## Applying a Preference Modeling Structure to User Privacy

Buffett, Scott; Fleming, Michael



*Workshop on Sustaining Privacy in Autonomous Collaborative Environments (SPACE 2007), held in conjunction with the IFIP Conference on Trust Management (IFIPTM07) [Proceedings], 2007*

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

# *Applying a Preference Modeling Structure to User Privacy**

Buffett, S., Fleming, M.W.
July 2007

* published at the Workshop on Sustaining Privacy in Autonomous Collaborative Environments (SPACE 2007), held in conjunction with the IFIP Conference on Trust Management (IFIPTM07). Moncton, New Brunswick, Canada. July 30, 2007. NRC 49372.



Canada

# Applying a Preference Modeling Structure to User Privacy

Scott Buffett[1] and Michael W. Fleming[2]

[1] National Research Council Canada, Fredericton, NB, E3B 9W4
Scott.Buffett@nrc.gc.ca
[2] University of New Brunswick, Fredericton, NB, E3B 5A3
mwf@unb.ca

**Abstract.** In this paper, we demonstrate how a preference elicitation technique can be applied in the domain of user privacy. A structure known as a Conditional Outcome Preference Network (COP-network) is used to model preferences and estimate utilities for various private data collection practices. These utilities can then be used by an autonomous software agent to advise an Internet user on whether or not to interact with a website, or even to facilitate or conduct negotiations of a mutually acceptable privacy policy. Experiments show that preferences and utilities are estimated significantly better than by a previously used technique.

## 1 Introduction

Whether they realize it or not, privacy is a prevalent issue for Internet users virtually every time a web site is visited. Banking websites ask for financial information. News and information websites often ask a user to register and provide some personal data before providing free access to content. These are examples of explicit requests for information that a user can choose to accept or decline. Websites can be more subtle in how they acquire information by leaving a "cookie" (a small piece of data) on the user's machine. This can be used to remember what actions the user took during previous visits to the website. Thus search engines can build a profile for a user by tracking and analyzing search history, for example.

The main concern here is not that such data collection is being done; for the most part this collection is beneficial to the user. Of utmost importance here is the fact that there is little or no help for a user to automate the process of 1) deciding whether or not data collection is potentially malicious or damaging, and 2) facilitating the process of protecting this data. This can be done either by advising a user on what information should not be disclosed or which websites should be avoided, or by negotiating a more restrictive privacy policy that allows the user to enter and do business with the website at a privacy level at which he or she is comfortable.

The use of software agents that can help with this process has been proposed in the literature. Buffett et al. [4] discuss the use of automated agent negotiation

to determine a data-collection policy that both the user and website can agree on, given the particular benefit that the user will receive by divulging the private data. Here a new policy can be negotiated each time a user visits a site and new data is requested. All negotiation is performed quickly and autonomously behind the scenes. Cranor et al. [9] propose the use of "user agents" to interact with a website to determine whether the user's preference criteria for private information are consistent with the website's privacy policy. This technology stems from the Platform for Privacy Preferences Project (P3P), and relies on the website's usage of P3P to provide a machine-readable version of its privacy policy.

In order for any such agent to assist the user, the agent must have some prior knowledge of the user's preferences. All users are different; one user may, for example, be quite willing to divulge information such as her name and email address in order to receive automatic updates on news and weather, while another user may wish to keep such information private. Until the agent receives some specific information on privacy preferences from the particular user it represents, there is no way to know how to automate any of the aforementioned processes satisfactorily for the particular user.

Research in preference elicitation [1, 2, 6, 12] has produced techniques for extracting preference information from a human user for use by an autonomous software agent that works on the user's behalf. Such techniques may involve asking questions that will indicate the user's preference over two or more outcomes, while others attempt to assess the user's *utility* for some outcome, which gives a measure of the degree of the user's preference. One of the main challenges in preference elicitation is the fact that, while there may be a high number of possible outcomes for which user preference must be known, realistically a system should only ask a user a small number of questions. A system that bombards a user with a high number of questions about her preferences will likely overly annoy the user, and such annoyance can easily outweigh any benefit that would be achieved by using agents in the first place. Thus more successful preference elicitation techniques will be those that can infer or predict large amounts of preference information from a small amount of directly elicited data.

