Multi-user Norm Consensus

Marc Serramia
King's College London
City, University of London
London, United Kingdom
marc.serramia-amoros@city.ac.uk

Natalia Criado
Universitat Politècnica de València
València, Spain
nacrpac@upv.edu.es

Michael Luck
University of Sussex
Brighton, United Kingdom
Michael.Luck@sussex.ac.uk

ABSTRACT

Many agents act in environments with multiple human users, from care robots to smart assistants. When interacting in multi-user environments it is paramount that these agents act as all users expect. However, it is not always possible to have well-defined collective preferences, nor to easily infer them from individual preferences. This is especially true in fast changing environments, like a device placed in a public space where users can enter and exit freely. In response, this paper proposes a model to represent individual preferences about the behaviour of an agent and a mechanism to find multi-user consensuses over these preferences. Norms can then be generated to ensure that when the agent follows them it will act according to the preferences of all users. We formalise what a consensus norm is and what properties the set of consensus norms should satisfy (i.e. generate the minimum number of norms while maximising the coverage of user preferences). We provide an optimisation approach to find this set of norms and show that our approach satisfies the aforementioned properties.

KEYWORDS

Norms, Consensus, Context

ACM Reference Format:
Marc Serramia, Natalia Criado, and Michael Luck. 2024. Multi-user Norm Consensus. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 9 pages.

1 INTRODUCTION

The advent of AI and IoT technologies has allowed humans to easily interact and communicate with many intelligent agents (like smart devices and services). While these technologies have undoubtedly simplified our lives, they present new challenges. Many of these agents interact with multiple human users, for example, smart assistants [17], or care robots [8]. When these agents act in multi-user environments it is paramount that they act in accordance to all users' preferences; this is especially important for environments where users may enter and exit freely. Consider the following case: a nursing home has a care robot in its common room, and the care robot can do multiple actions (e.g. talk or assist residents, interact with other devices, etc.). For this example we focus on the action of playing music. Some residents like listening to music, while others prefer to have silence to do other activities. Depending on which residents are in the room the robot should decide whether to play music or not.

Appropriateness of actions is strictly linked to their context, so that laughing at a social gathering can be appropriate at a birthday party or inappropriate at a funeral. Therefore, user preferences establish the appropriateness or inappropriateness of actions performed in a particular context. In multi-user settings an agent must guide its behaviour following consensus among all user preferences. Our vision is that when a group of users approaches an agent, the agent receives their preferences (which could, for example, be transmitted from their smartphones via Bluetooth), determines the consensus on the spot and acts accordingly. Importantly, however, contexts may not be independent; the contexts "social gathering" and "birthday" or "funeral" are not independent, with the first being a generalisation of the other two. Hence, users may define preferences at different levels of generalisation and this must be taken into account when detecting consensus.

Agents act in a multiagent system populated by other agents (e.g. a smart assistant may also interact with service providers, or other assistants), hence we resort to norms to regulate their behaviour[7]. Normative multiagent systems and methods to engineer norms have been widely studied (e.g., norm synthesis [19], norm emergence [21], norm programming [10, 25]...), norms are naturally understood by humans, and they can be simplified and generalised which is helpful to communicate system behaviour.Our aim in this paper is therefore to define a process that can detect consensus and generate norms off-line, and satisfying two desirable properties: minimality and maximal regulation. First, with a focus on human comprehension, the property of minimality becomes essential. Fitoussi et al. [12] envisioned this as minimising the set of constraints that agents ought to follow, and the reasoning necessary to process them respectively. Minimality was also addressed by [20] in relation to minimising the number of norms to regulate a particular scenario, which is especially attractive to better communicate regulations to people and for explainability. Second, since we want the system to act according to user preferences, we also seek to satisfy the property of maximal regulation with regard to user consensus. This ensures the system respects consensus preferences whenever they exist. To the best of our knowledge this has not been considered in the context of norms, in contrast to the complementary property of allowing maximal freedom to agents [24].

The paper therefore advances the state of the art by providing the following key contributions.

- A novel formalisation for the *set of consensus norms* and the properties it should satisfy.
- A novel data structure, the preference graph, to represent user preferences and to search for consensus among users.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

- A resolution of the norm consensus problem (i.e. finding the set of consensus norms) as an optimisation problem, a formal validation of the solution, and an analysis of its computational cost.

The paper is structured as follows: Section 2 motivates norm consensus in terms of privacy preferences and introduces a running example to illustrate our findings. Section 3 formally defines the norm consensus problem. In Section 4 we describe how to solve this problem, Section 5 discusses related work and in Section 6 we provide conclusions and paths to continue the research.

2 MOTIVATION: PRIVACY AND SMART ASSISTANTS

People are often worried about how their personal data is shared, used and stored by different services and devices [16]. Yet, while most people consider privacy to be important, they have different privacy preferences. Like in many other areas, in privacy, appropriateness of actions is linked to contextual integrity (i.e. the context and what that context means for those affected) [23]. While the issue of understanding individual user privacy preferences (from the perspective of contextual integrity) has recently received attention [29], to the best of our knowledge multi-user privacy consensus has not yet been addressed. We therefore motivate our work with a running example on smart assistants and privacy.

Smart assistants are usually placed in common areas, and commands are sent by voice, facilitating interaction with multiple users [17]. These devices (e.g. Alexa) only allow one user to define their preferences and are unable to distinguish different users [11] making multi-user privacy preference management already a problem. Members of a family might have different privacy preferences for their smart assistant, for example, elders may be more cautious than youngsters. These issues transcend the home assistant environment; smart assistant devices are even now being placed in meeting rooms, raising the question of how the device should respect the privacy of all participants in a group meeting. In other words, how the device should act in a way that is aligned with the consensus privacy preferences. This example shows the need to address multi-user privacy preference consensus and the complexities it can bring.

Example 1. *Imagine a company where three co-workers, Anna, Ben, and Claire, are due to meet. The company's meeting room has a smart assistant that can help them in their work, but it must also comply with their personal privacy preferences. We consider preferences with regard to sharing files and, in particular, voice recordings. An additional preference requires the co-workers be notified when sharing any type of data. We only have partial knowledge of each of the user's privacy preferences, as indicated in Table 1.*

	Files	*Voice recordings*	*Notified*
Anna	✓	×	
Ben	✓		
Claire			✓

Table 1: Preferences of the running example. ✓ represents approval, while × represents disapproval.

3 NORM CONSENSUS PROBLEM

Let Ag be a set of agents in a multi-agent system (MAS), and let A be the set of actions the agents in Ag can perform. These actions will be appropriate or inappropriate depending on the context in which they are performed. While in general we express contexts as logic formulas, we first formalise the building blocks of contexts, which we call our base contextual knowledge.

