Modeling of Personalized Privacy Disclosure Behavior: A Formal Method Approach

A K M Nuhil Mehdy
Computer Science Department, Boise State University
Boise, Idaho, USA
akmnuhilmehdy@u.boisestate.edu

Hoda Mehrpouyan
Computer Science Department, Boise State University
Boise, Idaho, USA
hodamehrpouyan@boisestate.edu

ABSTRACT

In order to create user-centric and personalized privacy management tools, the underlying models must account for individual users' privacy expectations, preferences, and their ability to control their information sharing activities. Existing studies of users' privacy behavior modeling attempt to frame the problem from a request's perspective, which lack the crucial involvement of the information owner, resulting in limited or no control of policy management. Moreover, very few of them take into the consideration the aspect of correctness, explainability, usability, and acceptance of the methodologies for each user of the system. In this paper, we present a methodology to formally model, validate, and verify personalized privacy disclosure behavior based on the analysis of the user's situational decision-making process. We use a model checking tool named UPPAAL to represent users' self-reported privacy disclosure behavior by an extended form of finite state automata (FSA), and perform reachability analysis for the verification of privacy properties through computation tree logic (CTL) formulas. We also describe the practical use cases of the methodology depicting the potential of formal technique towards the design and development of user-centric behavioral modeling. This paper, through extensive amounts of experimental outcomes, contributes several insights to the area of formal methods and user-tailored privacy behavior modeling.

CCS CONCEPTS

• **Security and privacy → Formal methods and theory of security**.

KEYWORDS

behavioral analysis, user behavior modeling, privacy, security, formal methods

ACM Reference Format:
A K M Nuhil Mehdy and Hoda Mehrpouyan. 2021. Modeling of Personalized Privacy Disclosure Behavior: A Formal Method Approach. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3465481.3470102

ARES 2021, August 17–20, 2021, Vienna, Austria

ACM ISBN 978-1-4503-9051-4/21/08…$15.00
https://doi.org/10.1145/3465481.3470102

1 INTRODUCTION

Privacy in the information domain refers to the right of a person to monitor and control the processing, exposition, and preservation of information about themselves. [34]. Accordingly, the responsibility is on the user themselves to take control of what kind of information should be shared with whom, when, and how [24, 40, 41, 51]. However, for an individual, it is quite cumbersome and difficult to manage and control their information sharing preferences [52]. This is because different devices, applications, and software require different privacy configurations from the users, and most importantly they are not designed to be personalized or assisting. Therefore, it is important than ever before to develop and provide suitable tools and algorithms to the users so that they can define, manage, and make the best use of their privacy preferences with ease. Existing methodologies and protocols intend to tackle this problem by employing technique such as access control policies [44, 46], machine readable privacy policy languages [5, 16], formal methods [6, 11], machine learning [15, 39, 48], etc. However, most of the works attempt to frame the problem from a request's perspective which lack the crucial involvement of the information owner, resulting in limited or no control of policy adjustment. Moreover, a very few of them take into consideration the aspect of personalization and explainability of such tools. Most importantly, while there is a significant amount of research aimed at design and development of privacy management tools and techniques, 'their practical usability and acceptance remains an important challenge' [28].

Therefore, this paper applies model-based analysis to personalize privacy behavior which answers two key research question: how to model privacy behaviour and how to use this privacy behavior model for analysis. We decomposed this problem into three subcategories: (I) Identification of relevant privacy behavior and situational factors, (II) applying proper modeling techniques, (III) validating the models.

As part of model-based approach, we focuse on formal methods that are concerned with modeling, specifying, and verifying any systems using mathematical techniques otherwise known as model checking [13]. A system could be physical or conceptual comprised of interconnected components such as processes, states, nodes, etc. Model checking is an automated approach to verify that a model of a system, usually a finite-state machine, satisfies a set of desired properties (i.e., requirement specifications) written in a temporal logic [20]. This is achieved by exhaustively searching a system's state space in order to determine if these criteria hold. If there is a violation, an error trace is produced (i.e., a counterexample). Model checkers take system description (i.e., formal model) and a set of requirements as input and reason whither the requirements are satisfied or not. In privacy literature, human decision making, in

other words, individual's intention to disclose private information is also considered as process which involves different components, otherwise known as influential factors [3]. When the number of factors is large, doing manual specification and testing of the privacy policies is difficult. It is also possible that subtle conditions get unnoticed. Again, a way to tackle this problem to a certain extent, is the use of mathematically-based techniques. Hence, we adopt the analogy of finite state machines from the theory of computation and aim to model human privacy disclosure behavior based on this specific formalism technique.

That being said, to learn user's privacy behavior towards the development of user-specific models, it is important to investigate the factors and parameters that influence users to make dynamic privacy decisions [3, 30, 35, 45]. The decision to exchange private information, as well as the risk perceptions that drive this decision, differs from situation to situation. Various considerations, such as the type of information, the receiver of the information, and the source of confidence underlying the reason for sharing, all play a role in the decision making process [23, 47]. Moreover, risk assessment, potential risks consideration, and alternate exploration are all part of the process of deciding what to do in a specific situation [2, 50]. Additionally, individual variations in demographics, personality traits, and decision-making styles as well as their effect on users' privacy-related habits must be studied before developing any behavioral model. Therefore, we work on a dataset from [38] which was obtained by conducting a custom designed survey on Amazon Mechanical Turk [1] (N=401) based on the theory of planned behavior (TPB) to measure the way users' perceptions of privacy factors and intent to disclose information are affected by three situational factors embodied hypothetical scenarios: information type, recipients' role, and trust source.

In this work, we chose to focus on user's situational decision-making process and represent our approach to formally model, validate, and verify personalized privacy behavior. We represent a scaled down version of our proposed methodology where we model each individual's privacy disclosure behavior where their disclosure decision merely rely on three factors— information type, recipients' role, and trust source. Even though human decisions depend on many more factors, we chose this level of abstraction because the dataset in hand capture the users' privacy behavior based on these three factors. On the other hand, we wanted to evaluate our approach on top of ground truth dataset. Nevertheless, the methodology presented in this paper depicts potential of formalism towards the development of privacy management tools. This paper is the first to our knowledge to leverage extended version of automata based transitioned systems towards modeling individual's privacy behavior. This work provides insight into:

- Model-based analysis of personalized privacy behavior
- Formulate personalized privacy policies
- Detect and reason about unwanted disclosure behavior
- Validate the proposed model-based approach and demonstrate its practicality

[1] A crowd sourcing website for businesses and researchers to hire remotely located "crowdworkers" to perform on-demand tasks such as survey, data labeling, etc.

2 LEARNING PRIVACY PREFERENCE

In this work, we represent and evaluate our formal method approach to model users' privacy disclosure behavior based on a dataset that we obtained through a survey. We captured users' situational privacy decisions, through a custom scenario-based survey with 401 participants, each responding to a subset of 48 total unique scenarios. Every data point is referred to the responses to a series of questionnaires that assess participants' attitudes toward each situation, as well as their expectations of and willingness to reveal personal information in the given situation. By manipulating three situational factors: information type, recipient's role, and trust source, we use path analysis to model participants' privacy perceptions and plans, taking into account their assessments on subjective norm, perceived behavioral control, and attitude. This choice of factors is partly inspired from the theory of contextual integrity (CI) [8, 42]. The findings show how users make privacy decisions in a variety of contexts, as well as how situational factors influence users' views of privacy factors and their willingness to share private information. Most importantly, the results also reveal how every individual has their own preferences and concerns about disclosing their private information in certain situations. Therefore, this dataset best suit our personalized behavioral modeling experiment. The following sections describe the survey strategy and the data set in more detail.

2.1 Survey

After agreeing to participate in the survey, a person is given a series of eight hypothetical scenarios and asked to answer to them one by one. Each scenario places the subject in a position where he or she must choose whether or not to reveal the information embodied in that scenario. This includes the situational factors on which participants can place a high degree of confidence in their interpretation and decision on whether or not to disclose. We manipulate three situational factors to see how they affect participant responses:

Information Type (IT) The type of the information that is illustrated in the scenario. Each scenario is about one of three information types: health, finance, or relationship.