The focus of this paper is to show how a technique from the preference elicitation literature for modeling and predicting preferences can be applied to the specific domain of user privacy preferences. The goal is to assist the agent in building an accurate model of the user's privacy preferences, so that it can perform customized privacy-preserving activities on behalf of the user, whether it be restricting website access or conducting policy negotiation. The paper is organized as follows. In Section 2, we discuss previous literature on autonomous privacy agents and modeling user privacy preferences. In Section 3 we discuss the preference structure we use, known as the Conditional Outcome Preference Network (COP-network). Section 4 then details our method for applying COP-networks to the problem of privacy, and demonstrates how it can be used to compute a user's personal value for a possible exchange. Finally, Section 5 gives some results and Section 6 concludes the paper with some closing remarks.

## 2 Modeling Privacy Preferences

In 2002, the W3C released work on the Platform for Privacy Preferences Project (P3P) [10]. P3P is used by websites to express their privacy practice. A computerized agent, acting on behalf of the user, can fetch and read the P3P policy file, can inform the user about the site's privacy practices and can make an automatic or semi-automatic decision on behalf of the user. The P3P policy file is an XML file that is defined for certain regions of a website or the entire website. Each P3P file contains at least one statement, and each statement describes what data will be collected, with whom it will be shared, for how long it will be retained and for what purpose. A user that visits a website that expresses their privacy policy in P3P can configure a "user agent" to read and understand the website's policy. Based on the preferences specified by the configuration, the agents can decide whether or not the website's policy would be agreeable to the user. The AT&T "Privacy Bird", which can be used with Internet Explorer, appears at the top of the browser and indicates whether the site is acceptable (by appearing green) or unacceptable (by appearing red). There are several problems with this user agent. First, the preferences that a user can specify are not sufficiently expressive. Second the preferences are too rigid, as the same set of preferences are used for every website. In practice personal privacy preferences are dependent on who is receiving the information, whether the website is reputable, and what the collected information would likely be used for, among other issues. Finally, the bird relies on websites' use of P3P, for which adoption has been slow.

Utility-based models have also been considered. Here, the agent attempts to determine the user's utility for releasing a particular data item. Utility is a number typically in the $[0, 1]$ range that indicates a measure of value or worth. An outcome with high utility for a decision-maker is considered to be more preferable than an outcome with low utility. So, for example, a user that has strong reservations about revealing her email address might have low utility for an action or outcome that results in revealing that e-mail address. The idea is then to determine as accurately as possible the user's utility for release of each data item of interest. These utilities allow for much more flexibility in how an agent decides whether to interact with a website. For example, instead of always blocking access or warning a user whenever such undesirable data requests are received, the agent could decide on its own whether to proceed by considering such issues as the reputation of the website, or what is being received by the user in exchange for this data. For example, if the requesting website is considered quite reputable, and the data request is necessary in order to complete a purchase that user is trying to execute, the agent can look at the user's utility and make a decision on whether the data exchange would be acceptable.

The concept of utility can be advantageous in negotiation of privacy policies. Buffett et al. [4] discuss the concept of performing automated negotiation of private information exchange. By knowing the user's utility, the negotiation agent can compute the value of each offer submitted, by computing the trade-off for the specified requested private data versus the benefit the user would receive.

The problem with applying utility models in the privacy domain is that a user's utility for an exchange of two or more items often cannot simply be computed as a function of the individual utilities. This is because putting information together on an individual often reveals a great deal more information indirectly. Thus a user may want to express *conditional preferences*. For example, a user may not mind releasing information which identifies his place of employment, nor would he mind exposing his job title. However, he may have strong reservations when it comes to giving away both of these particular items of information, as together they may personally identify him. So perhaps his utility for exposing his job title is dependent on whether his place of employment is also part of the final outcome.

Given this, one can see that it is quite complex to determine a global utility function that is consistent with all preferences that can be derived given the known interdependencies. In order to determine such utilities, a preference structure is needed. We use Conditional Outcome Preference Networks.

## 3 Conditional Outcome Preference Networks

In this section, we describe a structure for representing the user's known preferences in such a way that new preferences that can be directly inferred will be immediately evident. The structure is a directed graph that represents preferences over the set of outcomes, and is referred to as a *Conditional Outcome Preference Network (COP-network)* [8]. This network is a good candidate to deal with the conditional and multi-attribute nature inherent in privacy preferences.

