to the opponent's offers will have satisfactory utility for the user and should have relatively high utility for the opponent. The attitude of the strategy can be varied from miserly, accomplished by using a small range of utilities which will include only the very best offers for the user, to co-operative, where a larger range is used and is thus likely to produce better offers for the opponent.

V. Experimentation

In this section, we test the performance of the protocol utilized by MONOLOGUE. The PrivacyPact protocol is tested for two reasons: first as a demonstration of its feasibility, and second to experiment with various strategies for dealing with the possibly exponentially long negotiations. Our initial intention is not to study how a highly effective negotiation strategy can be built, but rather to demonstrate that simple strategies can be effective for reducing the lengths of the conversations, and thus provide assurance that the protocol is not without merit.

A. Assessing the utility of an offer

The first task is to create a utility value for each exchange for each of the two participants in the negotiation. This function maps each P3P statement and token to a real value from $[0, 1]$ representing the utility of that exchange for that participant. This function should meet two criteria: transparency and smoothness. We consider each of these in turn.

It should be transparent to a participant that the utility assignment accurately reflects his opinions about the relative importances of the various aspects of the offer and combinations of these aspects. Thus it is essential that the participant's opinions about these importances are expressed in a simple way. We provide two types of statements that allow the participant to express these importances: for individual items, and for combinations of items. Recall that each offer exchanges a tuple $\langle d, r, p, \tau \rangle$ for some token t, where d is a subset of the data D, r is a subset of recipients R, p is a subset of purposes P, and these are the multi-valued attributes. Also, τ is a real-valued duration and t is a token selected from T, and these are the single-valued attributes. For the purposes of this experiment, user utilities are produced as follows. In the first type of statement, the participant gives each item in $D, R, P, \Re$ and T a number in the range $[0, 1]$ that represents his opinion on the importance of this item. For each such item e let $l(e)$ be this assigned value. These numbers are directly translated into utility values as follows, where e is from an attribute Y and this participant is on the receiving end for this item (the business recieves data, recipients, purposes and retention times while the customer recieves tokens):

$$u(e) = \begin{cases} l(e)/\Sigma_{i \in Y} l(i) & \text{if } Y \text{ is a multivalued attibute} \\ l(e)/\max_{i \in Y} l(i) & \text{if } Y \text{ is single-valued} \end{cases} \tag{1}$$

These values for the importance numbers can come from any distribution, and this equation scales them to a number in $[0, 1]$. If the participant is the holder of this attribute (i.e. the business holds tokens while the customer holds data, recipients, purposes and retention times), then the utility value is one minus the value of u computed by Equation 1. There is less utility in giving away more important items.

For example, a business that wants to receive a customer's name most and email address least, where the Y attribute also contains phone number, might express that $l(\text{name}) = 9, l(\text{email}) = 1$, and $l(\text{phone}) = 5$. Then $u(\text{name}) = 9/15 = 0.6$, while $u(\text{email}) = 1/15$. A customer who shares the same opinions of relative importance would have a utility of $1 - 0.6 = 0.4$ for giving his name.

A combination of several items is given the utility equal to the sum of the utilities of these items, except when there is information from the user that gives such a set special importance, which is considered in the next paragraph. In the absence of any special instructions from the business, the offer of a name and email would be given utility of $9/15 + 1/15 \approx 0.67$.

Also a participant might express that a combination of items has a special importance. For instance the business may want to express that receiving a name and phone number combined has a higher importance, since it may be used to identify that person uniquely. In this case the participant would express that the combination has higher importance, perhaps 18. The utility of receiving a name and phone number would be based on considering the name and phone number combination to be one item, redefining Y accordingly, and applying Equation 1. Keeping $l(\text{email}) = 1$, and defining $l(\text{name-and-phone}) = 18$ we have that $u(\text{name-and-phone}) = 18/19 \approx 0.947$.