Def. 1 (Base contextual knowledge). *The base contextual knowledge is a structure $\mathcal{K}^b = \{C^b, g^b\}$, where C^b is a set of atomic propositions representing contexts and g^b is a generalisation relation, so that if $c_1, c_2 \in C^b$ and $c_1\ g^b\ c_2$, then c_1 generalises c_2. For each context $c \in C^b$ we require its negation to also be in the knowledge $\neg c \in C^b$. We also require g^b to be a transitive relation, so if $c_1, c_2, c_3 \in C^b$, then $c_1\ g^b\ c_2, c_2\ g^b\ c_3 \Rightarrow c_1\ g^b\ c_3$. We denote the set of contexts that generalise $c \in C^b$ as $g^b(c)$.*

We illustrate the base contextual knowledge in Example 2 using the contexts in the motivating example of Section 2.

Example 2. *Following the earlier example, we have $C^b = \{files, voice\_recordings, notified, \neg files, \neg voice\_recordings, \neg notified\}$. In this case, for example, $files$ generalises the $voice\_recordings$ file types.*

The base contextual knowledge only contains atomic contexts, but preferences may be defined over more complex contexts. For example, in Section 2 we considered Ben approved sharing files if notified, yet as shown in Example 2 $files$ and $notif$ are two different contexts. In order to define preferences over these more complex contexts we expand the contextual knowledge allowing compositions.

Def. 2 (Contextual knowledge). *Given a base contextual knowledge $\mathcal{K}^b = \{C^b, g^b\}$, we define the contextual knowledge $\mathcal{K} = \{C, g\}$, where C is the set of formulas of the propositional logic $\mathcal{P}$ derived from the language that has the contexts in C as terms and the operator $\wedge$[1], and g is the generalisation relation, where:*

- $\forall x, y \in C^b,\ x\ g^b\ y \Rightarrow x\ g\ y$.
- $\forall x, y \in C^b,\ x\ g\ (x \wedge y)$ *and* $y\ g\ (x \wedge y)$
- *g satisfies the transitive property, $x\ g\ y$ and $y\ g\ z \Rightarrow x\ g\ z$.*

Contextual knowledge covers all possible contexts and their relations. Of course, in real applications we might only have partial preference information over a subset of these contexts.

As mentioned above, user preferences stipulate whether or not an action is appropriate in a given context, so preferences act over pairs of action and context. We call the set containing the pairs of action and context the *domain*, but because not all actions can be performed in all contexts, we formalise this as follows.

Def. 3 (Domain). *Given a set of actions A and contextual knowledge $\mathcal{K} = \{C, g\}$, a domain is a set $\mathcal{D} \subseteq A \times C$.*

[1]We do not consider the $\vee$ operator to avoid disjunctions that can capture many independent contexts into one single logical formula. For the same reason, we do not consider the $\neg$ operator (disjunctions can be defined with $\neg$ and $\wedge$). Note though, we will use the notation $\neg c$ to refer to the negation of context c in the base contextual knowledge (see Def. 1), but this notation should not be confused with the negation operator which, for example, can be used to negate complex logic sentences.

Example 3. *Following the running example, our domain consists of the pairs of the action $A = \{share\}$ and the contextual knowledge $\mathcal{K}$ composed of the logical formulas of the contexts in Example 2.*

Consider U to be a set of users, where different users have different preferences over which actions are appropriate under which contexts. Our aim in this paper is to find the consensus among all users. We define users' preferences as the appropriateness of actions in certain contexts. Thus, for each user we consider their preferences in a *preference profile*, which we define as follows.

Def. 4 (Preference profile). *Let $\mathcal{D}$ be a domain, we define the preference profile of user $u \in U$ as a function $p_u : \mathcal{D} \to \{-1, 0, 1\}$, where for each pair $(a, c) \in \mathcal{D}$, $p_u(a, c) = 1$ means that user u thinks a is appropriate in context c, $p_u(a, c) = -1$ means the user thinks it is inappropriate, and $p_u(a, c) = 0$ means we do not know the user's preference.*

While we would ideally like complete knowledge of each user's preference profile, in real cases we cannot guarantee this (since users are not willing to be involved in defining their preferences [13]), so we do not assume complete knowledge of p_u.

Example 4. *Following the earlier example, the preference profiles for the three users would be as follows (we omit unknown preferences for clarity):*

Anna: $p_A(share, files) = 1$, $p_A(share, voice\_recording) = -1$.
Ben: $p_B(share, files) = 1$.
Claire: $p_C(share, notified) = 1$.

Since the aim of this paper is to find the point at which users reach consensus among their preferences and ensure that agents (devices, services, etc.) act with respect to this consensus, we must first define what we understand by consensus. Considering several user preference profiles, a pair $(a, c) \in \mathcal{D}$ of action and context in the domain can be one of the following.

- Consensual: All users agree that the action is appropriate (or all agree it is inappropriate) in the context.
- Non-consensual: There are some users who think the action is appropriate in the context, while others think it is inappropriate.
- Unknown: The pair (a, c) is neither consensual nor non-consensual. In other words, all users for which we know the preferences agree that (a, c) is appropriate (or all agree it is inappropriate), but we do not have all users' preferences.

Importantly, not all users might specify preferences for the same pair of action and context (a, c), yet this does not mean that a non-consensus or a consensus does not exist. First, non-consensus will also happen where there are two users who disagree on the appropriateness of an action in related contexts. For example, in a family one user thinks the smart assistant should be able to share everything while another user thinks it should not share banking details, even if the first user did not directly specify their preference for banking details, the action of sharing in the context banking details is non-consensual. If we discard a non-consensus, then a consensus can happen in three cases:

- When all users approve (or all disapprove) performing an action in different related contexts (in terms of generalisation relations);
- When all users approve (or all disapprove) performing an action in different unrelated contexts.
- A mix of both of the above two points.

In more detail, consider two users u_1 and u_2, and consider contexts c_1 and c_2, with $c_1\ g\ c_2$. On the one hand, if u_1 thinks a is appropriate in c_1 and u_2 thinks a is appropriate in c_2, we can say that their consensus is that a is appropriate in the most specific context out of the two (c_2). We call this a positive consensus. Furthermore, consider another user u_3 and a context c_3 unrelated to c_1 or c_2. If u_3 thinks a is appropriate in c_3, then the consensus between u_1 and u_3 would be to perform a when both of their approved contexts apply ($c_1 \wedge c_3$). This is because $c_1 \wedge c_3$ is generalised by both c_1 and c_3, so it becomes the context in which both users agree. Finally, these two clear-cut cases can be mixed with one another; for example, if u_1 thinks a is appropriate in c_1, u_2 thinks a is appropriate in c_2, and u_3 thinks a is appropriate in c_3, then the consensus between the three users is that a is appropriate in $c_2 \wedge c_3$. We formally categorise each pair of action and context as follows. Plus, all of these cases can be analogously defined for inappropriateness, and in that case we would call them negative consensus.

