Arnor: Modeling Social Intelligence via Norms to Engineer Privacy-Aware Personal Agents

Nirav Ajmeri[†], Pradeep K. Murukannaiah[‡], Hui Guo[†], and Munindar P. Singh[†]
[†]Dept. of Computer Science, North Carolina State University, Raleigh, NC 27695-8208, USA
{najmeri, hguo5, mpsingh}@ncsu.edu
[‡]Dept. of Software Engineering, Rochester Institute of Technology, Rochester, NY 14623-5608, USA
pkmvse@rit.edu

ABSTRACT

We seek to address the challenge of engineering socially intelligent personal agents that are privacy-aware. We propose Arnor, a method, including a metamodel based on social constructs. Arnor incorporates social norms and goes beyond existing agent-oriented software engineering (AOSE) methods by systematically capturing how a personal agent's actions influence the social experience it delivers. We conduct two empirical studies to evaluate Arnor. First, via a multiphase developer study, we show that Arnor simplifies application development. Second, via simulation experiments, we show that Arnor provides improved privacy-preserving social experience to end users than personal agents engineered using a traditional AOSE method.

Keywords

Social context; Norms; Personal agents; Methodology; Developer study

1. INTRODUCTION

Our actions and interactions in a society are not driven solely by individual needs. Instead, we adapt our behavior considering the needs of others, e.g., by being courteous and lending a helping hand. Such acts, even if inconvenient at times, deliver a pleasant social experience.

Privacy encompass both social and technical aspects. However, most of the traditional works have approached privacy from a technical standpoint. We tackle the science of privacy from a sociotechnical viewpoint [13, 8].

Consider a society in which an agent acts and interacts on behalf of a *stakeholder* (human user). Our objective is to engineer the agents such that they deliver a *social experience* relative to that society, as opposed to individual user experiences. We refer to an agent delivering a social experience as a *socially intelligent personal agent* (SIPA). The *primary* stakeholder of a SIPA is the user who directly interacts with the SIPA, and on whose behalf the SIPA acts and interacts. A *secondary* stakeholder of a SIPA may not directly interact with the SIPA, but the SIPA's actions affect the secondary stakeholder.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

Example 1. *Consider a ringer manager as a SIPA installed on Alice's phone. The ringer manager decides appropriate ringer modes (e.g., loud or silent) for incoming calls. Alice is the ringer manager's primary stakeholder. Bob, Alice's friend, calls her when Charlie and Dave, Alice's coworkers, are in her vicinity. Bob, Charlie, and Dave are the ringer manager's secondary stakeholders.*

We define social experience as the collective experience a SIPA delivers to each of its primary and secondary stakeholders. Respecting stakeholders' privacy is an important aspect of delivering social experience.

Example 2. *Bob calls Alice when she is in an important meeting with Charlie and Dave.*

In Example 2, should Alice's phone ring loud during the meeting, privacy implications may follow [23, 16]. A loud ring *intrudes* upon Alice's and other meeting attendees' privacy in that the call violates their reasonable expectation to be left alone. Further, Alice may receive nasty looks from other attendees (*disapprobation*). If Alice answers the call, those overhearing Alice and Bob's conversation can gain knowledge about her and her interlocutor (*information leak*).

Example 3. *Alice is in a meeting with Charlie and Dave. Bob is in a car accident and calls Alice for assistance. Bob's ringer manager communicates the urgency to Alice's ringer manager, which then sets her phone to ring loud. It also notifies Charlie and Dave about the situation.*

Should Alice's phone stay silent for Bob's urgent call, Bob's trust for Alice may reduce, affecting their social relationship. Instead, if the phone rings loud and Alice communicates a rationale to Dave and Charlie, presumably, they would not frown at her.

These examples demonstrate the nontrivial decisions a SIPA must make and the implications those decisions have on the stakeholders' social experience and privacy. These nuances prompt us to investigate the research question:

RQ. How can we engineer a SIPA such that it delivers a social experience but respects its stakeholders' privacy?

Three key challenges in engineering a SIPA to deliver a social experience are understanding (1) what constitutes social experience; (2) how a SIPA's actions influence the social experience and privacy for each stakeholder; and (3) how a SIPA's actions evolve when it is put to use in a variety of social contexts.

Existing agent-oriented software engineering (AOSE) methods provide a good starting point for addressing the first challenge. For example, Tropos [7] actor models and Gaia [29] interaction models capture stakeholders and coarse dependencies between them. However, these methods provide little guidance on capturing how an agent's actions and interactions influence each stakeholder involved (second challenge). Also, these methods provide design-time constructs to model an agent, but fall short in modeling social interactions that support agents to adapt to evolving social contexts at run time (third challenge). Our formulation contrasts with Tropos where the stakeholders are characterized by their goals, as in caller, callee, and neighbor, but a single perspective is taken in the actor produced. We consider multiple perspectives where each agent corresponds to one user and has its loyalty to that user.

Norms have been widely studied with several works addressing norm conflicts, compliance, and emergence via either simulation or formalization [3, 9]. Van Riemsdijk [26] argue for a personal agent's need to explicitly represent norms. Social norms inform SIPAs about a set of reasonable actions in a social context. Norm compliance in a social context is achieved either by (1) establishment of norms, where SIPAs are made aware of norms by direct communication, or (2) via (positive and negative) sanctions, where SIPAs learn norms in the form of appropriate actions in a social context [5]. Also, a SIPA's decision rationale for its action influences how other stakeholders perceive satisfaction or violation of a norm, and the nature of sanctions that they apply.