To give the reader some intuition on multi-attribute preference modeling, consider a simple car example with attributes "Make" and "Colour". A user may specify preferences for "Make" such as "Pontiac is preferred over Volkswagen", or "Colour" such as "Black is preferred over silver". From this, it can be inferred that black Pontiacs are preferred over silver Volkswagens, all else equal. Additionally, conditional preferences can be used. For example, consider a buyer that only likes Pontiacs that were made after 2002. Then the preference for "make" is conditional on the outcome for "year".

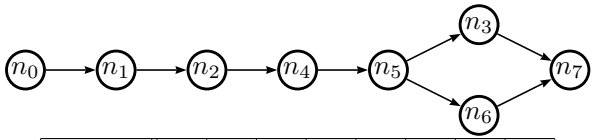
In the privacy example, attributes of outcomes correspond to items of personal information to be exchanged. Each attribute can then take on one of two values: "included in the agreement" or "not included in the agreement". Thus if $a$ represents "address", $n$ represents "name" and $p$ represents "phone number", then $anp$ represents the outcome where address, name and phone number are all included, whereas $\bar{a}np$ denotes that $a$ is not included but name and phone number are. The $\bar{a}$ is often just dropped and the preceding denoted more simply by $np$. We use $\succ$ to denote preference over two outcomes. For example "address" is preferred over "phone number" (i.e. the user would rather divulge their address over their phone number) is denoted by $a \succ p$. Conditional preferences are denoted as $n : p \succ a$ for example, which indicates that, if name is part of the outcome, then phone number is preferred over address.

Given a few initially specified user preferences, the COP-network can be used to determine other preferences that indirectly follow as a result, and give a partial model of the user's preference relation over outcomes. In a COP-network, every feasible outcome is represented by a vertex, and for vertices $n$ and $n'$ representing outcomes $o$ and $o'$, respectively, $n$ is a proper ancestor of $n'$ if and only if $o$ is preferred over $o'$.

**Example 3.1** Suppose that there is a set $\{A, B, C\}$ of attributes, and that each attribute has binary values ($a$ and $\overline{a}$ are values for attribute $A$, $b$ and $\overline{b}$ for $B$, $c$ and $\overline{c}$ for $C$), and that there are the following preferences:

$$\overline{a} \succ a \quad , \quad \overline{b} \succ b \quad , \quad \overline{c} \succ c \quad , \quad a\overline{b} \succ \overline{a}b \quad , \quad \overline{a} : b\overline{c} \succ \overline{b}c \quad , \quad a : \overline{b}c \succ b\overline{c}$$

To structure a COP-network with the above preferences, all feasible outcomes are listed: $\overline{a}\overline{b}\overline{c}$, $a\overline{b}\overline{c}$, $\overline{a}b\overline{c}$, $ab\overline{c}$, $\overline{a}\overline{b}c$, $a\overline{b}c$, $\overline{a}bc$, $abc$. Next, preference rules as dictated by the given preferences are applied to the outcomes, and a set of preferences over the outcomes is generated. For example, by applying the preference rule $a\overline{b} \succ \overline{a}b$, we can conclude that $a \succ b$, and also that $ac \succ bc$. The final step is to build a directed graph by creating a node for every outcome and adding a directed edge from node $n_i$ to node $n_j$ if the preference $o_i \succ o_j$ holds for the corresponding outcomes. The resulting graph is shown in Figure 1.




| Node: | $n_0$ | $n_1$ | $n_2$ | $n_3$ | $n_4$ | $n_5$ | $n_6$ | $n_7$ |
|---|---|---|---|---|---|---|---|---|
| Outcome: | $\phi$ | $a$ | $b$ | $ab$ | $c$ | $ac$ | $bc$ | $abc$ |

**Fig. 1.** An example COP-network. Bar values are removed from node representation table (e.g. $a\overline{b}\overline{c} \Rightarrow a$).

The graph can be used to check that the user's given preferences are consistent. A COP-network is said to be consistent if and only if there is no outcome that is preferred over itself - *i.e.*, if and only if the network is acyclic. If a COP-network corresponding to a given set of preferences is found to have a cycle, then the user must be consulted in order to correct the inconsistency.