Def. 5 (Consensus types). *Given a set of users U with their corresponding preference profiles $p_u \forall u \in U$, and given $(a, c) \in \mathcal{D}$, we say there is a:*

- *Non-consensus: If $\exists u, u' \in U$ and $c', c'' \in C$, where c' generalises or is equal to c ($c' = c$ or $c'\ g\ c$) and c'' is more specific or equal to c ($c'' = c$ or $c\ g\ c''$), such that $p_u(a, c') = 1$ and $p_{u'}(a, c'') = -1$.*
- *Positive Consensus: If non-consensus does not hold and $\forall u \in U$, either $p_u(a, c) = 1$ or $\exists c'$ that generalises c ($c'\ g\ c$) and $p_u(a, c') = 1$.*
- *Negative Consensus: If non-consensus does not hold and $\forall u \in U$, either $p_u(a, c) = -1$ or $\exists c'$ that generalises c ($c'\ g\ c$) and $p_u(a, c') = -1$.*
- *Unknown consensus: If (a, c) is neither consensual nor non-consensual.*

Example 5. *Following Example 4, we can see that $(share, files)$ is non-consensual because for Anna and Ben we can take $c' = files$, $c'' = voice\_recordings$ and $p_B(share, files) = 1$, and $p_A(share, voice\_recordings) = -1$. On the other hand, $files \wedge \neg\ voice\_recordings \wedge notified$ is consensual as it is not non-consensual and is generalised by both $files$ and $notified$ for which we have approval from all users (because $p_A(share, files) = 1$, $p_B(share, files) = 1$, and $p_C(share, notified) = 1$).*

To ensure the agents respect all users preferences, we encode the reached consensus as norms, which regulate the actions that the agents can or cannot perform in each context. While several definitions of norms have been proposed in the literature (see [5] for a discussion), for simplicity and since we only consider contextual knowledge to encode them we favour a minimal definition. This formalisation considers three elements: the norm's precondition, the action being regulated, and the deontic operator stating if the action should or should not be performed when the precondition is true.

Def. 6 (Norm). *A norm is a structure $n = \langle \varphi, \theta(a) \rangle$, where $\varphi \in C$ is a context acting as the precondition, $a \in A$ is an action, and $\theta \in$*

$\{Prh, Per\}$ *is a deontic operator, where* Prh *denotes a prohibition and* Per *a permission*[2].

Example 6. *As explained in Example 5, there is a consensus among users in files other than voice recordings if they get notified. Thus, a norm regulating this pair of action and context, would be:* $\langle files \wedge \neg voice\_recordings \wedge notified, Per(share)\rangle$.

Given our definition of a norm, we can informally introduce the norm consensus problem. Considering several users U and their respective preference profiles p_u, the norm consensus problem corresponds to finding the set of norms satisfying three conditions: preference alignment, maximum regulation, and minimality. Thus, to formally define the problem we must first formally define these properties. Firstly, preference alignment ensures that any of the consensus norms regulates an action in a way all users would agree; in other words, the norm stems from a consensus. Formally:

Def. 7 (Preference aligned norms). *Given domain* $\mathcal{D}$ *with users* U *and preferences* p_u*, we say the set of norms* N *is preference aligned if* $\forall n = \langle \varphi, Per(a)\rangle \in N$ *there is a positive consensus over* (a, c) *and* $\forall n = \langle \varphi, Prh(a)\rangle \in N$ *there is a negative consensus for* (a, c).

Secondly, we want the agents to adhere to the users' preferences, hence we want to restrict the freedom of action of agents as much as possible considering the knowledge we have. Therefore, we require that there is a norm regulating any situation for which users have a consensus preference. Note this requirement is the reverse implication of preference alignment, so we can formalise it as follows.

Def. 8 (Maximally regulatory norms). *Given domain* $\mathcal{D}$ *with users* $U = \{u_1, \ldots, u_k\}$ *and their preferences* p_{u_i}*, we say the set of norms* N *is maximally regulatory if* $\forall (a, c)$ *over which there is a positive (resp. negative) consensus* $\exists n = \langle \varphi, Per(a)\rangle \in N$ *(resp.* $\exists n = \langle \varphi, Prh(a)\rangle \in N$*) such that* $\varphi \vDash c$*).*

Finally, we want to have the least amount of norms possible, making the functioning of the system easier to understand for humans and more efficient for the agents. If we consider all sets of norms that imply the same action regulations (be it permission or prohibition) under the same contexts, the minimal set of norms is the one that contains the least amount of them. Formally:

Def. 9 (Minimality). *Given a domain* $\mathcal{D}$*, and given a set of norms* N*, we say* N^m *is minimal over* N *if it is the smallest set of norms*[3] *such that* $\forall c \in C$ *that activates* $n = \langle \varphi, \theta(a)\rangle \in N$*,* $\exists n' = \langle \varphi', \theta(a)\rangle \in N^m$*, that is also activated by* c*, and vice versa.*

With these preliminary definitions we can now tackle the formalisation of the problem we aim to resolve in this paper.

Def. 10 (Consensus norms and norm consensus problem). *Let* U *be a set of users with preference profiles* p_u *over domain* $\mathcal{D}$*, the set of consensus norms is that whose norms are preference aligned, maximally regulatory, and minimal. The norm consensus problem consists in finding the set of consensus norms.*

[2]Note that we only consider permissions and not obligations as the preferences we consider only tell us if actions are appropriate, but that is not enough to discern whether they should be obliged or not.

[3]Note that if the context logic used the $\vee$ or $\neg$ operators, all consensuses could be defined in just one formula, rendering minimality useless

The norm consensus problem aims at first finding maximal regulatory norms over the actions for which we have sufficient knowledge, and second having the minimal set of these norms possible. The latter, minimality, favours norms with more general preconditions. This is because the minimal set of norms must regulate all situations that could be regulated with more norms, so each norm will have the most general precondition that can be drawn from user consensus. The former aspect implies that non-consensual pairs remain unregulated.

4 SOLVING THE NORM CONSENSUS PROBLEM

The difficulty of solving the norm consensus problem lies in the gap between user-specified preferences and what these preferences mean in logic terms. As we have seen, users may define their preferences over different contexts, yet these preferences might imply a consensus over another context. If we consider a general context and multiple more specific contexts, a user that approves the action in more specific contexts might not define these preferences directly. Instead, the user might express approval for the general context but ruling out those contexts in which they think the action is inappropriate. This preference specification is not coherent in terms of logic as approving a general context means approving any more specific related context. Hence, we want our norm consensus problem solution to be resilient to human preference specifications.