Contribution

To address the aforesaid challenges, we propose Arnor, a systematic method enabling the development of privacy-aware socially intelligent personal agents via social constructs. Arnor facilitates agent developers in modeling stakeholders' social expectations and, how an agent's actions influence those expectations, thereby enabling SIPAs that deliver a rich social experience. Arnor employs Singh's [21] model of (social) norms to capture social requirements, and incorporates argumentation constructs [6] for sharing a decision rationale.

Testing a SIPA's adaptability in all possible social contexts would be infeasible. To overcome this challenge, Arnor incorporates a SIPA simulation testbed. Seeded with crowdsourced data, Arnor's testbed enables designers to test a SIPA's runtime adaptability. We rigorously evaluate Arnor via two studies: (1) a multiphase developer study in which developers engineer a SIPA, and (2) a set of adaptability studies in which we simulate the adaptability of SIPAs developed in the first study in a variety of social contexts.

Novelty

Arnor goes beyond existing AOSE methods by assisting developers to incorporate social norms and reason about how those norms influence social experience. In spirit, Arnor is a hybrid method in that it addresses the problem of engineering SIPAs combining top-down (via modeling) and bottom-up (via experience or social learning [20]) styles.

Section 2 briefly describes the background works on which we build. Section 3 describes Arnor in detail. Section 4 describes our developer and simulation studies, and Section 5 presents our results. Section 6 discusses related work, threats to validity, and concludes with important future directions.

2. BACKGROUND

Arnor builds on the AOSE methods of Tropos and Xipho, and on the constructs of social norms and sanctions.

2.1 Tropos and Xipho

Tropos [7] is an end-to-end AOSE methodology spanning requirements modeling, design, and implementation. Tropos provides systematic steps to model and refine an application to be developed via high-level abstractions.

We adopt the following Tropos abstractions. An *actor* is a social, physical, or a software agent. An actor has *goals* (strategic interests) and *plans* (means of achieving a goal) within a system. Further, an actor's goals can be *hard* (having a specific satisfaction condition) or *soft* (not have a specific satisfaction condition). A *belief* is an actor's perspective of the environment and a *resource* is a physical or information entity. An actor may have *dependencies* with other actors to satisfy goals, execute plans, or acquire resources.

Figures 1 shows a Tropos system-as-is model (the as-is model captures the setting in which the agent to be developed, e.g., the ringer manager, operates). This model identifies the stakeholders and dependencies between them as well as the goals and plans of the stakeholders.

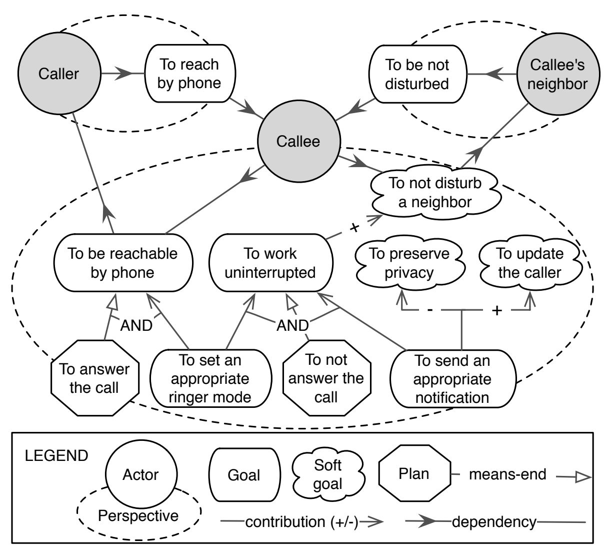

Figure 1: A Tropos system-as-is model of the ringer manager, expanding the callee's perspective [17].

Xipho [17] extends Tropos to engineer personal agents. Xipho introduces *context* as a high-level abstraction and treats an actor's goals, plans, and dependencies as inherently contextual. Xipho enables a developer to tailor a generic model of context to a specific application scenario via systematic steps through distinct development phases.

2.2 Norms and Sanctions

A norm as understood here [21] is directed from a subject to an object and is constructed as a conditional relationship involving an antecedent (which brings the norm in force) and a consequent (which brings the norm to satisfaction or violation). This representation yields clarity on who is accountable to whom. A norm can be formalized as:

$$\mathsf{N}(\text{SUBJECT}, \text{OBJECT}, \textit{antecedent}, \textit{consequent})$$

We employ the following types of norms in our approach.

- A *commitment* (C) means that its subject commits to its object to ensure the consequent if the antecedent holds. An example commitment is that, in a meeting room, the participants may be committed to each other to keep their phones silent: C(PHONE-USER, COWORKER, place = *meeting*, ring = *silent*).
- A *prohibition* means that its subject is forbidden by its object from bringing about the consequent if the antecedent holds. An example prohibition (P) is that, in an examination hall, the students may be prohibited by a proctor from answering phone calls: P(PHONE-USER, PROCTOR, place = *examination*, ring = *silent*).
- A *sanction* specifies the consequences its subject faces from its object for satisfying or violating another norm, such as a commitment or a prohibition. A sanction can be positive, negative, or neutral [18]. A sanction may be in the form of "feedback," e.g., a smile or a scowl, from one user to another. An example sanction (S) is that, in a meeting, if a participant's phone rings loud, he or she receives a scowl from other meeting participants: S(PHONE-USER, COWORKER, place = *meeting* $\wedge$ *ring* = *loud*, feedback = *scowl*).

3. ARNOR

Arnor is a four-step method build on social constructs to systematically model the social experience provided by a SIPA. Arnor's steps include modeling of: (1) goals, (2) environmental contexts, (3) social expectations, and (4) social experience. Figure 2 shows a conceptual model of Arnor. Table 3 provides an overview.