Recipient's Role (RR) The type of the recipient, based on the relationship to the survey participant, to whom the information may be disclosed. We take into account four such recipient roles: family, friend, colleague, and online service (e.g., facebook, twitter, discussion forum, etc).

Trust Source (TS) From whom the participant got the motivation of disclosing the information to the recipient. We consider four trust sources: family, friend, expert (e.g., physician, counselor, financial adviser, etc), and self (i.e., searching the internet).

Different combinations of these factors yield a total of 48 (3*4*4) unique scenarios. For each of the combination, we prepare a scenario where a trust source encourages the participant to share the information with a recipient. We made every scenario as similar as possible to minimize extraneous variability while incorporating the factors in a natural and coherent manner in the hypothetical scenario. In other words, we made sure the framing of the scenarios does not become significantly different from each other so that only

the factors get changed, and a proper parametric analysis is justified. An example scenario with *health* as information type, *friend* as trust source, and *family member* as recipient's role could be:

> Your doctor called and told you that your lab results came back positive for a disease. One of your friends suggested discussing the situation with a family member and asking their support, saying it could be helpful.

Another unique scenario could be generated by changing the trust source from friend to family and recipient's role from family to online:

> Your doctor called and told you that your lab results came back positive for a disease. A family member suggested asking other patients and doctors on an online discussion forum, saying they have found it helpful for dealing with their similar condition.

Every participant is assigned a set of 8 random scenarios with associated questionnaires. A participant has to read a given scenario and respond to all of the corresponding questions before proceeding to the next assigned scenario. We used rejection sampling to ensure that each user's 8 scenarios covered all 11 distinct factor levels at least once, ensuring a minimal degree of heterogeneity between their circumstances and, as a result, responses. To minimize order effects, we also randomly order the set of 8 scenarios for each participant. In the end, the individual completes a brief survey in which we intend to capture their general privacy attitudes regardless of any specific situation. This move is intended to capture expectations that are believed to be constant over time and do not alter in response to changing circumstances. Participants are asked to optionally enter their ethnicity, age group, country of origin, and period of residence in that country in the final phase of the survey for accumulating demographic information.

There are two sets of questions in the survey: i) scenario-specific questions (12 total) and ii) general attitude questions (4 total). For each of the eight scenarios allocated to each person, the first set of 12 questions is repeated. At the end of the survey, the second set of questions is presented. The scenario-specific questionnaire is inspired by Heirman et al. [22], and the second set of questions is inspired by prominent privacy research [1, 12]. Appendix A shows a screenshot of the survey system representing 1 of 8 random scenarios given to a participant, and appendix B shows the screenshot representing the general attitude questions given to a participant at the end of the survey. Before the main survey, we conducted a pilot test with six of our research lab's colleagues. Their feedback was instrumental in resolving problems with the survey interface, user experience, and clarity of the scenarios and questionnaires. Later we used Amazon Mechanical Turk, an online crowd-sourcing marketplace, to find participants for the final survey. We looked for workers from the United States who are at least 18 years old with at least 95% HIT (Human Intelligence Task) acceptance rate[2] and 50 hits approved.

[2]whose previous works got approved by 95% of the requesters.

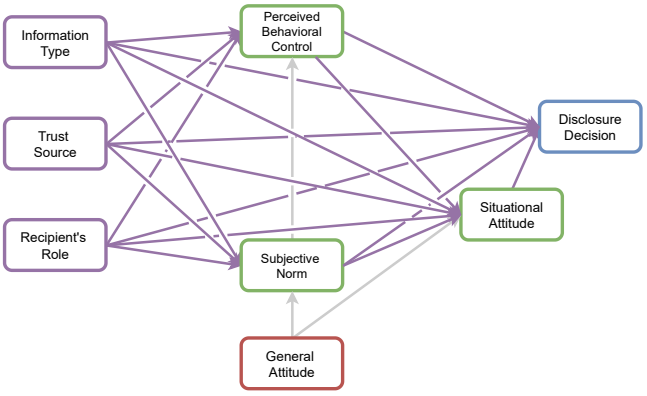


Figure 1: The Path Model for Analyzing Users' Privacy Decision-making Process.

2.2 Dataset

We employed a number of filters to ensure the quality of the data. First, we capture the time a participant spent on each scenario step and removed the data points from our analysis if the spent time was too low. Second, we randomly placed attention search questions in between survey questions. We also restricted repeated submissions from same participant by setting a browser cookie for 3 days after a satisfactory submission. The answers to the questions were translated into a numeric format (1 to 5) from the 5-point scale (ranging from Strongly Disagree to Strongly Agree). For the final decision question, we represent the Share and Not-Share options in logical numeric form, 1 and 0. In the end, we get 3208 data points, grouped by 401 participants, containing their information disclosure decisions based on different situational factors.

2.3 Path Model for Privacy Behavior Analysis

In one of our earlier works [38], we leveraged the data to measure users' behavioral intention and their situational perception of three constructs: attitude, subjective norm, and perceived behavioral control. These constructs and the path model is inspired from the theory of planned behavior (TPB) [3]. We also, incorporated the scenario factors— information type, recipient's role, and trust source in our path analysis to measure the correlation of these factors with the information disclosure decision of the user. Figure 1 depicts the path model. The analysis results show that the path model fits the data very well with $\chi^2_{11} = 12.017$, $p = 0.3623$, $CFI = 1.0$, $TLI = 0.99$, $SRMR = 0.008$, $RMSEA = 0.005$, 90% $CI = 0.000$ to 0.020. Also, the comparative fit index (CFI) and Tucker-Lewis index (TLI) values which ranges from 0 to 1 show near-perfect scores.

Among all the path analysis results published in our work, one of the findings shows that there exist significant (indirect) effects of the scenario factors on the users' disclosure decisions. In Figure 1, they refer to the paths from the purple leftmost boxes to the blue rightmost box via the mediator green boxes in between. These total effects describe *how* users' intention changes from one scenario to another; the mediating TBP factors provide an explanation for *why*. A few important findings include but not limited to— with regard to the recipient's role in the scenario, compared to the recipient "online service", the odds of disclosure were estimated to be 16.6% higher when the recipient was a family member and 12.9% higher when

the recipient was a friend; with regard to the type of information, compared to relationship information, the odds of disclosure were estimated to be 3.1% lower when the scenario involved financial information and recipient was a family member and 5.1% higher when the scenario involved health information, etc. These results indeed proof the influence of the situational factors towards users' disclosure decision and therefore act as the basic components of our formal privacy behavioral model.

3 FORMAL MODELING

This section describes the approach of developing the formal model of an user's privacy disclosure behavior by taking into account the privacy decisions made by that user. Our approach aims to address the issue of formally modeling the privacy behavior of an user which could be eventually utilized to develop a personalized privacy management system. The whole approach is divided into four main stages: i) observing user's historical sharing activity, ii) modeling users' personalized privacy behavior, iii) validating the model, iv) verifying the model given the privacy properties of the user. We have already detailed about the survey and the dataset in the earlier sections which refer to the first stage.

3.1 Model Assumptions

In this work, one of the main assumptions of the users' disclosure behavior is that user decides to share/not-share a specific type of information with a certain type of recipient(s) after being advised by a specific trust source. We represent these knowledge and the decision made by the user in the form of state model. Transitions between states occur with respect to a specific information type, trust source, and recipient's role. We also assume that there is no other factors/components involved in the user's decision making process. Additionally, we assume that user's behavior could conceivably be modeled as a finite state machine. This research utilizes finite state automata (FSA) extended with data variables to model the privacy disclosure behavior of the users.

3.2 Model Paradigm

FSA as a chosen formalism allows for a design and development of a well-structured tools to conduct an automated analysis during the early stages of studying user's privacy behaviour. Accordingly, there are various tools for designing and verifying such FSA based formal models, i.e. NuSMV, PVS, Z3, and UPPAAL are a few of examples. We choose UPPAAL because of its ability to support model checking over network of automata using temporal logic [9]. UPPAAL also supports formalism through parallel compositionality among the automata. This modeling paradigm helps us to retrieve the traces of the transition while checking for a given query. Therefore, this modeling paradigm enables us to execute the requirements as temporal logic queries which in turns exhaustively check the satisfaction of the privacy properties.On the other hand, counter examples are provided to reason about privacy properties that are violated.