Our proposed solution for the norm consensus problem is divided into three steps: information propagation, consensus detection, and norm generation. We consider each in the following subsections.

4.1 Information propagation

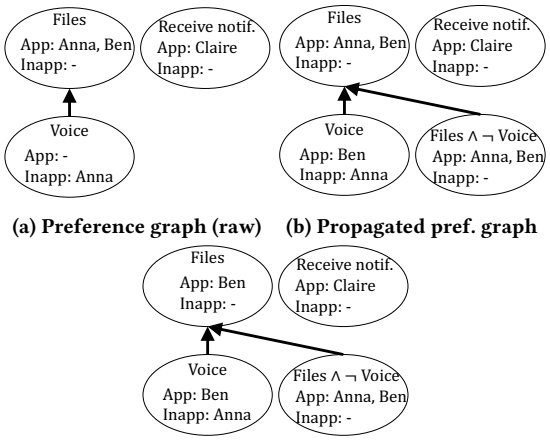
The goal of information propagation is twofold: to build a data structure for later steps from the input, and to make it logically coherent. First, we build a graph representing both user preferences and generalisation relations between contexts.

Def. 11 (Preference graph). *Consider a norm consensus problem with domain* $\mathcal{D}$*, contexts in* $\mathcal{K} = \{C, g\}$*, users* U*, and preferences* p_u $\forall u \in U$*. Given an action* $a \in A$*, we call the directed graph* $G(a) = \{No, Ed, app, inapp\}$ *a* preference graph*, where each of the nodes represents a context over which some user has defined a preference (*$No = \{c \in C, s.t.\ \exists u \in U, p_u(a, c) \neq 0\}$*), the directed edges represent the generalisation relations between the context nodes (*$Ed = \{(c, c') \in g, s.t.\ c, c' \in No\}$*), each of the nodes* $c \in No$ *has a set* $app(c)$ *with the users that find* (a, c) *appropriate (*$app(c) = \{u \in U, s.t.\ p_u(a, c) = 1\}$*) and a set* $inapp(c)$ *with the users that find* (a, c) *inappropriate (*$inapp(c) = \{u \in U, s.t.\ p_u(a, c) = -1\}$*).*

The preference graph represents the input of the problem, and contains the relevant contextual knowledge (restricted to the contexts over which we have preference information[4]) and the known preferences over these contexts.

Example 7. *Figure 1a represents the preference graph of our running example.*

[4]We make this restriction as the number of contexts in the contextual knowledge is 2^{C^b}, so in practical cases with enough base contexts it would be unfeasible to build the whole graph. Furthermore, we will not need the whole graph.



(a) Preference graph (raw) **(b) Propagated pref. graph**

(c) Coherent and propagated pref. graph

Figure 1: Preference graphs for the action share and the raw preferences for the running example (those in Table 1), each node represents a context and contains the lists of users that deem the context appropriate and inappropriate for the action, the arrows show generalisation relations (from the specific to the general context).

This graph serves to propagate knowledge (e.g. appropriateness of an action in a general context means appropriateness in a more specific context) and to make it *coherent*. We must therefore define coherence.

Def. 12 (Coherence). *A preference graph $G(a) = \{No, Ed, app, inapp\}$ is coherent if $\forall c \in No$ context node, and $\forall c' \in No$ such that $c\ g\ c'$, then $app(c) \cap inapp(c') = \emptyset$ and $inapp(c) \cap app(c') = \emptyset$.*

Example 8. *The preference graph of our running example shown in Figure 1a is not coherent; note that $files\ g\ voice\_recordings$ and $app(files) \cap inapp(voice\_recordings) = \{Anna\}$.*

As previously mentioned, human input might not be coherent (e.g. a human may find it appropriate to share information "with their family, but not with their father", but this statement is not coherent in logical terms as the father is part of the family). However, by carefully designing the propagation of preferences we can both propagate preferences and make the preference graph coherent. First, and as hinted in the previous section, preference information is propagated from general contexts to more specific ones. If a user thinks performing an action is appropriate/inappropriate in a general context, they will also agree that the action remains appropriate/inappropriate in a more specific context. Therefore, we can propagate appropriateness/inappropriateness preferences from general to specific contexts.Note though, that this propagation cannot overwrite other more specific user-specified preferences (nor those of the siblings of these contexts). Considering the preferences of our running example (see Figure 1a), when propagating the approval of sharing any files we cannot overwrite Anna's disapproval of sharing voice recordings.

It is also important to note that some contexts may be generalised by both appropriate and inappropriate contexts, in these clashing cases we leave the preference unspecified.

Given a preference graph $G(a) = \{No, Ed, app, inapp\}$, and a node $no \in No$, we note $Sib(no) = \{no' \in No, s.t. no\ g\ no'\}$ the sibling nodes of no in $G(a)$, then we formalise preference propagation as follows:

Def. 13 (Preference propagation). *Given a set of users U, and a preference graph $G(a) = \{No, Ed, app, inapp\}$, we define the preference propagation of $G(a)$ as $G^{prop}(a) = \{No^{prop}, Ed^{prop}, app^{prop}, inapp^{prop}\}$, where:*

- *$No^{prop} = No \cup No_{comp}$, where $No_{comp} = \{c \wedge \neg c_1 \wedge \cdots \wedge \neg c_k, \forall c \in No$, and $Sib(c) = \{c_1, \ldots, c_k\}\}$, are the complementary nodes.*
- *$Ed^{prop} = Ed \cup Ed_{comp}$, where $Ed_{comp} = (c \wedge \neg c_1 \wedge \cdots \wedge \neg c_k, c), \forall c \in No$, and $Sib(c) = \{c_1, \ldots, c_k\}\}$.*
- *For each user $u \in U$ and for each node $no \in No^{prop}$, if $u \in app(no)$, then $u \in app^{prop}(no)$ and $u \in app^{prop}(no')$ $\forall no' \in (Sib(no) \setminus \{no'' \in Sib(no), u \in inapp(no'')\})$ $\setminus \cup_{\{no'' \in Sib(no), u \in inapp(no'')\}} Sib(no'')$.*
- *The analogous definition for $inapp^{prop}$.*

First, for each node with siblings we add a complementary node (representing the context not covered by its other siblings). Then, the propagated appropriateness affects the original node and any of its siblings except those that the user has deemed inappropriate and the siblings of these inappropriate nodes.