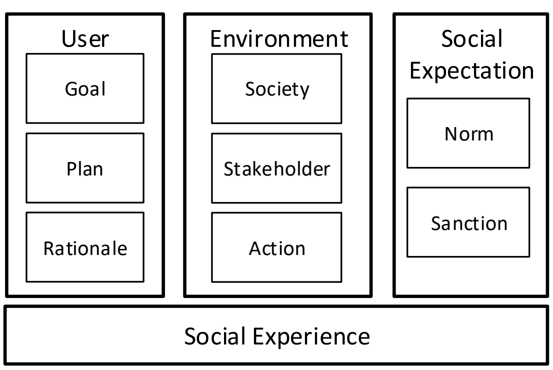

Figure 2: Arnor's conceptual model schematically.

3.1 Goal Modeling

For a SIPA to provide a social experience, it needs to be aware of the associated stakeholders, their goals and relevant plans. Goal modeling in Arnor uses Tropos constructs to elicit stakeholders, their goals, and relevant plans.

A stakeholder is a user that participates in a society and interacts with or is affected by the SIPA. *Primary* stakeholders are the users that interact directly with the SIPA. *Secondary* stakeholders do not have direct interaction with the SIPA, but are affected by its interactions with the primary stakeholder.

A goal is a set of states of the environment that are preferred by the stakeholders.

A plan is a sequence of actions that can bring about a state in which a stakeholder's goal is satisfied. The SIPA acts on behalf of the stakeholders or assists stakeholders in bringing their goals.

Stakeholders in Arnor map to actors in Tropos or Xipho. Whereas Tropos and Xipho explicitly relate actors to the users that have goals, Arnor forces designers to additionally identify (secondary) stakeholders that do not necessarily have a goal, but are affected by the plans that (primary) stakeholders execute to achieve their goals. Capturing secondary stakeholders is necessary to providing a social experience. A stakeholder may adopt different roles.

Following Table 3, we create the goal model for the ringer manager SIPA described in Examples 1–3 and Figure 1.

Primary stakeholder. Alice, the phone user (S_1).

Secondary stakeholders. Bob (Alice's friend, S_2), Charlie and Dave (Alice's coworkers, S_3 and S_4), Erin (Alice's mother, S_5) and strangers (those in the theater who are in Alice's vicinity when the ringer manager SIPA is in use, S_6). Here Bob, Charlie, Dave and Erin could assume the roles of caller and neighbors in different contexts. Note that, although the ringer manager SIPA includes only one primary stakeholder, other settings could involve multiple primary stakeholders.

Goals. The phone user's goals are to be tele-reachable (G_1), to notify caller if not reachable (G_2), to work uninterrupted (G_3), and to avoid annoying neighbors (G_4). Bob, Alice's friend has goals to (1) tele-reach Alice (corresponds to G_1), and (2) be notified if Alice is not reachable (corresponds to G_2). Charlie and Dave's goals are to not be disturbed at work by anyone (same as G_4). Erin's mother has the same goals as Bob. Strangers in Alice's vicinity share the same goal as Charlie and Dave. When Charlie and Dave assume the caller role, they share Bob and Erin's goal of tele-reaching Alice.

Actions. Alice, the phone user, can answer a call if she is available, or can notify the caller otherwise. She could decide not to answer calls if she does not want to be disturbed or does not want to annoy her neighbors. Based on Alice's actions, Bob, Charlie, Dave, Erin, and other stakeholders act. For example, if Alice answers Bob's or Erin's call, they could give Alice a positive feedback. In social expectation modeling, we capture these feedback actions as sanctions.

Plans. The plan corresponding to the *answer call* action is to *set ringer mode on loud* (P_1). The other plans could be to *set ringer mode on vibrate* (P_2) or *set ringer mode on silent* (P_3).

Goal-plan association. The plan of setting the ringer on loud promotes the phone user's goal of being tele-reachable, and caller's goal of tele-reaching the callee. The plan of setting the ringer on silent promotes the phone user's goal to work uninterrupted, and the neighbors' goal of not being disturbed.

3.2 Social Context Modeling

Context modeling includes identifying social contexts in which the stakeholders of a SIPA interact. The social context could include the place where the interaction occurs,

Table 1: Overview of Arnor tasks and examples to engineer a SIPA.

Step	Arnor Task	Example
Goal Modeling	Identify all actors	Alice, Bob, Charlie, Dave, Erin, and strangers in the theater
	Abstract actors as primary and secondary stakeholders, as appropriate	Phone user is a primary stakeholder; friend, coworker, stranger in the vicinity of phone users are secondary stakeholders
	Identify goals of each actor	Phone user's goals *to be tele-reachable*, and *to be not disturbed*
	Identify all actions, and abstract them as appropriate	*Phone users do not answer phone calls during meetings*; *phone users answers their coworkers' urgent phone calls*
	Identify plans for abstract actions	*Set ringer mode as loud* for the action *phone user answers a phone call*
	Associate goals with plans	Phone user's goal of *tele-reachable* can be realized by the plan of *setting ringer mode as loud*
Context Modeling	Identify the contexts in which each actor's goals and plans are relevant	Coworker's goal *to be not disturbed* is relevant in the *meeting* context
	Identify conflicting goals (and inconsistent plans)	Phone user's goal of *tele-reachable* conflicts with the goal *to not disturb neighbors* in the *meeting* context
Social Expectation Modeling	Identify norms relevant to social and privacy expectations	*The phone user is committed to answering urgent phone calls from family*
	Identify possible conflicts between norms	*Phone user's commitment toward friend to answer phone calls* conflicts with *phone user's commitment to keep phone on silent during meeting*
	Resolve conflicts by capturing contextual preferences between norms	In the *meeting* context, prefer *phone user's commitment to keep phone on silent during meeting* over *phone user's commitment toward friend to answer phone calls*
Social Experience Modeling	Identify effects of stakeholders' actions on social expectations	A norm that is consistently being violated, e.g., *phone users always answering calls during meeting*
	Promote actions that enhance social experience	

attributes of the place, neighbors in the vicinity, the social relationship between primary and secondary stakeholders, the activities the stakeholders are involved in, and so on. The social context is decisive in identifying the goals to be brought about or plans to be executed in case of conflicts.