4 MODELING IN UPPAAL

The reasons for selecting UPPAAL is because UPPAAL provides a better graphical user-interface that allows for the development,

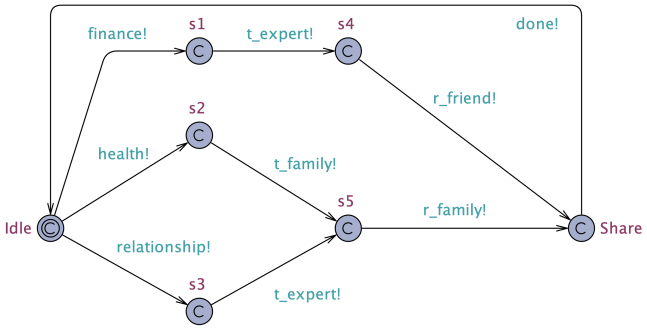


Figure 2: The Behavioral Model of User 89 Created in UPPAAL

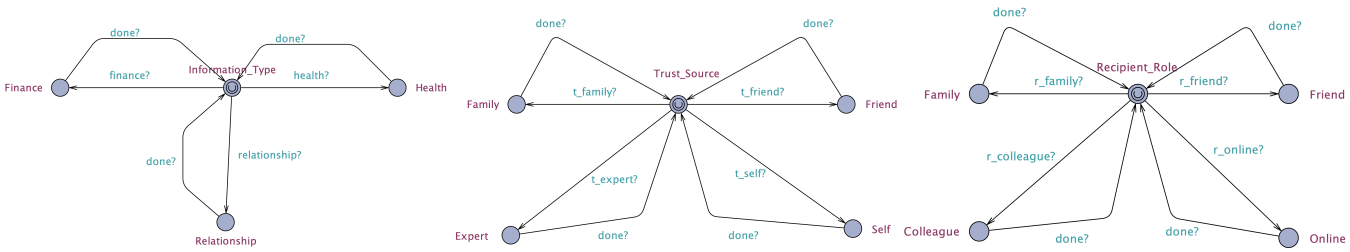
modification, validation, and verification of any system model with drag and drop interface[29]. In UPPAAL, a system is made up of several concurrent processes, each of which is modeled as an automaton. Each automaton has a set of locations otherwise known as states. Transitions between these states could be managed by guard and synchronization. A guard imposes conditions on variables and clocks ensuring when the transition is enabled. Synchronization in UPPAAL enables two or more processes to communicate with each other based on a hand-shaking synchronization. Two actions are possible while a transition happens— assignment of variables or reset of clocks. UPPAAL further extends timed automata with other types of data variables such as integer and Boolean towards developing a modeling language which is as close as a high level programming language [29].

4.1 Behavioral Analysis and Personalization

To model the privacy disclosure behavior of a specific user, we collect the user's responses to the survey questionnaire and observe the information sharing behavior in different scenarios. For this, we randomly pick a user, for example, number 89 in our tabular dataset. Table 1 contains the 8 random scenarios which were assigned to this user. Table 1 represents that the user agreed to share information in 3 out of 8 given situations (scenario 1, 2, and 7). Based on that, we model the privacy behavior by composing them into a data dependent transition graph (Figure 2). This graph contains a set of states and synchronization operations. When a transition happens from one state to another, a message is emitted to one or more observer processes through the synchronization channel. For example, when a transition happens from the *Idle* state to state *s1*, it emits a message titled *finance* to any listening processes. This is one of the many useful features of UPPAAL which allows to design network of FSMs (i.e., parallel composition). The start and end states are marked as "committed states", which means there would be immediate transitions from these two states as soon as the transitions are enabled. In UPPAAL, the committed states take prompt transitions when the simulation or exhaustive search happens. This feature allows us to simulate the transitions spontaneously without waiting for any external inputs. It is worth mentioning that, we only model the positive sharing behavior of each user. In other words, figure 2 only contains a composition of 3 different scenarios where this user agreed to share the information with the recipients. Hence, if an information sharing attempt, described as a query, fails

Table 1: Disclosure Decisions by the User 89 Captured by the Survey

No	Scenario	IT	TS	RR	Share?
1	You recently had a very bad argument with your partner. Your counsellor suggested sharing and discussing this matter with a family member, saying they could support you.	Relationship	Expert	Family	Yes
2	Your doctor called and told you that your lab results came back positive for a disease. A family member suggested discussing the situation with a family member and asking their support, saying it could be helpful.	Health	Family	Family	Yes
3	Your doctor called and told you that your lab results came back positive for a disease. A family member suggested asking other patients and doctors on an online discussion forum, saying they have found it helpful for dealing with their similar condition.	Health	Family	Online	No
4	You recently had a very bad argument with your partner. One of your friends suggested asking on an online discussion forum they use to get support from others, saying they have found it helpful for dealing with their situation.	Relationship	Friend	Online	No
5	Your doctor called and told you that your lab results came back positive for a disease. You did some research and found that people often find it helpful to get support from a colleague.	Health	Self	Colleague	No
6	You received a notice from a collection agency saying you have a debt which needs immediate attention. A family member suggested asking on an online discussion forum they use to get support from others, saying they have found it helpful for managing a similar situation.	Finance	Family	Online	No
7	You received a notice from a collection agency saying you have a debt which needs immediate attention. Your financial advisor suggested discussing the situation with a friend and asking their support, saying it could be helpful.	Finance	Expert	Friend	Yes
8	You recently had a very bad argument with your partner. A family member suggested sharing and discussing this matter with a colleague, saying they could support you.	Relationship	Family	Colleague	No



(a) Information Type Observer **(b) Trust Source Observer** **(c) Recipient Role Observer**

Figure 3: Observer Models Created in UPPAAL.

to comply with the model in figure 2, then the model checker tells that the corresponding query was not satisfied and also shows a counter-example trace (if available).

4.2 Observer Models

An observer is an add-on automaton which without perturbing the observed system can detect events. We use 3 such models along with the user's behavioral model (Figure 2) to keep track of the transitions and associated factors. This eventually help to prepare

and employ descriptive queries for the verification of the model. Figure 3 depicts those 3 separate observer models. Figure 3 (a) represents the observer which keeps track of the information types. It listens for the messages— *finance*, *health*, and *relationship* whenever a transition in the behavioral model emits one of these values. For example, if a transition happens from *Idle* state to the *s1* state in the behavioral model (Figure 2) then this observer model transitions from the *Information_Type* state to the *Finance* state. The activities of the other two observer models (Figure 3 (b) and (c)) are similar. Model 3 (b) listens for the messages *t_family*, *t_friend*, *t_expert*, and *t_self* to keep track of the trust source. Likewise, model 3 (c) listens for the messages *r_family*, *r_friend*, *r_colleague*, and *r_online* to keep track of the recipient's role. All the observer models return to their initial state once they get a specific message - *done* from the behavioral model.

4.3 Behavior as Systems

The user-specific behavior model along with the observer models create the network automata otherwise known as a concurrent system in UPPAAL. This type of composition is also known as parallel composition of processes made of automaton. In our setup, the user model synchronizes data between itself and the observer models by leveraging the channel features in UPPAAL. The formal definition of the system model could be defines as follows:

$$User||Information\_Type||Trust\_Source||Recipient\_Role$$

4.4 Validation

UPPAAL uses graphical simulation as the model validation strategy [9]. Therefore, we conduct a simulation step to validate our models by running the system automatically which ensure that the models behave as intended, without any unexpected crash or deadlock. By utilizing the simulation feature of UPPAAL, we manually conduct some transitions in the behavioral model, and also utilize the random simulation feature to make sure the transitions are taken as expected. Figure 4 shows the UPPAAL simulation control panel where, the button *Reset* and *Next* are used to manually perform some transition operations, and the button *Random* is used to start an automatic simulation that can run indefinitely. The simulation also allows us to make sure that the concurrency operation between the behavioral and the observer processes is taking place without any system breakdown.