Example 9. *Figure 1b shows the preference graph in Figure 1a after preference propagation.*

Second, we must ensure the preference graph is coherent. To do so, we perform preference cancellation, that is, if a general context has appropriate and inappropriate sibling contexts, then its preference must be unspecified. In our running example Anna approves sharing files but disapproves sharing voice recordings, what this means in reality is that Anna approves sharing any file except for voice recordings, after propagating appropriateness to this context (see Figure 1b) we have to cancel the preference for Files. In general we make the graph coherent as follows.

Def. 14 (Coherent and propagated preference graph). *Given a set of users U, and a propagated preference graph $G^{prop}(a) = \{No^{prop}, Ed^{prop}, app^{prop}, inapp^{prop}\}$, we define the coherent and propagated preference graph of $G^{prop}(a)$ as $G^*(a) = \{No^*, Ed^*, app^*, inapp^*\}$, where $No^* = No^{prop}$, $Ed^* = Ed^{prop}$ and for each user $u \in U$ and for each node $no \in No^{prop}$, if $u \in app^{prop}(no)$, then $u \in app^*(no)$ if $\forall no' \in Sib(no), u \notin inapp_{prop}(no')$. The analogous applies to $inapp^*$.*

In plain words, in the coherent and propagated graph, a user approves an action in a context if in the propagated graph the user approves it and does not disapprove it for any more specific context.

Example 10. *Figure 1c shows the preference graph after applying preference cancellation to that in Figure 1b.*

Although the definitions of preference propagation and cancellation are rather verbose, in practice we can easily define algorithms for both processes. On the one hand, appropriateness (inappropriateness) preference propagation can be implemented by exploring the graph in a depth first manner where we stop going deeper each time we encounter inappropriateness (appropriateness). On the

other hand, preference cancellation starts from appropriate (inappropriate) nodes and changes the preference of any inappropriate (appropriate) parent node to an unspecified preference. For brevity we do not provide the full algorithms here, interested readers can find an implementation in the code provided for the experiments in Sec. 4.4.

At this point we have obtained a graph representing all appropriateness and inappropriateness information available coherently. The next step is to detect consensus within this graph.

4.2 Consensus detection

Given an action $a \in A$ and a coherent and propagated preference graph $G^*(a) = \{No^*, Ed^*, app^*, inapp^*\}$ we are interested in finding the contexts (if any) in which all users agree the action is appropriate (or inappropriate). Once we have detected these consensual contexts, we can generate norms regulating them to solve the norm consensus problem. Since this requires norm minimality, we aim for the most general contexts with consensus. This means detecting consensus requires an intricate search in the graph. First, however, we introduce some notation.

Notation 1. *Hereafter, we call appropriate contexts those that were considered appropriate by some users and inappropriate by none, while we call inappropriate contexts those that were considered inappropriate by some users and appropriate by none.*

No^+/No^-: *The set of appropriate/inappropriate contexts before preference propagation.* $No^+ = \{c \in No, app(c) \neq \emptyset, inapp(c) = \emptyset\}$, $No^- = \{c \in C, inapp(c) \neq \emptyset, app(c) = \emptyset\}$.

No^{+*}/No^{-*}: *The set of appropriate/inappropriate contexts after preference propagation and cancellation.* $No^{+*} = \{c \in No^*, app^*(c) \neq \emptyset, inapp^*(c) = \emptyset\}$, $No^{-*} = \{c \in No^*, inapp^*(c) \neq \emptyset, app^*(c) = \emptyset\}$.

No^0: *The set of contexts that are neither appropriate nor inappropriate after propagation.* $No^0 = (No \setminus No^{+*}) \setminus No^{-*}$.

Detecting positive and negative consensus follows the same process, hence without loss of generality, in this section we will only focus on positive consensus. We are interested in finding positive consensus that maximise regulation while keeping the number of different consensus to the minimum possible. Initially, our search space is No^{+*}, comprising those contexts which, after preference propagation, are considered appropriate by some users and inappropriate by none. Thus, a positive consensus is one of these contexts or a logical formula combining several of them. For example, if there are two contexts $c_1, c_2 \in No^{+*}$, where c_1 is approved by half of the users and c_2 is approved by the other half, then $c_1 \wedge c_2$ is a positive consensus. While it is clear that all positive consensus are in No^{+*}, we can further restrict our search space. Since our aim is to find the most general contexts possible, we can exclude any context of No^{+*} that is completely generalised by other contexts in No^{+*}. Therefore we must exclude the contexts $c \in No^{+*} \setminus No^+$ for which $\forall c' \in C, c' \; g \; c$, either $c' \in No^{+*}$ or $\exists c_{mid} \in No^{+*}$, such that $c' \; g \; c_{mid} \; g \; c$. In this case, any consensus containing c' (if $c' \in No^{+*}$) or c_{mid} (otherwise) will be more general and therefore more desirable for our purposes. Importantly, we can only exclude contexts in $No^{+*} \setminus No^+$ because those in No^+ may have more preferences than the more general contexts in No^{+*}, hence excluding them could reduce the number of detected consensuses. As we will later prove in Lemma 1, all consensuses before the exclusion remain as the same or more general consensuses after the exclusions. Thus, the search space for positive consensuses is:

$$Con^+ = No^{+*} \setminus Exc \text{ where } Exc = \{c \in No^{+*} \setminus No^+, \forall c' \in C \; c' \; g \; c, \text{either } c' \in No^{+*} \text{ or } \exists c_{mid} \in No^{+*}, \text{s.t. } c' \; g \; c_{mid} \; g \; c\}. \quad (1)$$

Lemma 1. *Con^+ is the smallest search space for positive consensuses.*

Proof sketch[5]. *Given $a \in A$, we have to show that all consensus are in Con^+ and that any subset of Con^+ cannot be a valid search space. For the first part, if there is a consensus over a context $c \notin Con^+$ we divide the context into its atomic contexts joined by $\wedge$ and see that those atomic contexts that are not in Con^+ can be generalised by others in Con^+ meaning it exists $c' \in Con^+$ that generalises c. The second part can easily be proved by counterexample, providing a case in which all contexts in Con^+ are part of consensus.*

As stated in this lemma, Con^+ contains those contexts that can be building blocks for consensus contexts. These contexts might be approved by all users or might only be approved by a subset of users. Therefore, consensus will occur for sets of contexts which have joint approval of all users. The next step is to define a procedure to find all subsets of Con^+ that satisfy this condition, while also ensuring they produce maximally regulatory and minimal sets of norms. Thus, we want positive consensuses to be as general as possible, and also require that they cannot be generalised by other sets of contexts in Con^+. Additionally, positive consensus should not contain superfluous contexts (those that can be removed and the consensus remains). In summary, the positive consensuses $S \subseteq Con^+$ we aim at finding satisfy the following conditions:

- Coverage: $\cup_{c \in S} app^*(c) = U$.
- Non-redundancy: $\nexists S' \subseteq S, \cup_{c \in S'} app^*(c) = U$.
- Generality: $\nexists S'' \subseteq Con^+$ with $\cup_{c \in S''} app^*(c) = U$, such that $\forall c' \in S'' \setminus S, \exists c \in S, c' \; g \; c$.