Some of the contexts associated with goals, G_1–G_4, and plans, P_1–P_3, are based on stakeholders' locations (meeting or theater), social relationship (colleagues, friends or family), reason associated with a phone call (urgent phone call or a casual phone call), and so on.

Goal G_1 of being tele-reachable conflicts with goals G_3 and G_4 for both the meeting and theater scenarios. In these scenarios, the SIPA must rely on social contexts to determine which goal to accomplish. Potentially, where multiple plans may help realize the same goals. For example, in a library, both the *phone on silent* plan and *phone on vibrate* plans serve the goal of not disturbing one's neighbors. The SIPA relies on social context to choose between multiple plans.

3.3 Social Expectation Modeling

Social expectations including the privacy ones influence the stakeholders' goals and plans. We model these expectations between stakeholders in terms of social norms and sanctions. The social norms of a society regulate how stakeholders act and conduct themselves. Some norms could be local to a stakeholder, for example, one's commitment toward family members to always answer their phone calls, and some norms could be specific to a social context, for example, in the context of a meeting, a phone user is committed to keep his or her phone silent.

We express social expectations for the ringer manager SIPA via norms, sanctions and conflicts.

Norms. We identify the following norms.

- A phone user is committed to answering phone calls from callers. This commitment is satisfied by the plan of setting the ringer mode on loud.
  C_{caller}: $\mathsf{C}(\textsc{phone-user}, \textsc{caller}, \text{call}, \text{ring} = \textit{loud})$
- A phone user is committed to notifying the caller if he or she does not answer. The commitment is satisfied by the plan of setting the ringer mode on silent and sending a notification to the caller.
  C_{notify}: $\mathsf{C}(\textsc{phone-user}, \textsc{caller}, \text{call},$
  $\text{ring} = \textit{silent} \wedge \textit{notify})$
- A phone user is committed to coworkers to not let the phone ring during meetings. This is satisfied by the plan of setting the ringer mode on silent or vibrate.
  $C_{meeting}$: $\mathsf{C}(\textsc{phone-user}, \textsc{coworkers}, \text{call},$
  $\text{ring} = \textit{silent} \vee \text{ring} = \textit{vibrate})$

Sanctions. The associated sanctions are as below:

- A phone user is (negatively) sanctioned by coworkers for answering a phone call during a meeting.
  $S_{meeting}$: $\mathsf{C}(\textsc{phone-user}, \textsc{coworkers}, \text{call}$
  $\wedge\ \text{place} = \textit{meeting} \wedge \text{ring} = \textit{loud}, \text{feedback} = \textit{negative})$

Conflicts. If a caller calls the phone user during a meeting, the phone user's commitment C_{caller} toward a caller conflicts with his or her commitment $C_{meeting}$ toward coworkers to not answer phone calls during meetings, i.e., *conflict*$(C_{caller}, C_{meeting})$.

Conflicts in social expectations can be resolved by capturing contextual preferences between conflicting norms. For example, a phone user can have a preference of $C_{meeting}$ (*keep phone on silent during meetings*) to C_{caller} (*answer calls from family members*).

3.4 Social Experience Modeling

Norms are satisfied or violated as stakeholders act and execute plans to achieve their goals. Norm satisfaction or violation provides positive or negative experience to the stakeholders. As agents derive social experience from norms, over time, certain norms are preferred over others, and some lose significance. If a certain phone user is always answering phone calls during meetings, the phone user could be banished from meetings. A SIPA should execute actions that promote yield social experience by choosing which plans to execute, which goal states to accomplish, and which norms to satisfy. To decide which actions to promote, SIPAs could employ argumentation [6], and make use of argumentation schemes such as *arguments from consequences*, and *arguments from popular opinion* [28]. Additionally, a SIPA, depending upon its user's privacy attitude and information sharing preferences, can choose to share its decision rationale for choosing an action with the other stakeholders. The sharing of rationale could introduce nuances in social relationships of a SIPA's stakeholders such as increase of trust that we do not model.

4. EVALUATION

We investigate our research question by evaluating Arnor via a developer study and a simulation experiment.

4.1 Developer Study

We begin with a multiphase developer study in which participants develop ringer manager SIPAs. Our study was approved by the Institutional Review Board (IRB). We obtained informed consent from each participant. The developer study lasted for six weeks.

Study Unit

The study unit is a ringer manager SIPA discussed in Examples 1–3 and Figure 1.

Participants

The developer study involved 30 participants, enrolled in a graduate-level computer science course. The participants earned points toward their course grades for completing the tasks described. However, participation in the study was not mandatory. Nonparticipants were offered an alternative task to earn points equivalent to what they would earn by participating in the study.

Study Mechanics

This developer study has two phases: learning and development. The study follows the one-factor design with two alternatives (Arnor and Xipho). We use Xipho as our baseline method because it is best suited among the existing AOSE methods to engineer personal agents.