The graph is also transitively reduced by the removal of redundant edges. For example, for vertices $n_i$, $n_j$ and $n_k$, if there are two paths $n_i \rightarrow n_j \rightarrow \cdots \rightarrow n_k$ and $n_i \rightarrow n_k$, the second path (i.e. the arc from $n_i$ to $n_k$) is not necessary since preferences that are reflected by the first path include the preference that the second path reflects. Thus, the arc $(n_i, n_k)$ is said to be *redundant* and can be removed.

In addition to modeling the user's preferences during the elicitation stage, the COP-network can also be used to estimate a utility function over the set

of outcomes. Given an initial partial utility assignment, including at least the most preferred outcome (utility 1) and the least preferred (utility 0), and the preferences defined in the COP-network, a utility function $\hat{u}$ over the entire set of outcomes is produced. This is done in such a way as to preserve the preference ordering specified by the COP-net. Specifically, let $n$ and $n'$ represent outcomes $o$ and $o'$. If $n$ is a proper ancestor of $n'$, then $\hat{u}(o) > \hat{u}(o')$.

The method iteratively selects paths in the graphs containing outcomes for which a utility has not yet been assigned. Formally, let $p$ be a path in the network with endpoints representing outcomes $o_1$ and $o_n$. This path is a candidate for selection if it is a longest path such that:

- $\hat{u}$ is known for $o_1$ and $o_n$
- $\hat{u}$ is unknown for all other outcomes represented by vertices on $p$
- The assignment of utilities to such outcomes will not cause an inconsistency in the graph[3]

Once a suitable path $p$ has been selected, the utility $\hat{u}$ is assigned for each outcome $o_1, o_2, \ldots o_n$ represented on $p$, decreasing from $o_1$ to $o_n$, by

$$\hat{u}(o_i) = \hat{u}(o_n) + \frac{(n-i)(\hat{u}(o_1) - \hat{u}(o_n))}{n-1} \tag{1}$$

For example if $p$ represented four outcomes with $\hat{u}(o_1) = 0.8$ and $\hat{u}(o_4) = 0.2$, then $\hat{u}(o_2)$ and $\hat{u}(o_3)$ would be assigned utilities of 0.6 and 0.4, respectively. The process of selecting paths and assigning utilities in this way continues until all outcomes are considered.

## 4 Using COP-networks to Model Privacy Preferences

### 4.1 Modeling Structure

Any time a website collects private data, the terms of collection practice as specified by the privacy policy typically contain four elements:

1. the data to be collected
2. the purpose for collection
3. who will receive the data
4. the retention policy

Table 4.1 shows some selected examples of specified data, purposes, recipients and retention policies[4]. For elements 1, 2 and 3 above, some non-empty subset of items will be present, while for element 4 exactly one item will be specified. Thus in order to model user preferences over various data collection outcomes, one must be able to compute utilities for sets of data, sets of purposes, sets of recipients and for each retention policy. Four COP-networks are then needed,

[3] Refer to Chen [7] for more on ensuring consistency in path selection.
[4] See [10] for a more complete set of examples.

one for each element, where outcomes in COP-networks 1-3 are sets of items, and outcomes for COP-network 4 are single items. It may seem that building a structure to model all subsets of data elements, for example, would be infeasible due to the large number of outcomes. However, we need not model all outcomes, but rather all outcomes that a user cares to specify preferences over. This is likely to be a small, manageable set in most reasonable cases. All outcomes that are not specifically addressed by a user can be given utilities that are consistent with the general population, and essentially not be given extra attention here.

**Table 1.** Example privacy collection practice specifications

| Data | Purposes | Recipients | Retention |
|---|---|---|---|
| user.name | telemarketing | ours | no-retention |
| user.home-info.postal | admin | delivery | legal-requirement |
| user.bdate.ymd.year | tailoring | other-recipient | business-practices |
| business.name | contact | public | indefinitely |

More formally, let $D$, $P$, $R$ and $T$ be sets of possible data, purpose, recipient and retention policy values, and let $D_d$, $P_d$, $R_d$ and $T_d$ be the respective domains useable in a data collection specification particular to each. Then $D_d = 2^D$, $P_d = 2^P$, $R_d = 2^R$ and $T_d = T$. Finally let $C_D$, $C_P$, $C_R$ and $C_T$ be COP-networks modeling user preferences over $D_d$, $P_d$, $R_d$ and $T_d$.