Note that, each positive consensus in Con^+ is a set of contexts whose joint set of approval users covers U. Therefore, finding a positive consensus is equivalent to resolving the set cover problem [4] with some additional constraints. We must find the minimal combination of contexts $c \in Con^+$ whose set of approval users $\{app^*(c), c \in Con^+\}$ cover U. Each positive consensus $S \subseteq Con^+$ is a set of contexts that, combined, are approved by all users (and disapproved by none). Therefore the contexts in each S form a covering of U with regard to the users that approve each of the contexts in S or, in other words, with regard to the sets $app^*(c)$ for each $c \in S$. Furthermore, we must consider constraints to ensure the maximum generality of this set to satisfy the last property above. As we aim to find all possible consensuses we must also recurrently solve this problem (avoiding found consensus) until no new consensus exist. The set cover problem is a classic NP-hard problem whose solution and approximations have been long studied [9]. Finding all consensus is analogous to solving a set cover problem for each of the consensus. Therefore we adapt the classic binary integer program (BIP) used to solve the set cover problem for our

[5]We provide all full proofs in the supplementary material [26].

purposes. In terms of additional constraints, note that within Con^+ there might be contexts that generalise other contexts and, since we want to find the most general consensuses possible, our target function must benefit general contexts. Furthermore, to satisfy the second condition of Con^+, we also want the minimum amount of contexts. Therefore if we assign the binary variable $x_i \in \{0, 1\}$ to each context $c_i \in Con^+$, our target function is:

$$\text{Minimise} \sum_{c_i \in Con^+} (|\{c' \in Con^+, c' \; g \; c_i\}| + 1)x_i \tag{2}$$

This target function accomplishes two objectives: minimise the number of contexts selected, while selecting the most general contexts possible (in line with the above-mentioned requirements of non-redundancy and generality). Apart from this function, we must consider constraints of three kinds, namely coverage, generalisation, and found consensus constraints. First the set of $|U|$ coverage constraints ensuring selected contexts are jointly approved by all users:

$$CovConst = \{ \sum_{c_i \in Con^+ | u \in app^*(c_i)} x_i > 0, \forall u \in U\} \tag{3}$$

As mentioned earlier this constraint aims to find consensus but similar constraints can be defined for other types of agreement (for example, if we define an agreement as having at least a threshold θ number of users agreeing, we could use the constraint[6] $\sum_{u \in U} \min(\sum_{c_i \in Con^+ | u \in app^*(c_i)} x_i, 1) > \theta$. Then, the generalisation constraints ensuring any two contexts that generalise each other are not jointly selected:

$$GenConst = \{x_1 + x_2 \leq 1, \forall c_1, c_2 \in Con^+, \text{s.t. } c_1 \; g \; c_2\} \tag{4}$$

Finally, for each found consensus, we must consider a new constraint to ensure the BIP will not find the same consensus again nor any other consensus generalised by it; this will be done until the resulting BIP cannot be solved. If we have obtained a consensus of the form $con = \{c_1, \ldots, c_k\}$, then the constraint to consider is[6]:

$$FoundConConst = \{ \sum_{c_i \in con} \min(x_i + \sum_{c_j \in Con^+, \; c_i \; g \; c_j} x_j, 1) < |con| \quad \text{for each found consensus } con \subseteq Con^+\} \tag{5}$$

Note that the positive consensus contexts will be $c_1 \wedge \cdots \wedge c_k$ for each found consensus $\{c_1, \ldots, c_k\}$.

Example 11. *Following the running example, to detect the positive consensuses we use the BIP encoding consisting of the binary decision variables x_f (representing context $files$), x_{vo} (for $voice\_recordings$), x_{fnv} (for $files \wedge \neg voice\_recordings$) and x_{not} (for $notified$), and the target function:*

$$Minimise \; x_f + 2x_{vo} + 2x_{fnv} + x_{not}$$

with coverage constraints (for Anna, Ben, and Claire respectively):

$$x_{fnv} > 0, x_f + x_{fnv} > 0, x_{not} > 0$$

and generalisation constraints:

$$x_f + x_{vo} \leq 1, x_f + x_{fnv} \leq 1$$

The found consensus (the solution of the BIP) in this case is $x_f = 0$, $x_{vo} = 0$, $x_{fnv} = 1$, and $x_{not} = 1$, meaning that (as noted in previous examples) $files \wedge \neg voice\_recordings \wedge notified$ is a positive consensus for the action share. Once we have detected this consensus we add the constraint $x_{fnv} + x_{not} < 2$ and try to find other consensus solving again the BIP. Here, however, $files \wedge \neg voice\_recordings \wedge notified$ is the only positive consensus.

To finish this section we remind the reader that while in this section we focused on positive consensuses, detecting negative ones follows an analogous process[7].

4.3 Norm generation

Generating the consensus norms is straightforward; for each detected positive (resp. negative) consensus $\{c_1, \ldots, c_k\}$, we generate the norm $\langle c_1 \wedge \cdots \wedge c_k, Per(a)\rangle$ (resp. $\langle c_1 \wedge \cdots \wedge c_k, Prh(a)\rangle$). Importantly, we must prove that the resulting set of norms N is the solution of the norm consensus problem; that is, showing N satisfies the preference representation, maximally regulatory, and minimality properties.

Theorem 1. *N is the solution of the norm consensus problem.*

Proof sketch[5]. *To prove this theorem we have to show that N satisfies preference representation, is maximally regulatory and minimal. First, preference representation is naturally satisfied because our norms always stem from detected consensus. Second, we assume N is not maximally regulatory and there is a N_{max} which is. Then, there is a context $c \in C$ which activates one norm in N_{max} but not in N. By the definition, the context and action of this norm must form a consensus, but somehow it has not been found by our BIP approach. If it is not a solution of some BIP, it must not be optimal which means there is a more general consensus (contradiction), or it does not fulfil a constraint, which in all cases leads to the same contradiction. Finally, for minimality, assume there is an N_{min} that satisfies the two previous properties but has less norms than N. Assume a regulated context c and φ the precondition of a norm in N regulating c, we divide the proof in two cases. If φ is the most specific logic formula that is consensual we can follow a similar reasoning to that of maximal regulation to arrive at contradictions. If φ is not the most specific logic formula and a norm n in N_{min} has a precondition with this more specific formula. We look at the other norm n' in N_{min} regulating the contexts that are in φ but not in the more specific formula, we can show that then either n is redundant (contradiction) or using a similar reasoning as for maximality that the precondition of n' is not consensual.*

Example 12. *Following our running example and considering the positive consensus found over the action share in context $files \wedge \neg voice\_recordings \wedge notified$ (see Example 11), here we generate the norm $\langle files \wedge \neg voice\_recordings \wedge notified, Per(share)\rangle$.*

4.4 Discussion on performance

Detecting all positive and negative consensuses is equivalent to solving several set cover problems, which are NP-hard. Therefore it is important to evaluate the computational feasibility of our approach. We have generated synthetic preference graphs and solved them calculating the time it took[8]. Our experiments considered

[6]For brevity we write these constraints in non-linear form, but note that they can be linearised (or rewritten) in many ways (see [3, 6]).