After the utilities from each of the four data dimensions are considered, the utility $u^s(\langle d, r, p, \tau \rangle)$ of a statement is calculated. In this experiment we multiplied the four utilities together. That is, $u^s(\langle d, r, p, \tau \rangle) = u(d) \times u(r) \times u(p) \times u(\tau)$. Thus each component counts equally and the final utility of a statement is a number in $[0, 1]$. Once the utilities for tokens has been determined, the utility $u(s, t)$ of an offer is calculated using the bilinear function

$$u_z(s,t) = k_z^s u_z^s(s) + k_z^t u_z^t(t) + k_z^{st} u_z^s(s) u_z^t(t) \qquad (2)$$

for all $s \in S$ and $t \in T$, where k_z^s, k_z^t and k_z^{st} are scaling constants which sum to 1 (refer to Keeney and Raiffa [21], for example). In our experiments we assigned statement and token utility equal weight (i.e. $k^s = k^t = 0.5$, $k^{st} = 0$) for each participant.

We also allow that a combination of items from different sets be considered specially. For instance a participant might choose to place special importance on an offer that includes the phone number when it is allowed to be used for telemarketing purposes, combining consideration from separate dimensions. While this could be done in terms of importance measures, we found this difficult to explain to users because the different dimensions may be using different scales, so these numbers are given as utilities, as numbers in $[0, 1]$.

The utility function should also be smooth, so that similar offers that trade items of almost equal importance have similar utilities. If a utility function meets both the smoothness and the transparency criteria, it may be possible for a participant that knows how the competing participant has assessed some offers – e.g. whether some offers were assessed higher than others – to form an opinion about which items are important. We will revisit this point in the next section, and show that this smoothness can help cooperative negotiating partners to discover mutually acceptable offers within short negotiations.

B. Computing Negotiations with a Prototype

The goal of our experiment is to show that MONOLOGUE can accommodate simple strategies to give rise to short conversations. We define three negotiation strategies: "miserly" that always makes its next offer according to what is most beneficial to itself, "cooperative" whose next offer is chosen according to its similarity to any of the partner's previous offers, and "hybrid" which is a combination of the previous two.

The miserly negotiator makes a counteroffer by considering all of the valid offers, defined as those admitted by the conditions of the PrivacyPact protocol, and selects the one with maximal utility for itself, without any consideration of the opponent's previous offers. Thus a negotiation involving two miserly participants may require an excessive number of messages to converge, since the space of offers is essentially searched exhaustively to find an agreement.

The cooperative negotiator attempts to overcome this by considering all pairs of offers from two sets: the set of possible counteroffers allowed by the protocol, and the set of previous offers from the opponent. For each pair a similarity measure is determined, and the counteroffer selected is the one most similar to some previous offer of the opponent's. The similarity between a pair of offers is considered according to the similarity of each of the five dimensions. For single valued attributes, the similarity is some defined distance between the values. For instance, retention times are real numbers and the distance can be their difference. For a pair of multivalued attributes chosen from a set S, the distance is calculated according to the number of values from S that they agree upon as a fraction of the size of S. They agree on a value either if they both contain it, or both do not. They disagree if one contains it and the other does not. Once all five dimensions are considered, a linear combination of the five numbers gives the overall similarity; in our case each of data, recipients, purposes, retention time and token similarity counts as one fifth.

The hybrid negotiator combines the other two; it attempts to find good counteroffers quickly by looking first at deals most favourable to itself (i.e. with highest utility), and then choosing from these deals the one that maximizes the similarity measure. In our experiments, the number of such counteroffers was set at $n = |O|/10$ where O is the set of possible offers at the beginning of the negotiation. Thus at the beginning of the negotiation, the participant only considers the best 10% of the possible offers. Since the value of n remains fixed throughout the negotiation but the number of offers allowed by the protocol decreases, the percentage of valid offers considered increases. Thus the participant becomes more cooperative as the negotiation continues. Since one often wants to make more concessions as time elapses in a negotiation, this is still a very reasonable strategy.

In lieu of real-world examples, which do not (yet) exist, we selected a variety of examples, each with a selection of information items, recipients, purposes, retention times and tokens. Certain goals for the business were set by setting importance of certain subsets high, while goals for the customer were specified by setting low importance for some combinations.