5 VERIFICATION WITH MODEL CHECKING

In this section, the verification of the user's privacy disclosure model is explained. Figure 5 depicts a high level abstraction of the model checking process. In this approach, a set of desired properties (i.e., specifications) are checked against a model of a system[10, 18].

5.1 Specification Language in UPPAAL

The set of privacy properties (i.e., requirement specifications), which we expect the formal model to verify, are formulated based on the conducted survey in section 2.1. The specification languages that could be used to express these types of privacy properties are Linear Temporal Logic (LTL), Computation Tree Logic (CTL), and Timed

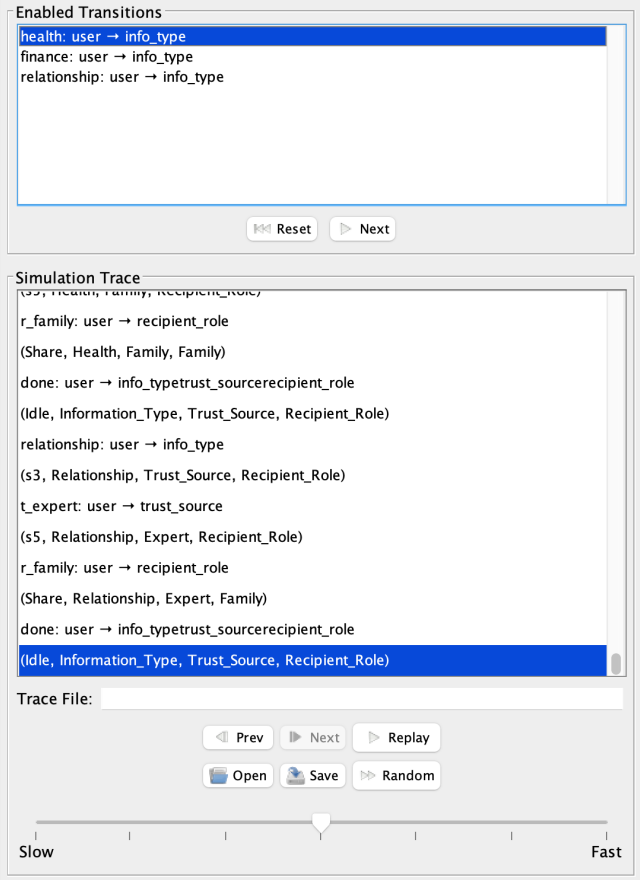


Figure 4: Part of the Simulation Window Containing the Control Buttons for Automatic and Manual Transition.

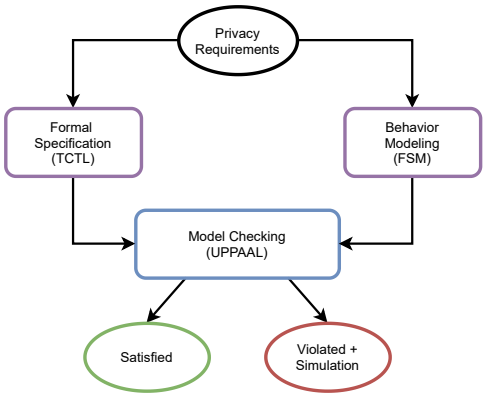


Figure 5: Model Checking Approach.

Computation Tree Logic (TCTL)[7]. In UPPAAL, the process of verification operates with a simplified version of TCTL which is a subset of CTL. In TCTL, temporal connectiveness are expressed as pairs of symbols where the first element represents one of the path quantifiers and the second element represents one of the state

Table 2: Requirement Specifications or Privacy Properties of User 89

No	Privacy Property
1	E<> (user.share and information_type.Health and trust_source.Family and recipient_role.Family)
2	E<> (user.share and information_type.Relationship and trust_source.Expert and recipient_role.Family)
3	E<> (user.share and information_type.Finance and trust_source.Expert and recipient_role.Friend)

quantifiers. Likewise, UPAAL query language consists of path formulae and state formulae [9]. The path formulae quantify over paths (traces) of the model whereas state formulae describe individual states. In UPPAAL, these quantifiers are expressed as follows:
E = exists a path (E in UPPAAL),
A = for all paths (A in UPPAAL),
F = some state in a path (<> in UPPAAL),
G = all states in a path ([] in UPPAAL),
Example queries could be written as $A[]p$, $A <> p$, $E <> p$, $E[]p$, and $p \rightarrow q$ where p and q are local properties. In other words, the query $E <> p$ tells that, 'it is possible to reach a state in which p is satisfied' or 'p is true in at least one reachable state. $E <> Process.End$ is the UPPAAL notation for the same temporal logic formula $\exists\Diamond Process.End$ and is understood as 'it is possible to reach the location *End* in automaton *Process*'.

5.2 Personalized Privacy Properties

In order to formulate the privacy properties of user 89, we translate the user's disclosure decisions that are represented in Table 1 in to the following statements: 'if the information type is *health* and the trust source is a *family* member and the recipient of the information is also a *family* member, then the user *share* the information. Similar to this specific criteria, every user has their own requirements when they agree to share the private information based on the situational factors. For each user, we translate their own privacy disclosure criteria into UPPAAL specification formulas. These formulas are then checked against his/her behavioral model to ensure the correctness of it. Since we use observer models (Figure 3) along with the behavioral model (Figure 2) to create a concurrent system model, the observers have their own formal specification. In Table 3, we represent the equivalent expressions of the scenario factors in UPPAAL's specification language, while Figure 3 visualizes the state transition graphs of those factors. Thus, the privacy disclosure properties for user 89 is represented in Table 2 that is a the transformation of his/her responses based on the scenarios 1,2, and 3 from Table 1. Therefore, property number 1 from Table 2 expresses: there exist a path, eventually where the properties enclosed in the parenthesis is true.

5.3 Reachability Analysis

There are three types of properties which are commonly checked against a formal model— safety, liveness, and reachability properties. Reachability properties are used in state-transition systems which helps to examine the type and number of states that can be accessed through a particular system model [26]. It is the simplest form of properties which determines whether a given state formula, Φ, possibly could be satisfied by any reachable state. In this work, we verify whether or not the user-specific privacy properties holds in any, some, or all state of that user's privacy behavior model. We prefer reachability analysis over other similar methods (e.g., graph matching approach) because it allows us to search all potential paths in which the properties may or may not be satisfied, in a thorough and automated manner. Using UPPAAL, we applied reachability analysis to check which privacy properties were satisfied and which were not. UPPAAL performs the reachability analysis using either Breadth-First-Search or Depth-First-Search for checking whether a state is reachable or not. We preferred BFS of DFS to verify our reachability properties because it is a complete algorithm, ends within a finite time, and consider fewest edges while searching. The results of this procedure allows us to examine an user's privacy disclosure behavior, and whether or not a new sharing attempt complies with her existing privacy policies.

Table 3: Scenario Factors' Properties

No	Knowledge Base Property
1	$E <> (information\_type.Health)$
2	$E <> (information\_type.Finance)$
3	$E <> (information\_type.Relationship)$
4	$E <> (trust\_source.Family)$
5	$E <> (trust\_source.Friend)$
6	$E <> (trust\_source.Expert)$
7	$E <> (trust\_source.Self\_Search)$
8	$E <> (recipient\_role.Family)$
9	$E <> (recipient\_role.Friend)$
10	$E <> (recipient\_role.Colleague)$
12	$E <> (recipient\_role.Online\_Service)$

Table 4 contains a few verification queries that we check against the privacy disclosure model of user 89. Query 1 indeed gets satisfied since there is a valid transition in the FSM model (Idle -> s2 -> s5 -> Share) as well as in it's CTL version which is verified by the TCTL formula. Query 2 does not get satisfied since this user had no history of sharing his *Health* information to either *Friend* or *Online* even when the trust source was *Family*. Query 3 does not get satisfied because there is indeed one path where the property is true, (in Figure 2, Idle -> s1 -> s4 -> Share). We can even see the diagnostic trace when this query is executed (Figure 6). Additionally, we can verify that the model will not face any deadlock in it's lifespan by executing queries like #4.