[7]The formal process and proofs can be derived simply changing No^+, No^{+*}, Con^+ app, and app^* for No^-, No^{-*}, Con^- $inapp$, and $inapp^*$

[8]Code available at [26], we solved the BIPs using CPLEX [14] on a standard 2018 13" MacBook Pro (Intel Core i5-8259U processor, 8Gb of RAM, running Mac OS 12.6.)

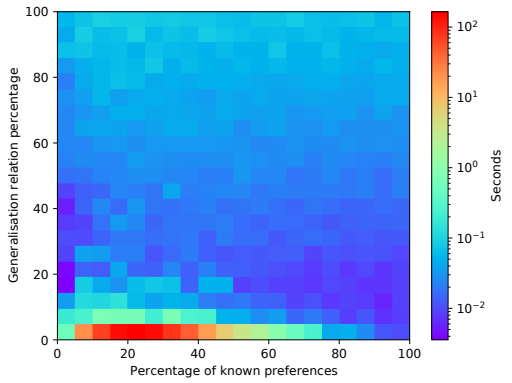


Figure 2: Average time in seconds for each problem configuration. Note the logarithmic scale for better analysis.

several variables: number of nodes, number of users, generalisation relation percentage (out of all possible), and percentage of known preferences (out of all possible). In particular, we considered a fixed number of 100 nodes (this number represents more contexts than those considered in recent studies on norms for smart personal assistants [1]) and 5 users (representative of a family or a meeting). With these settings, we tested ranges of generalisation relation densities between 0 to 100%[9] in steps of 5. The ranges of preference probabilities are between 5 and 100% in steps of 5. For each configuration we synthesised and solved 10 graphs, giving a total of 4200 preference graphs (over which we detected a total number of 179296 consensuses). Figure 2 shows the average solving times for each of these configurations.

Most problems, except those without generalisation relations, are solved almost instantly. First, when there are close to no generalisation relations, the search space is larger (more nodes), and we have no constraints apart from coverage. Therefore the problem is equivalent to that of finding all subsets of the search space only requiring the coverage constraint. Conversely, problems with generalisation relations not only have generalisation constraints, but their found consensus constraints are stricter. Second, when the known preference percentage is close to 20% the number of consensus (and solving time) peaks; fewer preferences limit the size of the search space, while more preferences lead to discrepancies on the appropriateness of nodes, thus making more nodes ineligible.

To conclude, we see in our experiments that computation times are mostly a consequence of the number of consensus, not the difficulty of finding each of them individually. On average, finding one of the 179296 consensuses we detected (solving its associated BIP) took 0.03757 seconds (with a standard deviation of 0.04566, and a maximum of 0.8804 seconds). We argue that in real cases it is unreasonable that a domain can lead to this number of consensuses (e.g. the individual problem that took the longest to solve had 7612 consensuses). Plus, we have not considered domain knowledge, with it we could add constraints to avoid finding consensus containing contexts that cannot happen jointly (e.g. $daytime \wedge nightime$).

[9]This percentage is over the maximum number of relations a preference graph (a DAG) can have (which is $\frac{|nodes|\cdot(|nodes|-1)}{2}$).

5 RELATED WORK

This paper is related to multi-party privacy conflicts, which have been studied for a long time. Some works in this area include: Such et al. [30] who propose a computational mechanism to resolve these conflicts in online social networks; Ulsoy et al. [31] who consider agents that bid in multi-user auctions to publish content online about multiple users; and Mosca et al. [22] who consider ELVIRA agents that help address multi-user privacy conflicts once detected. However, these works aim at addressing (possible) multi-user privacy conflicts and the agents considered represent individual users. In contrast, in our work, we aim at finding consensus among all users before any conflict arises and to avoid them arising. Apart from this, we are not only focused on privacy preferences. Our problem is also close to the area of value alignment. While there are normative solutions for alignment [18, 27, 28], these focus on a set of values instead of preferences. Values are more abstract and less specific than preferences, and different users may understand values differently (as is usually the case with privacy), therefore we argue that defining preferences over actions and contexts directly will produce more accurate norms. Continuing in the area of norms, Kafali et al. [15] propose a normative solution for privacy, but again we aim at considering any type of preference over the behaviour of agents. Also, Ajmeri et al. [2] consider agents reasoning about multiple values and norms to make decisions on the actions to perform, but this is focused on action decision-making, while we want to establish consensus norms before any action is performed.

6 CONCLUSIONS

In this paper we have shown how to detect user consensus over preferences on how an agent should act and generate norms to regulate them. This not only represents a novel approach to privacy for smart assistants as we motivated, but also to resolve similar issues in multi-agent systems in general. Importantly, our aim has been to provide a well-founded model that can serve as a basis to be used in more complex applications with only slight variations. For example, while the approach as we have presented it requires consensus to cover all users, this might be too strong a requirement for some real world problems. In scenarios with many users, reaching a consensus is more difficult than in cases with few users. Fortunately, this and similar requirements can be easily relaxed by slightly changing the constraints (in this case the coverage constraints) of our BIP encoding. As for future work, while we have aimed to find all user consensuses to then have maximally regulatory norms, this can still be improved if we know of more preferences. An interesting topic for future research is to smartly acquire new knowledge to maximise consensus (and their derived norms).

ACKNOWLEDGMENTS

Research funded by project SAIS Secure AI AssistantS via Grant EP/T026723/1, funded by the UK Engineering and Physical Sciences Research Council; and by project TED2021-131295B-C32, funded by MCIN/AEI/ 10.13039/501100011033 and the European Union NextGenerationEU/PRTR.

REFERENCES

[1] Noura Abdi, Xiao Zhan, Kopo M. Ramokapane, and Jose Such. 2021. Privacy Norms for Smart Home Personal Assistants. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 558, 14 pages. https://doi.org/10.1145/3411764.3445122

[2] Nirav Ajmeri, Hui Guo, Pradeep K. Murukannaiah, and Munindar P. Singh. 2020. Elessar: Ethics in Norm-Aware Agents. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 16–24.