Once the utility functions are defined, and before negotiation can begin, it is necessary to select alpha thresholds for each negotiator. This threshold specifies the lowest utility a partner has for accepting an offer. To make the negotiation as hard as possible, we set the alphas so that a small but non-zero number of offers could be accepted. We do this by considering each offer in turn and the pairs of utilities assigned by the partners. (Ordinarily no one party would have access to both of these functions.) For each pair, we selected the lower value, and from all these low values we select the highest. The offer associated with this highest low is arguably a hard exchange to find since it represents an offer not much favored by either partner. For each partner, the alpha value is assigned to be that partner's utility of this offer.

We tested the performance of each of the miserly, cooperative and hybrid customer negotiation strategies against a miserly business negotiator. This gives a clear demonstration of how quickly these simple strategies can converge. Note that each strategy was tested against the miserly negotiator rather than against a cooperative or hybrid negotiator since those results, while still much better than miserly versus miserly, were more erratic. For example, in some cases a negotiation involving two cooperative agents would take longer than one involving one cooperative and one miserly. This is because they would both try too hard to please each other and thus make convergence more difficult. Occurrences such as this were completely example-dependent and thus did not warrant consideration in our analysis, simply because our goal is only to show that the protocol can converge quickly under reasonable strategies. Table II gives the results. For each run, the number of negotiable items and the number of possible offers are given, as well as the number of messages required for convergence for each of the three strategies. For the sake of simplicity, only elements of D in the statements are negotiated. That is, the purposes, recipients and retention time are agreed upon in the initial phase. Also there are four tokens up for negotiation. Thus for $|D|$ negotiable items there are $(2^{|D|} - 1) \times 4$ possible offers. Experiments show that most of the space of offers is searched when the miserly strategy is employed. However, a considerably smaller number of exchanges are required when the hybrid and cooperative strategies are employed. This is a good indication that the protocol has potential to converge quickly when simple but effective strategies are used.

Number of negotiable items	Number of possible offers	Business Strategy	Customer Strategy	Number of messages to converge
3	28	miser	miser	24
			hybrid	18
			co-op	4
4	60	miser	miser	48
			hybrid	12
			co-op	6
5	124	miser	miser	104
			hybrid	16
			co-op	4
6	252	miser	miser	194
			hybrid	16
			co-op	10
7	508	miser	miser	388
			hybrid	20
			co-op	4
8	1020	miser	miser	818
			hybrid	32
			co-op	4
9	2044	miser	miser	1670
			hybrid	52
			co-op	38
10	4092	miser	miser	3178
			hybrid	66
			co-op	56

TABLE II

NEGOTIATION LENGTHS FOR VARIOUS PROBLEMS AND STRATEGIES

VI. CURRENT WORK

In this section we highlight a few of the major capabilities we are currently in the process of implementing in the MONOLOGUE system. Each of these components relies on existing and new research and theory, and can be viewed as important problems for research on their own. We briefly introduce each project and discuss how each addition should improve efficiency and/or performance of the overall system.

A. Eliciting Complex Utility Functions

One method for determining a user's preferences over the set of candidate policies is to compute the utility of each policy as a function of the potential cost of exposure. The higher the cost of a policy, the lower the utility one would have for agreeing to it. Ideally, one could initially determine the cost of giving away each private item and then compute the cost of any set of items as the sum of the item costs. However, these costs (and therefore utilities) are not additive. For example, a user may associate very low costs with giving away his name or his postal code, but perhaps a very high cost with giving the two away together, since these two pieces of information may make him personally identifiable. So the cost of the two may far exceed the sum. Accounting for these dependencies makes the utility function extremely complex. Here we examine two techniques with which we are currently experimenting to address this complexity.