### 4.2 Eliciting Preferences

Users can specify preferences in one of two ways: using binary comparisons and standard gambles.

Binary comparisons are the simpler of the two. Here a user simply states which of two outcomes is preferred. For example, a user can indicate that "revealing my phone number is preferred over revealing my e-mail address", or that "revealing my name and phone number is preferred over revealing my email address, age and occupation". As mentioned previously, a user can also specify conditional preferences, such as "given that my name is revealed, I would prefer to reveal my email address over my phone number".

The use of standard gambles is slightly more complicated, but reveals more direct information regarding user utilities. Let $o_1$, $o_2$ and $o_3$ be outcomes such that utilities $u(o_1)$ and $u(o_3)$ are known, and $u(o_2)$ is unknown but known to be $u(o_1) \leq u(o_2) \leq u(o_3)$. Often the worst and best outcome (with utilities 0 and 1, respectively) are used for outcomes $o_1$ and $o_3$, respectively. Through a series of gamble questions, the system and user then work on zeroing in on a probability $p$ where the user would be indifferent between the following two events:

1. taking a gamble where he would receive $o_3$ with probability $p$ and offer $o_1$ with probability $1 - p$
2. receiving $o_2$ for sure

If the user is indifferent between the two events, then his utility for each is equal. And by utility theory, since the utility of the first event is equal to the expected utility of the outcomes, which is known, the utility of $o_2$ can be computed by

$$u(o_2) = u(o_3)p + u(o_1)(1-p) \tag{2}$$

This elicitation process stops when the expected benefit of receiving an answer from the user drops below a specified threshold or drops below the expected cost associated with bothering the user [5].

### 4.3 Determining Utilities for Information Exchange Outcomes

Given all of these pieces, the last remaining issue is determining the utility for a complete outcome consisting of a set of data, a set of purposes, a set of recipients and a retention policy. Let each of these four elements make up an attribute of the final data collection outcome. We denote the attributes as $D$, $P$, $R$ and $T$, with domains $D_d$, $P_d$, $R_d$ and $T_d$. Each outcome $o = \langle d, p, r, t \rangle$ is then a member of $D_d \times P_d \times R_d \times T_d$. In this section, we demonstrate how utility is computed for such an outcome.

Initially, a utility function is computed for each attribute using the COP-network utility computation described above. This yields four functions: $u_D : D_d \rightarrow \Re$, $u_P : P_d \rightarrow \Re$, $u_R : R_d \rightarrow \Re$, $u_T : T_d \rightarrow \Re$. The utility for an outcome $\langle d, p, r, t \rangle$ is then computed as a function of the attribute utility functions:

$$u(\langle d, p, r, t \rangle) = k_D u_D(d) + k_P u_P(p) + k_R u_R(r) + k_T u_T(t) \tag{3}$$

where $k_D$, $k_P$, $k_R$ and $k_T$ are scaling constants indicating the weight of each attribute utility, and sum to 1. These scaling constants can be determined by using a standard probability-equivalence approach [11]. For example, suppose the user is given a choice between (a) an outcome with the best possible value for attribute $D$ and the worst possible value for all other attributes, and (b) a lottery in which the user would receive the best overall outcome with some probability $p$ and the worst overall outcome with probability $1-p$. The probability $p$ that causes the user to be indifferent between options (a) and (b) is then the value that should be assigned to the constant $k_D$.

After calculating all of the scaling constants, the complete utility function for an outcome $\langle d, p, r, t \rangle$ can be specified.

## 5 Results

Experiments were run to compare the accuracy of the COP-network utility computation technique with a previously developed technique for determining utilities, which we refer to as the *additive utility* technique. This technique, which is used by the "MONOLOGUE" automated negotiation system [3], handles interdependencies among attribute values that result from the specified conditional

preferences by modifying the amount of utility that each attribute value contributes in a given outcome. For example, if an attribute value $a$ is considered less desirable when attribute value $b$ is present, then $a$ contributes less utility to an outcome including $b$ than it would to an outcome not including $b$. The overall utility for an outcome is then the sum of these modified utilities.