6 DIFFERENT USE CASES

In this section, we represent the privacy disclosure model of a different user. A user is selected from the dataset randomly and holds the ID 242. In this case, in order to demonstrate the potential of the proposed behavioral model approach to include complex privacy properties with additional constraints, we imposed limitations on the days of the week or the number of times specific information could be disclosed. For this user, the responses that was received to

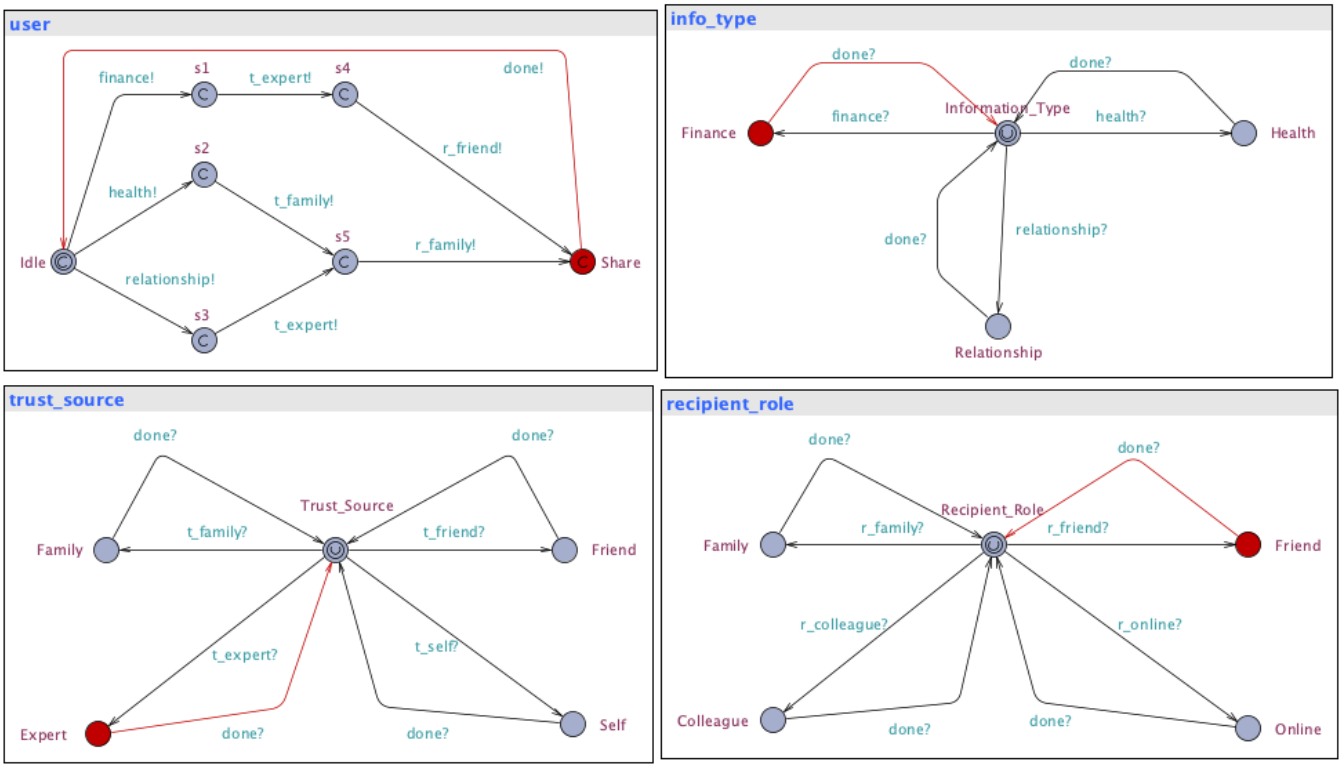


Figure 6: Diagnostic Trace of Query 3.

Table 4: Example of Some Queries to the User 89's Model and the Verification Results

No	To Verify	UPPAAL Query	Verification
1	There exists a path where eventually the user is in *Share* state and the information type was *Health*, trust source was *Family*, and recipient's role was *Family*	E<>(user.Share and info_type.Health and trust_source.Family and recipient_role.Family)	Satisfied
2	There exists a path where eventually the user is in *Share* state and the information type was *Health*, trust source was *Family*, and recipient's role was either *Friend* or *Online*	E<>(user.Share and info_type.Health and trust_source.Family and (recipient_role.Friend or recipient_role.Online))	Not Satisfied
3	For all paths, it should never be the case that the user is in *Share* state and the information type was *Finance*, trust source was *Expert*, and recipient's role was *Friend*	A[] not (user.Share and info_type.Finance and trust_source.Expert and recipient_role.Friend)	Not Satisfied
4	There should not be any states without successors	E<> not deadlock	Satisfied

the randomly assigned scenarios, we observed that this user agreed to share the information in 6 out of 8 situations. Therefore, we model his/her disclosure behavior in terms of a transition system (i.e., finite state machine) which is depicted in Figure 7. As mentioned already, we added two guard conditions on two edges of the FSM: I) the day of the week for information sharing has to be between Monday to Friday (encoded as 1-5) to make the path- *Idle -> Expert -> Family -> Share* enabled, II) *Health* type information could be shared *Online* no more than twice through the path *Idle -> Health -> Family -> Online*. However, while verifying the model, we find a deadlock by querying *E<> not deadlock* to the model checker. This property does not get satisfied depicting that there is indeed a deadlock. This

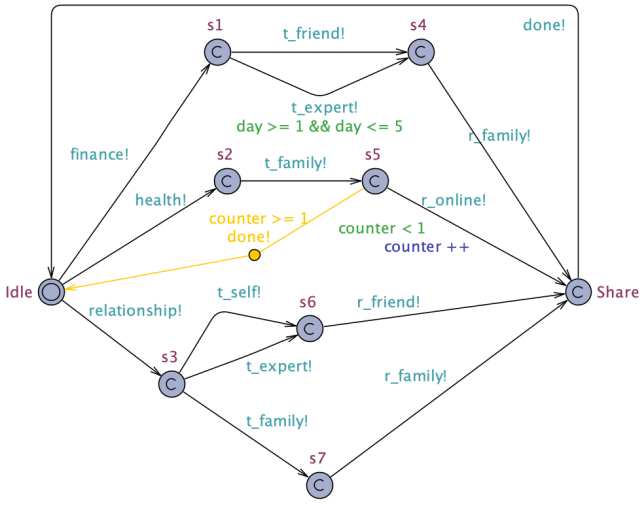


Figure 7: The Model of User 242 Created in UPPAAL.

happens because of the *counter* guard which was imposed on the path, *Idle -> Health -> Family -> Online*. Since a query is checked exhaustively, by running/simulating the model for hundreds of iterations, UPPAAL reaches to this deadlock state after simulating through this path for twice. In other words, the *counter* become 1 and the path become disabled from the state *s5* to *Share*. However, we then resolve that deadlock in the model by creating a path from *s5* to *Idle* (colored in yellow). Thus, whenever the model checker tries to go through this path for more than twice and faces a guard in state *s5*, it can then safely get back to the initial state without blocking the simulation operations.

In some other cases, incorporating additional decision-making factors or adding subcategories to the existing ones may result in a more complex network of automata with added granularity. For example, the information type *health* could have two sub categories: mental health and physical health. A user might want to share *physical health* condition with *family* but *mental health* condition to both *family* and *friends*. Representing this sort of scenario is also quite feasible in our proposed technique.