One method takes explicit preferences from the user, along with dependencies over those preferences, and determines which of any two sets of items is preferred. For example, a user may specify that his utility for giving away his postal code is higher than that for his e-mail address, but his utility for name and postal code is less than that for his name and e-mail. Thus the presence of name effectively makes his utility for giving away his postal code less than that for his e-mail address. When several of these dependencies exist, it becomes much more difficult to determine which of two sets of data should be preferred by the user. To solve this problem, we are currently attempting to build on a technique proposed by Boutilier et al. [1]. With this technique, a dependency tree is used to help determine a sequence of intermediate sets that monotonically increase in preference level, proving that one set is preferred over the other.

The second method examines data over the set of previous users and makes use of collaborative filtering to find likely dependencies for the given user. The idea is that people similar to the user will likely have similar dependencies in their preferences. Elicitation is used to determine utilities over individual items, and collaborative filtering is then used to determine which sets are likely to be additive and which are likely to be special cases because of dependencies.

B. Classification of Opponent Preferences

In order to construct effective negotiation strategies that will maximize the user's expected utility, the negotiation engine needs to know a few things about the opponent's preferences. We are currently implementing a classification method that attempts to learn the opponent's preferences during negotiation [4]. The technique works by grouping similar candidate preference relations into classes. While it is unlikely that the opponent's preferences can be fully learned, it may be possible to place the opponent in a particular class by examining its sequence of offers thus far in a negotiation. All preference relations in a class are relatively similar. That way, we do not need to pinpoint the full preference relation with certainty; we only need to examine the evidence and determine which class is most likely to include the relation. Then any relation in the class should be reasonably close to the opponent's relation and a reasonably effective negotiation strategy can be computed. A Bayesian technique is used to determine the likelihood that the opponent's true preference relation over the set of offers lies in each class. Evidence used for classification decision-making is obtained by observing the opponent's sequence of offers and applying the concession assumption, which states that negotiators usually decrease their offer utilities as time passes in order to find a deal. Initial experiments show that the technique can find the correct class after very few offers and can select a preference relation that is likely to match closely with the opponent's true preferences.

C. System/User Interaction

As mentioned earlier, an important component of our model is a quantitative estimate of the degree of bother that will be experienced by a specific user when a question is asked at a particular time. The formula proposed by Buffett et al. [3] is still preliminary and has not been tested empirically. One of our current projects is a deeper investigation of this notion of bother cost. In addition to refining our own model, we are performing a literature search to find other approaches to quantifying the bother or annoyance experienced by users due to questions or interruptions. We will test several of these approaches by performing some simple experiments with real users working on a small problem. The goal of these experiments is to find the most accurate predictor of the true degree to which users are bothered by a particular line of questioning. By having a better representation for the bother cost, we can make better decisions about when the expected benefits of asking a question outweigh the costs.

VII. Conclusions and Future Work

The MONOLOGUE system integrates a number of innovative features: It allows multi-attribute offers where each attribute can be scalar-, discrete- or set-valued. We use it to negotiate P3P statements in exchange for rewards, so we elicit the user's utility of the information and the reward before entering the negotiation. The elicitation phase is sensitive to the user's tolerance for answering questions, so the utility of the answer is measured against the bother cost of the question. The system then performs the negotiation, according to a strategy selectable by the user which can range from miserly to co-operative. It analyses the opponent's pattern of responses and classifies the opponent, in order to better predict future counter-offers. This increases the effectiveness of the negotiation and the benefits to the user.

Simple experiments were carried out with the main goal of testing the efficiency of the negotiation protocol used by MONOLOGUE, PrivacyPact. Three simple negotiation strategies were employed: miserly, cooperative, and a hybrid of the two. Results show that, while the number of exchanges required for the negotiation to converge to an agreement are exponential in the worst case, in practice such negotiations tend to converge quite quickly. One unexpected observation was the surprising inefficiency of negotiations involving two cooperative negotiators. One explanation for this phenomenon is that since both negotiators focus on pleasing the opponent, neither really reveals their true preferences. Thus negotiations can tend to converge toward deals that are not preferred by either party.