[3] Mohammad Asghari, Amir M. Fathollahi-Fard, S. M. J. Mirzapour Al-e hashem, and Maxim A. Dulebenets. 2022. Transformation and Linearization Techniques in Optimization: A State-of-the-Art Survey. *Mathematics* 10, 2 (2022), 26. https://doi.org/10.3390/math10020283

[4] Egon Balas and Manfred W. Padberg. 1972. On the Set-Covering Problem. *Operations Research* 20, 6 (1972), 1152–1161. https://doi.org/10.1287/opre.20.6.1152 arXiv:https://doi.org/10.1287/opre.20.6.1152

[5] Tina Balke, Célia da Costa Pereira, Frank Dignum, Emiliano Lorini, Antonino Rotolo, Wamberto Vasconcelos, and Serena Villata. 2013. Norms in MAS: Definitions and Related Concepts. In *Normative Multi-Agent Systems*, Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W. N. van der Torre (Eds.). Dagstuhl Follow-Ups, Vol. 4. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 1–31. https://doi.org/10.4230/DFU.Vol4.12111.1

[6] Johannes Bisschop. 2006. *AIMMS optimization modeling*. Paragon Decision Technology B.V., Haarlem, The Netherlands.

[7] Guido Boella and Leendert van der Torre. 2004. Regulative and Constitutive Norms in Normative Multiagent Systems. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning* (Whistler, British Columbia, Canada) *(KR'04)*. AAAI Press, Washington DC, 255–265.

[8] Robert Booth. 2020. Robots to be used in UK care homes to help reduce loneliness. https://www.theguardian.com/society/2020/sep/07/robots-used-uk-care-homes-help-reduce-loneliness Accessed Jul. 2023.

[9] Alberto Caprara, Paolo Toth, and Matteo Fischetti. 2000. Algorithms for the set covering problem. *Annals of Operations Research* 98, 1-4 (2000), 353–371.

[10] Daniela Dybalova, Bas Testerink, Mehdi Dastani, and Brian Logan. 2013. A framework for programming norm-aware multi-agent systems. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*. Springer, Cham, 364–380.

[11] Jide S. Edu, Jose M. Such, and Guillermo Suarez-Tangil. 2020. Smart Home Personal Assistants: A Security and Privacy Review. *ACM Comput. Surv.* 53, 6, Article 116 (dec 2020), 36 pages. https://doi.org/10.1145/3412383

[12] David Fitoussi and Moshe Tennenholtz. 2000. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence* 119, 1-2 (2000), 61–101.

[13] Christian Hoffmann, Christoph Lutz, and Giulia Ranzini. 2016. Privacy cynicism: A new approach to the privacy paradox. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace* 10 (12 2016). https://doi.org/10.5817/CP2016-4-7

[14] IBM. 1988. CPLEX. https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer. Accessed 06/2021.

[15] Özgür Kafalý, Nirav Ajmeri, and Munindar P. Singh. 2016. Revani: Revising and Verifying Normative Specifications for Privacy. *IEEE Intelligent Systems* 31, 5 (2016), 8–15. https://doi.org/10.1109/MIS.2016.89

[16] Mary Madden. 2014. Public perceptions of privacy and security in the post-Snowden era. https://www.pewresearch.org/internet/2014/11/12/public-privacy-perceptions/ Accessed on April 2022.

[17] Nicole Meng-Schneider, Rabia Yasa Kostas, Kami Vaniea, and Maria K Wolters. 2023. Multi-User Smart Speakers - A Narrative Review of Concerns and Problematic Interactions. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 213, 7 pages. https://doi.org/10.1145/3544549.3585689

[18] Nieves Montes and Carles Sierra. 2022. Synthesis and Properties of Optimally Value-Aligned Normative Systems. *J. Artif. Int. Res.* 74 (sep 2022), 36. https://doi.org/10.1613/jair.1.13487

[19] Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. 2013. Automated Synthesis of Normative Systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* (St. Paul, MN, USA) *(AAMAS '13)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 483–490.

[20] Javier Morales, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. 2014. Minimality and Simplicity in the On-line Automated Synthesis of Normative Systems. In *AAMAS 2014* (Paris, France). IFAAMAS, Richland, SC, 109–116.

[21] Andreasa Morris-Martin, Marina De Vos, and Julian Padget. 2019. Norm emergence in multiagent systems: a viewpoint paper. *Autonomous Agents and Multi-Agent Systems* 33 (2019), 706–749.

[22] Francesca Mosca and Jose Such. 2022. An explainable assistant for multiuser privacy. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 10. https://doi.org/10.1007/s10458-021-09543-5

[23] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Washington Law Review* 79 (2004), 119.

[24] Thomas Ågotnes, Wiebe Van Der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael Wooldridge. 2007. On the Logic of Normative Systems. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (Hyderabad, India) *(IJCAI'07)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1175–1180.

[25] Murat Sensoy, Timothy J Norman, Wamberto W Vasconcelos, and Katia Sycara. 2012. OWL-POLAR: A framework for semantic policy representation and reasoning. *Journal of Web Semantics* 12 (2012), 148–160.

[26] Marc Serramia, Natalia Criado, and Michael Luck. 2024. Code and Supplementary material. https://github.com/marcserr/NormConsensus.

[27] Marc Serramia, Maite Lopez-Sanchez, Stefano Moretti, and Juan A. Rodriguez-Aguilar. 2023. Building rankings encompassing multiple criteria to support qualitative decision-making. *Information Sciences* 631 (2023), 288–304. https://doi.org/10.1016/j.ins.2023.02.063

[28] Marc Serramia, Manel Rodriguez-Soto, Maite Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Filippo Bistaffa, Paula Boddington, Michael Wooldridge, and Carlos Ansotegui. 2023. Encoding Ethics to Compute Value-Aligned Norms. *Minds and Machines* (2023). https://doi.org/10.1007/s11023-023-09649-7

[29] Marc Serramia, William Seymour, Natalia Criado, and Michael Luck. 2023. Predicting Privacy Preferences for Smart Devices as Norms. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2262–2270.

[30] Jose M. Such and Natalia Criado Pacheco. 2016. Resolving multi-party privacy conflicts in social media. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2 June 2016), 1851–1863. https://doi.org/10.1109/TKDE.2016.2539165 ©2016 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

[31] Onuralp Ulusoy and Pinar Yolum. 2021. PANOLA: A Personal Assistant for Supporting Users in Preserving Privacy. *ACM Trans. Internet Technol.* 22, 1, Article 27 (sep 2021), 32 pages. https://doi.org/10.1145/3471187