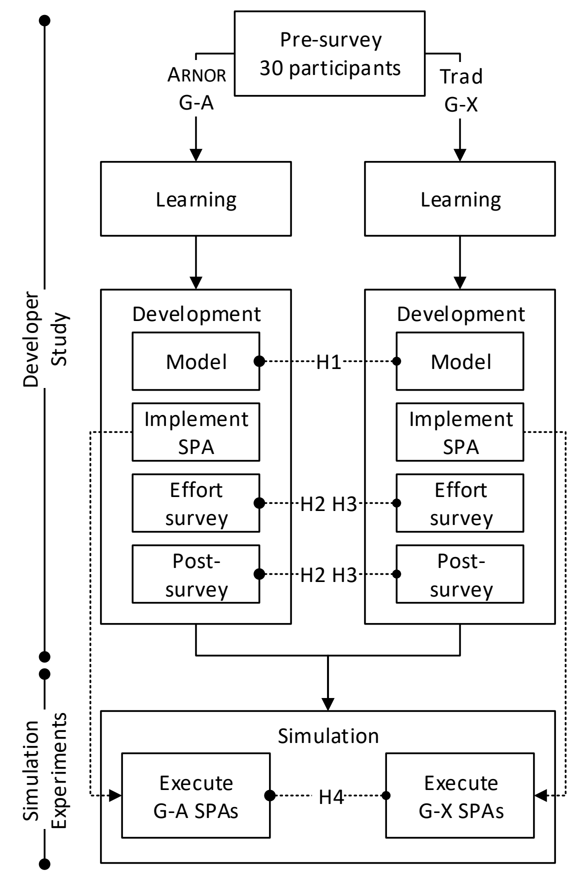

Figure 3: Experimental Design.

We split participants into two groups (A that follows Arnor, and X that follows Xipho) balanced on skills indicated in a presurvey. All participants develop a ringer manager SIPA.

Learning Phase. During the learning phase of the study, participants proposed a SIPA, and created models of the proposed SIPA. This phase sought to help participants understand the nuances of a SIPA, and to teach them how to model requirements. The data collected in the learning phase is not used in the evaluation.

Development Phase. In the development phase, participants modeled and implemented a ringer manager SIPA that adapts according to expectations of callers and neighbors, and sanctions received from callers and neighbors for each action.

In the two development phases, participants were provided with a testbed to verify the working of their SIPAs.

Deliverables

The participants submitted models and source code at the completion of the development phase. Additionally, the participants completed a time and effort survey for each work session, and completed a post-phase survey at the end of each phase.

Metrics

To measure the effectiveness of Arnor, we compute the following metrics.

Model coverage measures the completeness of the model. It is the ratio of the number of requirements identified correctly in the produced model to the total number of requirements of the SIPA. Higher is better.

Model correctness measures how correct the model is. It is the ratio of the number of correctly identified requirements to the total number of requirements of the SIPA identified. Higher is better.

Model quality is the product of model coverage and model correctness. Higher is better.

Time to develop is the actual time spent by participants in hours to develop the SIPA. Lower is better.

Difficulty of development is the subjective rating by participants on how easy it is to develop the SIPA on a Likert scale of 1 (very easy) to 7 (very difficult). Lower is better.

Effort to develop is the product of time spent in hours and ease of development rating for each work session. Lower is better.

Hypotheses

We consider the following hypotheses.

H_1. Developers who follow Arnor produce better quality models than those who follow Xipho.

H_2. Developers who follow Arnor spend less time to develop a SIPA, than those who follow Xipho.

H_3. Developers who follow Arnor feel it is easier to develop a SIPA, than those who follow Xipho.

H_4. Developers who follow Arnor expend less effort to develop a SIPA, than those who follow Xipho.

Threats and Mitigation

We mitigated three main threats to our studies. Differences amongst participants' programming and modeling skills are inevitable. To handle the skill differences between participants, we surveyed participants about their educational backgrounds and prior experiences with programming and conceptual modeling. We balanced the two groups based on the survey. To mitigate the risk of participants' failing to report information, participants were instructed to complete a time and effort survey after each work session, while it was fresh in their minds. Communication between participants of different groups is yet another threat. To mitigate the risk of contamination, we created separate message boards for each participant group, and restricted participants to only posting clarification questions on the group message boards.

4.2 Simulation Experiments

We further investigate our research question via simulation experiments. We execute the ringer manager SIPAs developed in the developer study on a testbed fabricated to simulate different real-world environments.

Ringer adaptation scenarios

To test runtime adaptability, we test the applications for multiple iterations of incoming phone calls during a meeting.

Norms fixed. The meeting room participants are committed to keeping their phones silent.

Norms change. The meeting room participants are initially committed to keeping their phones silent, but later the commitment expires.

Context change. The meeting room participants are always committed to keeping their phones silent. Initially there are several participants in the meeting, but later all but two leave the meeting.

Sanction change. The meeting room participants are always committed to keeping their phones silent. Initially they give negative feedbacks for loud ringing but later give more neutral feedbacks.

Metrics

To measure social experience, we compute the following social metrics in each of the above adaptation scenarios.

Adaptability coverage measures the completeness of code for adaptability requirements. It is the ratio of the number of adaptability requirements implemented correctly to the total number of adaptability requirements. Higher is better.

Adaptability correctness measures the correctness of the code for adaptability requirements. It is the ratio of the number of correctly implemented adaptability requirements to the total number of adaptability requirements implemented. Higher is better.

Norm compliance refers to the proportion of norm instances that are satisfied. Higher is better.

Sanction proportion measures the percentage of sanctions imposed. Lower is better.

Hypotheses

We consider these additional hypotheses:

H_5. SIPAs developed using Arnor yields better adaptability than SIPAs developed using Xipho.

H_6. SIPAs developed using Arnor provide a richer social experience than SIPAs developed using Xipho.

We use adaptability coverage and correctness to test hypothesis H_5, and use norm compliance and sanction proportion measures to test hypothesis H_6.

5. RESULTS

We analyze deliverables produced by participants at the end of each phase, and compute the study parameters for each deliverable.