To test the accuracy of the algorithms, a number of test cases were generated for different numbers of attributes and different numbers of conditional preferences. 10,000 trials were then run on these cases to determine how accurately the techniques could estimate a simulated user's true utilities for all outcomes, given a small number of preferences and known utilities. The winning technique was determined for each of the following criteria:

1. **Total difference winner**: For each test case, the technique that more accurately predicted utility most often out of 10,000 trials.
2. **Mean difference winner**: The technique with the lower *mean* of differences over all outcomes.
3. **Total standard error winner**: For each test case, the technique with the lower standard error most often.
4. **Mean standard error winner**: The technique with the lower *mean* standard error.

The accuracy of each technique is evaluated by considering the *total difference winner*, the *difference mean winner*, the *total standard error winner*, and the *standard error mean winner*. Table 5 shows the number of times each technique was the winner for each of these four criteria. Clearly, our COP-network is shown to more accurately predict utility regardless of the numbers of attributes and conditional preferences.

**Table 2.** Experimental results

| Technique | Total difference winner | Mean difference winner | Standard error winner | Mean standard error winner |
|---|---|---|---|---|
| COP-network | 72% | 92% | 100% | 100% |
| Additive | 28% | 4% | 0% | 0% |

## 6 Conclusions

In this paper we have shown how a general preference elicitation technique from the literature can be applied to the problem of learning and modeling user privacy preferences. Having a complete model of user preferences and utilities for various private information exchanges can help an autonomous software agent a great deal when determining how data requests should be handled on behalf of a

user. A structure known as a Conditional Outcome Preference Network (COP-network) is used to model preferences and estimate utilities for various private data collection practices. These utilities can then be used by an autonomous software agent to advise an Internet user on whether or not to interact with a website, or even to facilitate or conduct negotiations of a mutually acceptable privacy policy. Experiments show that preferences and utilities are estimated significantly better than by a previously used technique.

For further research, we plan to investigate how anonymized preference data from other users can be used to further refine utility estimates for a particular user. Clustering and other statistical and machine learning techniques should be applicable here.

## References


1. Boutilier, C., Brafman, R. I., Domshlak, C. , Hoos, H. H., and Poole, D.: CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
2. Boutilier, C., Patrascu, R., Poupart, P., and Schuurmans D.: Regret-based utility elicitation in constraint-based decision problems. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 929–934, Edinburgh, Scotland, 2005.
3. Buffett, S., Comeau, L., Fleming, M. W., and Spencer B.: MONOLOGUE: A tool for negotiating exchanges of private information in e-commerce. In *Third Annual Conference on Privacy, Security and Trust (PST05)*, pages 79–88, 2005.
4. Buffett, S., Jia, K., Liu, S., Spencer, B., and Wang, F.: Negotiating exchanges of P3P-labeled information for compensation. *Computational Intelligence*, 20(4):663–677, 2004.
5. Buffett, S., Scott, N., and Spencer B., Richter, M. M., Fleming, M. W.: Determining internet users' values for private information. In *Second Annual Conference on Privacy, Security and Trust (PST04)*, pages 79–88, 2004.
6. Chajewska, U., Koller, D., and Parr, R.: Making rational decisions using adaptive utility elicitation. In *AAAI-00*, pages 363–369, Austin, Texas, USA, 2000.
7. Chen, S.: Reasoning with conditional preferences across attributes. Master's thesis, University of New Brunswick, 2006.
8. Chen, S., Buffett, S., Fleming, M. W.: Reasoning with conditional preferences across attributes. In *Proc. of the 20th Canadian Conference on Artificial Intelligence (AI2007)*, pages 369–380, Montreal, Canada, 2007.
9. Cranor, L., Arjula, M., and Guduru, P.: 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.
10. Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J.: The Platform for Privacy Preferences (P3P) 1.0 Specification. http://www.w3.org/TR/P3P/, 16 April 2002. W3C Recommendation.
11. Goodwin, P. and Wright, G.: *Decision Analysis for Management Judgment.* John Wiley & Sons, Ltd., 2004.
12. Sandholm, T. and Boutilier, C. Preference elicitation in combinatorial auctions. *Combinatorial Auctions, Cramton, Shoham, and Steinberg, eds.*, 2006.