6.1 Syntax and Semantics of the Models

Each of the models in a system consists of a set of control nodes otherwise known as states. In addition to these control states, a composed model uses integer variables, simple channels, and broadcast channels. The edges of the automata contain two types of labels: guards and synchronization. The guards express the conditions on the values of the integer variables. These conditions need to be satisfied in order for the edges to be taken for transitions. In our models (e.g., Figure 7), we add guards on transitions to ensure the traversal of the paths that represent the desired information sharing activity of the user. We also add synchronization variables in the models which enable the communication between the behavioral model and the observer models. In Figure 2 and Figure 7, all the variables marked with an exclamation character "!" represent message transmission. Similarly, the observer models contain synchronization variables marked with a question mark "?" (Figure 3) that represent message reception. For example, whenever a transition happens

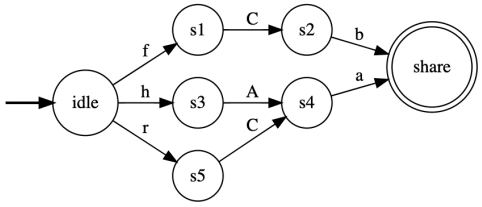


Figure 8: The DFA of the User 89.

from the state *Idle* to the state *s1*, on the behavioral model (Figure 2), it transmits a message *finance* which is then received by the "Information Type" observer model through the *finance?* path. The simple channels (e.g, finance!, finance? t_family!, r_friend!) help the observer models to keep track of the scenario factors and the broadcasting channel (i.e., *done!*) helps the observers to get back to the start position once an iteration (i.e., sharing activity) is completed. The model also consists of urgent (as soon as transition is enabled, current state will change to the next state) and committed locations. Since the information sharing behavior is assumed to be an non time-dependent process, we make the states urgent so that the transitions happen as soon as the flags are available. Its worth mentioning that, in time-dependent systems, the use of urgent locations reduces the complexity of the analysis by reducing the number of clocks.

7 APPLICATION AND USABILITY

In this section, we briefly describe the application and usability of our proposed methodology. In the following section, we describe a process of creating the baseline model from user's historical sharing activities.

7.1 Automatic Translation of Activities to DFA

Formal modeling, formal verification and validation approaches are mostly used in the area of physical systems (e.g., industrial control systems, and cyber-physical systems). In this context, the process is mostly completed by human domain experts. Once a model is developed, validated, and verified; it is then used as the foundation of any downstream tasks such as hardware assembly, resiliency test, etc. In contrast, in the user-specific behavioral study, the formal modeling part has to be a automatic process that translated user's desired privacy properties into formal specifications. This is because the end users of an application will not have sufficient expertise to create the mathematically based model which is a core requirement to the model checking technique. Therefore, we utilize an existing tool [43] to automatically translate an user's historical sharing activities, manifested as regular expression, into deterministic finite automata (DFA).

First, we generate a regular expression string for every shared activity. We get three such strings— *rCa*, *hAa*, and *fCb* from the sharing activity (i.e., survey response) of the user 89 (Table 1). Where, h,f, and r represent the information types- *Health*, *Finance*, *Relationship* respectively. A,B, C, and D represent the trust source— *Family*, *Friend*, *Expert*, and *SelfSearch* respectively. a,b, c, and d represent the recipient's role— *Family*, *Friend*, *Colleague*, and *OnlineService* respectively. Then we combine the strings together

with the regular expression's choice character + to get

$$(rC + hA)a + fCb$$

Finally, we use the tool [43] to generate a minimized DFA that accepts the regular expression (i.e., model the shared activities in terms of finite automata). Below is the formal Definition of the DFA:

$$\text{Set of state, } Q = \{idle, s_1, s_2, s_3, s_4, s_5, Share\}$$
$$\text{Alphabet, } \Sigma = \{h, f, r, A, B, C, D, a, b, c, d\}$$
$$\text{Initial state, } q_o = idle$$
$$\text{Set of final states, } F = \{share\}$$
$$\text{Transition function, } \delta = Q \times \Sigma \rightarrow Q$$

Figure 8 is the result of the translation which acts as the foundation of the UPPAAL model depicted in Figure 2. This preliminary automation step could be later taken over by another downstream automation tool (discussed later) to eventually develop a UPPAAL acceptable formal model.

7.2 Standalone Privacy Management Tool

Normally, the verification engine of UPPAAL is by default executed on the same computer as the user interface, but it can also run on a more powerful server which allows to host a complex behavioral model. Another supporting utility named *verifyta* is able to accept *.ta*, *.xta*, and *.xml* files as an input and use high-level programming language (e.g., Java)) API to perform the modeling, simulation, and model checking through a pragmatically native environment. This API makes it possible to interpret user's historical sharing activities and then develop a UPPAAL compatible formal model. The API could additionally be utilized to validate the developing model and verify it against a set of queries.

7.3 In Software Design and Development

One of the many advantages of formal modeling is its ability to allow for an early assessment of the model [4, 54]. In other words, it is possible to design, validate, and exhaustively verify user's privacy behavior model and think of it as the algorithm of his allowed behavior. Later on, programmers can leverage this model as the template for coding a function (e.g., shouldShare()) for that user in their software system. This process will enable the programmers to write a function which is already exhaustively tested, and therefore, no need to conduct typical unit testing on the program. An existing software can also integrate the privacy management tool by interacting with the verification engine through high level API. Thus the software can achieve a proper privacy management component inside it's ecosystem.

7.4 User-Interface (UI) of the Privacy Settings

The user-interface of any software, mobile-app, or web-app plays an important role in providing its users with a more flexible privacy settings. Users of the communication platforms are found to be less careful about properly setting their privacy preferences offered by the apps [32, 33, 37]. This is because of the generic and 'one fits all' nature of the privacy preference pages. Therefore, user-specific formal modeling can help with the UI/UX designers and programmers are better equipped to provide personalized privacy settings pages to their users by utilizing their underlying behavioral model.

8 RELATED WORK

Researchers from the field of privacy, decision making, and personalization have shed light on the area of behavior modeling. They have been exploring, how the psychological factors of humans relate to their concerns about their information privacy [1, 32, 51]. Accordingly, many behavioral theories have been established and adopted to the privacy management domain [3, 8, 21]. Theory of planned behavior (TPB) tells that people's behavior is directly determined by their behavioral *intentions*. These intentions are in turn influenced by their *attitude* (positive or negative evaluation of the decision), perception of the *subjective norms* (generally expected behavior in their social group), and *perceived behavioral control* (ease or difficulty to perform the behavior). The theory also states that these constructs together determine an individual's behavioral intentions and provide a model to capture humans' decision making behavior. Therefore, researchers from various areas (e.g., privacy, use of the internet, health, environmental psychology, etc) have used TPB and demonstrated its effectiveness in predicting human behavior in terms of privacy decision making[14, 22, 36, 49, 53].

Another privacy management theory which is relevant to our work is known as the theory of contextual integrity (CI) [8]. In the CI theory, privacy is formulated as an appropriate flow of information that conforms with the contextual informational norms (i.e., rules governing flow of information in CI format). An example of a norm in the context of *health* could be: a husband usually shares his diagnosis result with his family doctor, or his wife but not with his friends or in the social media. In this example, the husband is recognized as the data subject and the sender, the doctor or wife as the recipient of the information, health as the information type, and the recipient will hold the information confidentially as the transmission principle. Based on the theory of (CI) [8], privacy is violated if the information is shared or transferred with friends or financial advisers, as they are not usually and explicitly included as part of the 'allowed' recipients of the information.

Consequently, many researchers have studied modeling users' privacy decision-making process in the context of various types and recipients of the information. Knijnenburg et. al. discovered about how the type of the information and their recipients have significant effect on user's information disclosing tendency [25]. In their study participants were asked to set their privacy settings on a custom made privacy settings UI of an imagined Facebook-like social network site by indicating which of their profile information they would share with whom. In another study [17], authors have examined the idea of users' privacy calculus (i.e., costs vs benefits) and how it led the users to disclose their different types of private information to different types of recipients (websites), in a purpose-specific fashion. Lederer et. al. [31] investigated the relative effects of different recipients and the situations towards users' information disclosure intention. By surveying 130 participants, given two hypothetical situations, they found that situation is an important determinant and highly correlated with the information recipient.