5.1 Developer Study

To test hypothesis H_1, we compare the models produced by Groups A and X. For hypothesis H_2, we compare the development time expended by Groups A and X during the two development phases. For hypothesis H_3, we compare the ease of development ratings reported by Groups A and X during the two development phases, and for hypothesis H_4, we compare their expended effort.

Model quality. We evaluated models produced by the participants for correctness and coverage, and computed a quality metric. We found no significant difference in model quality.

Time and effort to develop. We found that average time (13.27 hours) and effort (61.54) expended by the participants using Arnor to be lower than average time (17.72 hours) and effort (96.6) expended by the participants using Xipho. Figures 4 and 5 shows the boxplots for time and effort expended by participants using Arnor and Xipho to develop the social ringer SIPA.

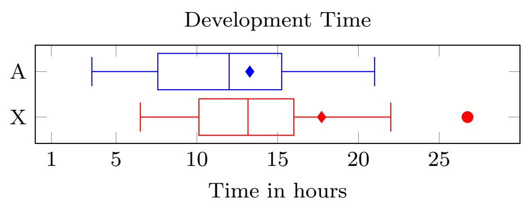

Figure 4: Arnor vs. Xipho's development time in hours as reported in the work session surveys.

Difficulty of development. The participants using Arnor found it easier to develop SIPAs with Arnor, compared to participants using Xipho. Figure 6 shows the difficulty of development boxplots.

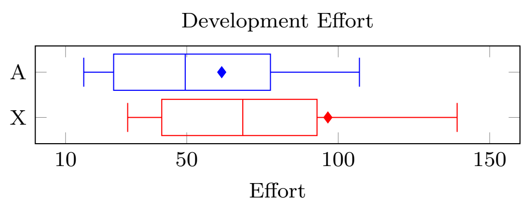

Figure 5: Arnor vs. Xipho's development effort as reported in the work session surveys.

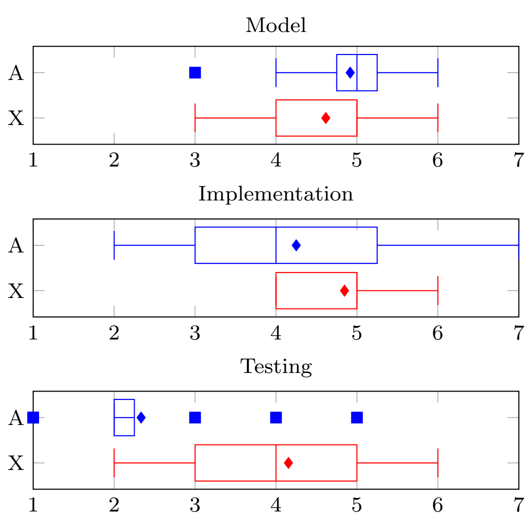

Figure 6: Arnor vs. Xipho's difficulty of development on a Likert scale of 1 (very easy) to 7 (very difficult).

5.2 Simulation Experiments

To evaluate H_5 and H_6, we analyzed the SIPA's implementation code and executed the SIPAs in diverse scenarios. We compare the execution results of Arnor and Xipho groups.

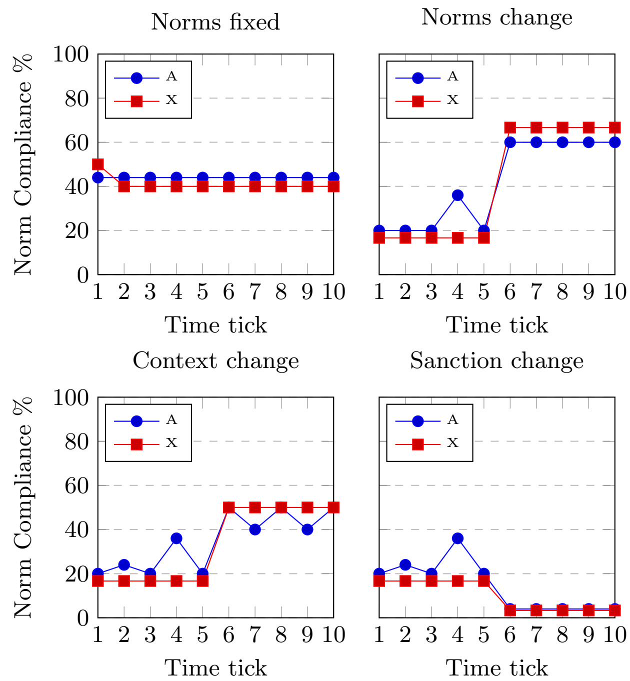

Figure 7: Arnor vs. Xipho's norm compliance.

Adaptability features. We found average adaptability coverage (80%) to be the same for SIPAs developed by the Arnor and Xipho groups. This result could be attributed to the limited time we gave the participants to develop the SIPA. Average adaptability correctness was found to be higher for Arnor (100%) compared to the Xipho (95%). This gain could be attributed to the systematic steps provided by Arnor to engineer SIPAs.

Norm compliance. Figure 7 shows line plots for norm compliance in the four ringer adaptation scenarios. Though the average norm compliance values for SIPAs developed using Arnor and Xipho are mostly similar, Arnor performs slightly better in the fixed norms scenario.

Sanction proportion. Figure 8 shows the plots for sanction proportion in the four adaptation scenarios. For the first three scenarios (norms fixed, norms change, and context change), the SIPAs developed using Arnor have a lower sanction proportion. For the sanction change adaptation scenario, the SIPAs developed using Arnor take slightly longer to adapt, and only have a slightly higher sanction proportion than the SIPAs developed using Xipho.

5.3 Threats to Validity

In the developer study, we mitigated the threats of skills difference, participants' failure to report information, and the risk of contamination. However, some threats remain.