For future work, one significant project will involve the development of negotiation strategies that will perform well against a cooperative opponent, either to work together to find agreements quickly, or to exploit the opponent's good intentions and work to find the best deal for the user. A significant effort has focused recently on finding trade-offs in offers that better accommodate the opponent's goals in order to cooperate towards a solution [8], [15]. A natural next step is to look at methods to compete or cooperate with an agent using these techniques.

We also plan to examine the usefulness of the system in domains other than privacy. Complex purchase negotiations, such as those involved in purchasing automobiles or travel packages could benefit from some of the technology created for MONOLOGUE.

References

[1] C. Boutilier, R. I. Brafman, H. H. Hoos, and D. Poole. Reasoning with conditional ceteris paribus preference statements. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 71–80, Stockholm, Sweden, 1999.

[2] S. Buffett, K. Jia, S. Liu, B. Spencer, and F. Wang. Negotiating exchanges of P3P-labeled information for compensation. *Computational Intelligence*, 20(4):663–677, 2004.

[3] S. Buffett, N. Scott, B. Spencer, M. M. Richter, and M. W. Fleming. Determining internet users' values for private information. In *Second Annual Conference on Privacy, Security and Trust (PST04)*, pages 79–88, 2004.

[4] S. Buffett and B. Spencer. Learning opponents' preferences in multi-object automated negotiation. In *Proc. of the 7th International Conference on Electronic Commerce (ICEC2003)*, Xi'an, China.

[5] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. In *AAAI-00*, pages 363–369, Austin, Texas, USA, 2000.

[6] R.K. Chellappa and R. Sin. Personalization versus privacy: An empirical examination of the online consumers dilemma. *Information Technology and Management*, 6(2-3), 2005. To appear.

[7] Cheskin Research . Trust in the wired americas, July 2000. Available from http://www.cheskin.com/.

[8] R. M. Coehoorn and N. R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In *Proc. of the 6th International Conference on Electronic Commerce (ICEC2004)*, pages 113–120, Delft, The Netherlands, 2004.
[9] G. F. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
[10] L. Cranor, M. Arjula, and P. Guduru. Use of a P3P user agent by early adopters. In *Proceedings of the ACM workshop on Privacy in the Electronic Society*, pages 1–10. ACM Press, 2002.
[11] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The Platform for Privacy Preferences (P3P) 1.0 Specification. http://www.w3.org/TR/P3P/, 16 April 2002. W3C Recommendation.
[12] L. Cranor and P. Resnick. Protocols for automated negotiations with buyer anonymity and seller reputations. *Netnomics*, 2(1):1–23, 2000.
[13] M.J. Culnan and P.K. Armstrong. Information privacy concerns, procedural fairness, and impersonal trust: An empirical investigation. *Organization Science*, 10(1):104–115, 1999.
[14] M.J. Culnan and R.J. Bies. Customer privacy: Balancing economic and justice considerations. *Journal of Social Issues*, 59(2):104–115, 2003.
[15] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.
[16] P. Faratin, C. Sierra, and N.R. Jennings. Using similarity criteria to make issue trade-offs in automated negotiations. *Artificial Intelligence*, 142:205–237, 2002.
[17] S.S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, 152(1):1–45, 2004.
[18] I. Hann, K. Hui, T.S. Lee, and I.P.L. Png. Online information privacy: Measuring the cost-benefit trade-off. In *23rd International Conference on Information Systems*, 2002.
[19] N. R. Jennings, P. Faratin, A. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 10(2):199–215, 2001.
[20] N. R. Jennings, S. Parsons, C. Sierra, and P. Faratin. Automated negotiation. In *5th International Conference on the Practical Application of Intelligent Agents and Multiagent Systems (PAAM-2000)*, pages 23–30, Manchester, UK, 2000.
[21] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. John Wiley and Sons, Inc., 1976.
[22] M. J. Metzger. Privacy, trust, and disclosure: Exploring barriers to electronic commerce. *Journal of Computer-Mediated Communication*, 9(4), 2004.
[23] G. J. Udo. Privacy and security concerns as major barriers for e-commerce: a survey study. *Information Management and Computer Security*, 9(4):165–174, 2001.