Despite the existence of many behavioral theories and analysis, only a handful of works address the issue of personalized modeling

of human behavior. Most importantly, a few of them acknowledge the issue of practical usability and application of the derived models. Joshaghani et. al. extends the concept of CI theory and provide mathematical models that enables the creations and management of privacy norms by the individual users [24]. They propose and develop a custom formal verification technique which ensure privacy norms are enforced for every information sharing attempt by the user. Similar to our transition system based formalism, Lu et. al. proposed a technique that translates the privacy specification or requirements of web services to LTL formulas [34]. Then the create the privacy policy model by utilizing a privacy interface automata (PIA) that transforms the messaging structure extracted from the web service business process execution language into an automaton. Krishnan et al. propose a semi-formal approach that enforce privacy requirements by leveraging the role-based access control technique along with LTL formulas [27]. Grace et al. propose a technique for modeling user-centric privacy management using labeled transition systems. The goal of this model is to compare the user's privacy preferences with the privacy policies of the cloud service provider [19]. Thus the users 'can be informed of the privacy implications of the services' and warned of potential privacy breaches. However, they mentioned about two limitations— i) requirement of human intervention for creating initial model, ii) limited extensibility and scalability.

In our work, we address many of the above-mentioned limitations and open questions by representing personalized situational behavior, proposing a technique for automatic translation of activities to FSM, demonstrating the practical usability, and describing the scalability of this formal approach.

9 CONCLUSION

Users' ability to better manage their data-sharing practices is limited due to the lack of suitable user-centric privacy management tools and techniques. Moreover, very few of the existing methodologies take into consideration the aspect of personalization, correctness, and explainability. Most importantly, their practical usability and acceptance remain a significant challenge. In this paper we have presented an approach to formally model, validate, and verify personalized privacy disclosure behavior based on the analysis of user's situational decision-making process. The proposed methodology demonstrates a privacy formalism and verification technique based on UPPAAL which is a tool for modeling, validation, and verification of automata based systems. Most importantly, the methodology depicts the potential of formalism towards the development of user-centric privacy management tools. In future work, we plan to extend the user's privacy behavior model to incorporate additional decision making factors towards more granularity. We also plan to develop an end-to-end framework on top of UPPAAL to fully automate the process of transforming the historical sharing activities into UPPAAL compatible network of automata.

ACKNOWLEDGMENTS

The authors would like to thank National Science Foundation for its support through the Computer and Information Science and Engineering (CISE) program and Research Initiation Initiative(CRII) grant number 1657774 of the Secure and Trustworthy Cyberspace (SaTC) program: A System for Privacy Management in Ubiquitous Environments.

REFERENCES

[1] Mark S Ackerman, Lorrie Faith Cranor, and Joseph Reagle. 1999. Privacy in e-commerce: examining user scenarios and privacy preferences. In *Proceedings of the 1st ACM conference on Electronic commerce*. 1–8.
[2] Walid A Afifi and Laura K Guerrero. 2000. Motivations underlying topic avoidance in close relationships. *Balancing the secrets of private disclosures* (2000), 165–180.
[3] Icek Ajzen et al. 1991. The theory of planned behavior. *Organizational behavior and human decision processes* 50, 2 (1991), 179–211.
[4] Vangalur S Alagar and Kasilingam Periyasamy. 2011. *Specification of software systems*. Springer Science & Business Media.
[5] Paul Ashley, Satoshi Hada, Günter Karjoth, Calvin Powers, and Matthias Schunter. 2003. Enterprise privacy authorization language (EPAL). *IBM Research* 30 (2003), 31.
[6] Guillaume Aucher, Guido Boella, and Leendert Van Der Torre. 2011. A dynamic logic for privacy compliance. *Artificial Intelligence and Law* 19, 2-3 (2011), 187.
[7] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. MIT press.
[8] Adam Barth, Anupam Datta, John C Mitchell, and Helen Nissenbaum. 2006. Privacy and contextual integrity: Framework and applications. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE, 15–pp.
[9] Gerd Behrmann, Alexandre David, and Kim G Larsen. 2006. A tutorial on Uppaal 4.0. *Department of computer science, Aalborg university* (2006).
[10] Matthew L Bolton, Noelia Jiménez, Marinus M van Paassen, and Maite Trujillo. 2014. Automatically generating specification properties from task models for the formal verification of human–automation interaction. *IEEE Transactions on Human-Machine Systems* 44, 5 (2014), 561–575.
[11] Travis D Breaux, Hanan Hibshi, and Ashwini Rao. 2014. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requirements Engineering* 19, 3 (2014), 281–307.
[12] Tom Buchanan, Carina Paine, Adam N Joinson, and Ulf-Dietrich Reips. 2007. Development of measures of online privacy concern and protection for use on the Internet. *Journal of the Association for Information Science and Technology* 58, 2 (2007), 157–165.
[13] Edmund M Clarke and Jeannette M Wing. 1996. Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)* 28, 4 (1996), 626–643.
[14] Mark Conner, Sara FL Kirk, Janet E Cade, and Jennifer H Barrett. 2003. Environmental influences: factors influencing a woman's decision to use dietary supplements. *The Journal of nutrition* 133, 6 (2003), 1978S–1982S.
[15] Elisa Costante, Yuanhao Sun, Milan Petković, and Jerry Den Hartog. 2012. A machine learning solution to assess privacy policy completeness: (short paper). In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*. 91–96.
[16] Lorrie Cranor. 2002. *Web privacy with P3P*. " O'Reilly Media, Inc.".
[17] Tamara Dinev and Paul Hart. 2006. An extended privacy calculus model for e-commerce transactions. *Information systems research* 17, 1 (2006), 61–80.
[18] G Eleftherakis and P Kefalas. 2001. Towards model checking of finite state machines extended with memory through refinement. *Advances in signal processing and computer technologies* (2001), 321–326.
[19] Paul Grace and Mike Surridge. 2017. Towards a model of user-centered privacy preservation. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*. 1–8.
[20] Orna Grumberg, Doron A Peled, and EM Clarke. 1999. Model checking.
[21] Jerold L Hale, Brian J Householder, and Kathryn L Greene. 2002. The theory of reasoned action. *The persuasion handbook: Developments in theory and practice* 14 (2002), 259–286.
[22] Wannes Heirman, Michel Walrave, and Koen Ponnet. 2013. Predicting adolescents' disclosure of personal information in exchange for commercial incentives: An application of an extended theory of planned behavior. *Cyberpsychology, Behavior, and Social Networking* 16, 2 (2013), 81–87.
[23] Leslie K John, Alessandro Acquisti, and George Loewenstein. 2011. Strangers on a plane: Context-dependent willingness to divulge sensitive information. *Journal of consumer research* 37, 5 (2011), 858–873.
[24] Rezvan Joshaghani, Stacy Black, Elena Sherman, and Hoda Mehrpouyan. 2019. Formal specification and verification of user-centric privacy policies for ubiquitous systems. In *Proceedings of the 23rd International Database Applications & Engineering Symposium*. 1–10.
[25] Bart Piet Knijnenburg and Alfred Kobsa. 2014. Increasing sharing tendency without reducing satisfaction: finding the best privacy-settings user interface for social networks. (2014).
[26] Soonho Kong, Sicun Gao, Wei Chen, and Edmund Clarke. 2015. dReach: δ-reachability analysis for hybrid systems. In *International Conference on TOOLS and Algorithms for the Construction and Analysis of Systems*. Springer, 200–205.
[27] Padmanabhan Krishnan and Kostyantyn Vorobyov. 2013. Enforcement of privacy requirements. In *IFIP International Information Security Conference*. Springer,