First, our results are based only on the development of a single SIPA (ringer). For conclusive results on the effectiveness of Arnor, future studies may require participants to develop more than one kind of SIPA.

Second, the SIPAs developed by the study participants mostly reflect the participants' (developers) privacy attitudes and information sharing preferences. To generalize our results, it is required to collect real data on SIPA users' privacy attitudes and information sharing preferences.

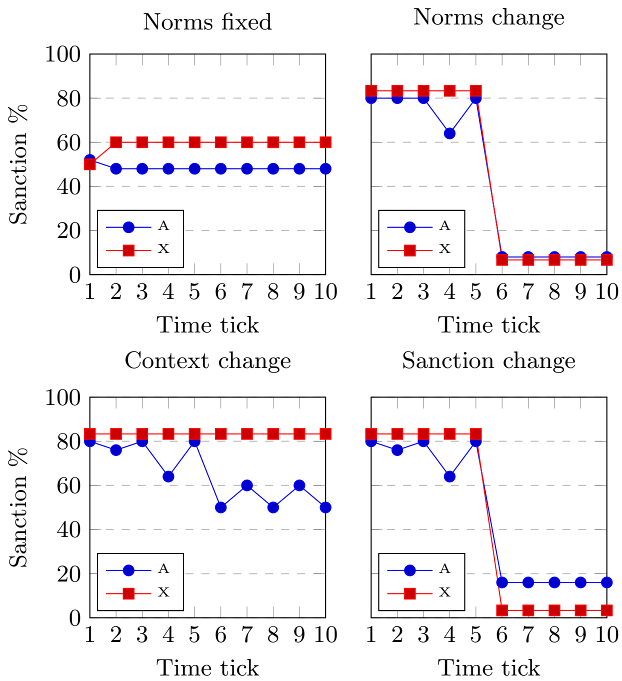

Figure 8: Arnor vs. Xipho's sanction proportion.

Third, in simulation experiments, we tested runtime adaptability of SIPAs under diverse, but a limited set of scenarios. The scenarios we incorporated may not represent all real world scenarios in which a ringer SIPA would be employed.

Collecting real data about users' attitudes, preferences, and contexts is essential, though nontrivial, to mitigate the second and third threat. Crowdsourcing is a promising avenue for future studies to collect such data at a large scale.

6. DISCUSSION

We advance the science of privacy by tackling nuanced notions of privacy, including intrusion, disapprobation, and information leakage, in personal agents. We treat respecting stakeholders' privacy as an inherent aspect of delivering a social experience. We envision socially intelligent personal agents that (1) adapt to the social contexts of their stakeholders; and (2) act and interact in their best interest (not just the primary stakeholder).

We develop Arnor, a method that provides social constructs to engineer privacy-aware social agents. We demonstrate the method via a ringer manager SIPA. We evaluate Arnor using a developer study and simulation experiments. Compared to Xipho, we find that Arnor (1) facilitates faster development of SIPAs; and (2) yields SIPAs of higher quality, higher adaptability correctness, lower sanction proportion, and similar adaptability coverage and norm compliance. These observations suggest that Arnor promotes SIPAs to deliver a rich social experience.

6.1 Related Works

Ali et al. [4] propose an AOSE-based contextual requirements engineering framework, with a focus on consistency and conflict analysis. Arnor goes beyond conflict analysis, and promotes goals, plans, and norms that promote greater social experience. Rahwan et al. [19] propose a framework to integrate goal models and social models. Arnor models subsume social models, and provide richer abstractions to capture agents' interactions and affects on experience.

Sugawara et al. [24] attempt to resolve conflicts through reinforcement learning. Mashayekhi et al. [14] propose a hybrid mechanism to monitor interactions and recommend norms to resolve conflicts. Mihaylov et al. [15] study convergence and propose a decentralized approach based on strategies in game theory. Villatoro et al. [27] introduce social instruments to facilitate norm emergence via social learning. Yu et al. [30] study norm emergence through collective learning from local interactions, and find that collective learning is superior to pairwise learning. Arnor provides constructs to engineer socially adaptable SIPAs that can make use of these approaches for norm emergence.

Hao et al. [12] propose a lightweight formal method to design normative systems, which uses Alloy modeling language and analyzer to synthesize and refine norms. Van Riemsdijk et al. [25] propose a semantic norm compliance framework for socially adaptive agents. They use LTL to express norms. Agents in van Riemsdijk et al.'s framework identify and adopt new norms, and determine execution mechanisms to comply to these norms. Aldewereld et al. [2] present a formalism for group norms, and provide mechanisms to reason about these norms. Ajmeri et al. [1] propose Coco, a formalism to express and reason about conflicting commitment instances at runtime, and dominance among them. Coco employs Answer Set Programming to compute the nondominated commitment instances and determines compliance of actions with nondominated commitment instances. These formalisms could use Arnor's social constructs to assist SIPAs in compliance, adoption of new norm, and resolution of conflicts amongst norms at runtime.

6.2 Future Directions

Ferreira et al. [10] propose a computational model for emotional agents that considers norms, social relations, roles and socially acceptable behaviors in a given context. Sollenberger and Singh [22] introduce Kokomo to develop affective applications, and provide a middleware for building such applications. Incorporating an affective [22] and emotional basis of norms in social agents is an interesting future direction. Modeling affect could assist SIPAs learn contextually relevant norms. A middleware implementation of Arnor could facilitate development.

Fogues et al. [11] study how context, users' preferences, and arguments influence a sharing decision in a multiuser privacy scenario. They collect data about appropriate sharing policies for a variety of multiuser scenarios from human participants in a large scale study. We conjecture that such data can be used to seed SIPAs with an initial set of norms, which the SIPAs can evolve once put to use.