272–285.
[28] O Rivera Kurkovsky, Oscar Rivera, and Jay Bhalodi. 2007. Classification of privacy management techniques in pervasive computing. *International Journal of u-and e-Service, Science and Technology* 11, 1 (2007), 55–71.
[29] Kim G Larsen, Paul Pettersson, and Wang Yi. 1997. UPPAAL in a nutshell. *International journal on software tools for technology transfer* 1, 1-2 (1997), 134–152.
[30] Robert S Laufer and Maxine Wolfe. 1977. Privacy as a concept and a social issue: A multidimensional developmental theory. *Journal of social Issues* 33, 3 (1977), 22–42.
[31] Scott Lederer, Jennifer Mankoff, and Anind K Dey. 2003. Who wants to know what when? privacy preference determinants in ubiquitous computing. In *CHI'03 extended abstracts on Human factors in computing systems*. 724–725.
[32] Heather Richter Lipford, Andrew Besmer, and Jason Watson. 2008. Understanding Privacy Settings in Facebook with an Audience View. *UPSEC* 8 (2008), 1–8.
[33] Yabing Liu, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2011. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. 61–70.
[34] Jiajun Lu, Zhiqiu Huang, and Changbo Ke. 2014. Verification of Behavior-aware Privacy Requirements in Web Services Composition. *JSW* 9, 4 (2014), 944–951.
[35] May O Lwin and Jerome D Williams. 2003. A model integrating the multidimensional developmental theory of privacy and theory of planned behavior to examine fabrication of information online. *Marketing Letters* 14, 4 (2003), 257–272.
[36] Octav-Ionuț Macovei. 2015. Determinants of consumers' pro-environmental behavior–toward an integrated model. *Journal of Danubian Studies and Research* 5, 2 (2015).
[37] Alessandra Mazzia, Kristen LeFevre, and Eytan Adar. 2012. The pviz comprehension tool for social network privacy settings. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*. 1–12.
[38] AK Mehdy, Michael D Ekstrand, Bart P Knijnenburg, and Hoda Mehrpouyan. 2021. Privacy as a Planned Behavior: Effects of Situational Factors on Privacy Perceptions and Plans. *UMAP'21, June 21–25, 2021, Utrecht, Netherlands© 2021 Association for Computing Machinery.* (2021).
[39] AKM Nuhil Mehdy and Hoda Mehrpouyan. 2020. A User-Centric and Sentiment Aware Privacy-Disclosure Detection Framework based on Multi-input Neural Network.. In *PrivateNLP@ WSDM*. 21–26.
[40] Nuhil Mehdy, Casey Kennington, and Hoda Mehrpouyan. 2019. Privacy Disclosures Detection in Natural-Language Text Through Linguistically-motivated Artificial Neural Network. In *2nd EAI International Conference on Security and Privacy in New Computing Environments*. EAI.
[41] Hoda Mehrpouyan, Ion Madrazo Azpiazu, and Maria Soledad Pera. 2017. Measuring personality for automatic elicitation of privacy preferences. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 84–95.
[42] Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.* 79 (2004), 119.
[43] Noam. 2015. Noam is a JavaScript library for working with automata and formal grammars for regular and context-free languages. https://github.com/izuzak/noam. [Online; accessed 10-May-2021].
[44] Sylvia Osborn, Ravi Sandhu, and Qamar Munawer. 2000. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC)* 3, 2 (2000), 85–106.
[45] Sandra Petronio. 2015. Communication privacy management theory. *The international encyclopedia of interpersonal communication* (2015), 1–9.
[46] Hai-bo Shen and Fan Hong. 2006. An attribute-based access control model for web services. In *2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06)*. IEEE, 74–79.
[47] Itamar Simonson and Amos Tversky. 1992. Choice in context: Tradeoff contrast and extremeness aversion. *Journal of marketing research* 29, 3 (1992), 281–295.
[48] Welderufael B Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna. 2018. I read but don't agree: Privacy policy benchmarking using machine learning and the eu gdpr. In *Companion Proceedings of the The Web Conference 2018*. 163–166.
[49] Paul Van Schaik. 1999. Involving users in the specification of functionality using scenarios and model-based evaluation. *Behaviour & Information Technology* 18, 6 (1999), 455–466.
[50] Ryan West, Christopher Mayhorn, Jefferson Hardee, and Jeremy Mendel. 2009. The weakest link: A psychological perspective on why users make poor security decisions. In *Social and Human elements of information security: Emerging Trends and countermeasures*. IGI Global, 43–60.
[51] Alan F Westin. 1968. Privacy and freedom. *Washington and Lee Law Review* 25, 1 (1968), 166.
[52] Xiaokui Xiao and Yufei Tao. 2006. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. 229–240.
[53] Mike Z Yao and Daniel G Linz. 2008. Predicting self-protections of online privacy. *CyberPsychology & Behavior* 11, 5 (2008), 615–617.

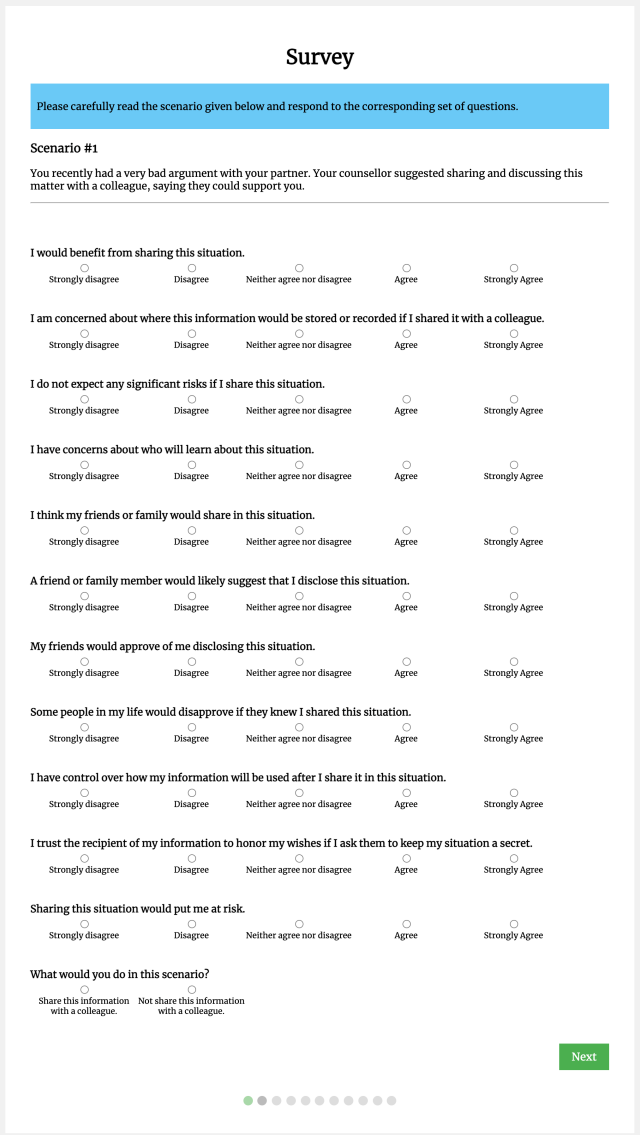


Figure 9: Screenshot of the Survey System Representing 1 of 8 Random Scenarios Given to a Participant.

[54] Junbeom Yoo, Eunkyoung Jee, and Sungdeok Cha. 2009. Formal modeling and verification of safety-critical software. *IEEE software* 26, 3 (2009), 42–49.

A SURVEY INTERFACE 1

Screenshot of the survey system representing 1 of 8 random scenarios given to a participant (Figure 9).

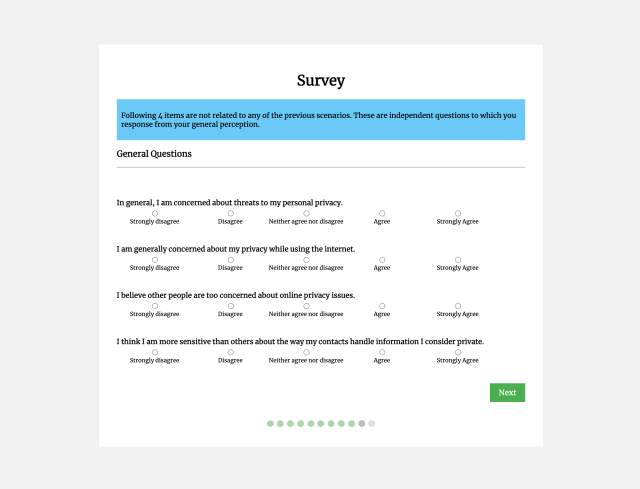


Figure 10: Screenshot of the survey system representing the general attitude questions given to a participant at the end of the survey.

B SURVEY INTERFACE 2

Screenshot of the Survey System Representing the General Attitude Questions Given to a Participant at the end of the Survey (Figure 10).