Acknowledgments

We thank the US Department of Defense for support through the Science of Security Lablet at NC State University and the anonymous reviewers for helpful comments.

REFERENCES

[1] N. Ajmeri, J. Jiang, R. Chirkova, J. Doyle, and M. P. Singh. Coco: Runtime reasoning about conflicting

commitments. In *Proc. IJCAI*, pages 17–23, New York, 2016.

[2] H. Aldewereld, V. Dignum, and W. W. Vasconcelos. Group norms for multi-agent organisations. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 11(2):15:1–15:31, June 2016.

[3] N. Alechina, J. Y. Halpern, I. A. Kash, and B. Logan. Decentralised norm monitoring in open multi-agent systems: (extended abstract). In *Proc. AAMAS*, pages 1399–1400, Singapore, 2016.

[4] R. Ali, F. Dalpiaz, and P. Giorgini. Reasoning with contextual requirements: Detecting inconsistency and conflicts. *Information and Software Technology*, 55(1):35–57, 2013.

[5] G. Andrighetto, J. Brandts, R. Conte, J. Sabater-Mir, H. Solaz, and D. Villatoro. Punish and voice: Punishment enhances cooperation when combined with norm-signalling. *PLoS ONE*, 8(6):1–8, 06 2013.

[6] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, July 2007.

[7] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multiagent Systems*, 8(3):203–236, May 2004.

[8] A. K. Chopra and M. P. Singh. From social machines to social protocols: Software engineering foundations for sociotechnical systems. In *Proc. WWW*, pages 903–914, Montréal, 2016.

[9] N. Criado and J. M. Such. Selective norm monitoring. In *Proc. IJCAI*, pages 208–214, New York, 2016.

[10] N. Ferreira, S. Mascarenhas, A. Paiva, G. Di Tosto, F. Dignum, J. McBreen, N. Degens, G. J. Hofstede, G. Andrighetto, and R. Conte. An agent model for the appraisal of normative events based in in-group and out-group relations. In *Proc. AAAI*, pages 1220–1226, Bellevue, 2013.

[11] R. L. Fogués, P. K. Murukannaiah, J. M. Such, and M. P. Singh. Understanding sharing policies in multiparty scenarios: Incorporating context, preferences, and arguments into decision making. *ACM Transactions on Computer-Human Interaction*, 24(1), Feb. 2017. To appear.

[12] J. Hao, E. Kang, J. Sun, and D. Jackson. Designing minimal effective normative systems with the help of lightweight formal methods. In *Proc. FSE*, pages 50–60, Seattle, 2016.

[13] Ö. Kafalı, N. Ajmeri, and M. P. Singh. Revani: Revising and verifying normative specifications for privacy. *IEEE Intelligent Systems*, 31(5):8–15, Sept. 2016.

[14] M. Mashayekhi, H. Du, G. F. List, and M. P. Singh. Silk: A simulation study of regulating open normative multiagent systems. In *Proc. IJCAI*, pages 373–379, New York, 2016.

[15] M. Mihaylov, K. Tuyls, and A. Nowé. A decentralized approach for convention emergence in multi-agent systems. *Autonomous Agents and Multiagent Systems*, 28(5):749–778, 2014.

[16] P. K. Murukannaiah, N. Ajmeri, and M. P. Singh. Engineering privacy in social applications. *IEEE Internet Computing*, 20(2):72–76, Mar 2016.

[17] P. K. Murukannaiah and M. P. Singh. Xipho: Extending Tropos to engineer context-aware personal agents. In *Proc. AAMAS*, pages 309–316, Paris, 2014.

[18] L. G. Nardin, T. Balke-Visser, N. Ajmeri, A. K. Kalia, J. S. Sichman, and M. P. Singh. Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *The Knowledge Engineering Review (KER)*, 31:142–166, Mar. 2016.

[19] I. Rahwan, T. Juan, and L. Sterling. Integrating social modelling and agent interaction through goal-oriented analysis. *Computer Systems Science and Engineering*, 21(2), 2006.

[20] S. Sen and S. Airiau. Emergence of norms through social learning. In *Proc. IJCAI*, pages 1507–1512, Hyderabad, 2007.

[21] M. P. Singh. Norms as a basis for governing sociotechnical systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):21:1–21:23, Dec. 2013.

[22] D. J. Sollenberger and M. P. Singh. Kokomo: An empirically evaluated methodology for affective applications. In *Proc. AAMAS*, pages 293–300, Taipei, 2011.

[23] D. J. Solove. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477–564, 2006.

[24] T. Sugawara. Emergence and stability of social conventions in conflict situations. In *Proc. IJCAI*, pages 371–378, Barcelona, 2011.

[25] M. B. van Riemsdijk, L. Dennis, M. Fisher, and K. V. Hindriks. A semantic framework for socially adaptive agents: Towards strong norm compliance. In *Proc. AAMAS*, pages 423–432, Istanbul, May 2015.

[26] M. B. van Riemsdijk, C. M. Jonker, and V. Lesser. Creating socially adaptive electronic partners: Interaction, reasoning and ethical challenges. In *Proc. AAMAS*, pages 1201–1206, Istanbul, 2015.

[27] D. Villatoro, J. Sabater-Mir, and S. Sen. Robust convention emergence in social networks through self-reinforcing structures dissolution. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 8(1):2:1–2:21, Apr. 2013.

[28] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[29] M. Wooldridge, N. R. Jennings, and D. Kinny. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, Sept. 2000.

[30] C. Yu, M. Zhang, F. Ren, and X. Luo. Emergence of social norms through collective learning in networked agent societies. In *Proc. AAMAS*, pages 475–482, Saint Paul, 2013.