ABSTRACT

AJMERI, NIRAV. Multiagent Systems for Privacy-Aware Social Computing. (Under the direction of Munindar P. Singh.)

Research Theme:

A socially intelligent personal agent understands and helps its user respect the norms governing the user's interaction in a society. This research addresses the research question of how can we engineer social intelligence in personal agents to deliver a privacy-preserving social experience. Addressing this research question, we develop multiagent system techniques to engineer such personal agents.

Methods:

This research develops (1) Arnor, an agent-oriented software engineering (AOSE) method to engineer social intelligence in personal agents, and (2) Poros, a framework that enables personal agents to reason about shared contexts, and learn contextually relevant social norms.

Arnor goes beyond traditional AOSE methods to engineer personal agents by systematically capturing interactions that influence social experience.

A personal agent may deviate from norms under certain contexts. Poros (1) enables personal agents deviating from norms to share deviation contexts with other agents in the agent society, and (2) provides personal agents the ability to reason about shared contexts.

Claims:

We claim that (1) Arnor supports developers in engineering personal agents, and (2) personal agents engineered using Arnor provide a greater social experience than agents engineered using a traditional AOSE method. We evaluate Arnor via a developer study, and a set of simulation experiments.

We make two claims about the impact of context sharing and reasoning in Poros. First, ability to reason about deviation contexts helps personal agents accurately learn applicable context-dependent norms. Second, by acting according to such shared context-dependent norms, a personal agent can provide its users a more satisfying social experience than an agent that does not reason about shared context. We demonstrate these claims via social simulations involving agent societies of varying sizes and diverse characteristics reflecting pragmatic, considerate, and selfish agents.

Multiagent Systems for Privacy-Aware Social Computing

by
Nirav Ajmeri

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2017

APPROVED BY:

Jon Doyle	William Enck
Chris Mayhorn	Jessica Staddon
Laurie Williams	Munindar P. Singh Chair of Advisory Committee

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1 INTRODUCTION

Privacy encompass both technical and technical aspects. But the literature in privacy research has focused on these aspects as two different goals. One aims to design secured systems with the help of cryptographic protection. The other aims to protect personal information by facilitating informed choice options to an individual and assume that policies and regulations are enforceable. This research tackles the science of privacy from a sociotechnical viewpoint that bridges the two goals.

Human interactions in society are not merely driven by personal needs and expectations (defined later in Chapter 2.3). Others around us and their expectations play a prominent part on the way we act and interact. A personal agent acts and interacts on behalf of its human user.

A *socially intelligent personal agent* (SIPA) adheres to *social expectations* of multiple *stakeholders*—both *primary* and *secondary* (defined later in Chapter 2), adapts according to the circumstances or *social context* [Dey, 2001], acts on behalf of its human user (primary stakeholder), and provides a pleasing social experience to all of its stakeholders as opposed to individual experience to its human user.

The key objectives of this research are: (1) to engineer personal agents such that they deliver a pleasant social experience relative to the society, and yet preserve their stakeholders' privacy, and (2) to make engineering of such personal agents efficient and effective for developers. We recognize social norms and social context as important factors that influence the working of such personal agents.

1.1 Preliminaries

1.1.1 Privacy

The concept of privacy encircle several areas. An individual's notion of privacy is partly based upon the society's notion of privacy and partly based upon his or her personal experiences [Westin, 1967, 2003]. The general attitude towards sharing personal information varies from one individual to other. Westin [1967] classifies individuals based on their privacy preferences: *privacy fundamentalists* are the individuals who are extremely concerned about their privacy and are reluctant to share personal information; *privacy pragmatists* are concerned about privacy but less than fundamentalists and they are willing to disclose personal information when some benefit is expected; and , *privacy unconcerned* do not consider privacy loss when disclosing personal information [Westin, 1967]. The pragmatists are further grouped into identity-aware and profile-aware individuals [Spiekermann and Cranor, 2009]. Identiy aware individuals are the ones who are more concerned about revealing identifying information such as e-mail or physical address rather than revealing their interests. Profile aware individuals worry more about sharing their hobbies, age, interests, or preferences.

Privacy has always been a topic of debate and has no globally agreed-upon definition [Smith and Shao, 2007]. Theorists from the area of law, philosophy, sociology, politics and computer science have all tried to define privacy in their own perspective. The importance of privacy and what it brings to an individual is also debated. Although there is no single definition, researchers acknowledge the idea of privacy, and view it as a collection of concepts instead of one specific concept [Smith and Shao, 2007]. In this research, we adopt the nuanced notions (specifically intrusion, appropriation, and disclosure) of privacy as defined in Solove's taxonomy [Solove, 2006].

1.1.2 Social Norms and Multiagent Systems

The concept "norm" is used in different disciplines and thus has variant notions such as social expectations, legal laws, and linguistic imperatives [Boella et al., 2009b]. We adopt the concept of social norms, which describe interactions between principals in terms of what they ought to be, or reactions to behaviors, including attempts to apply sanctions. Thus, social norms regulate the interactions of the principals involved. Norms and normative systems are gaining increasing interest in the computer science community. Meyer and Wieringa [1993] define normative systems as "systems in the behavior of which norms play a role and which need normative concepts in order to be described or specified." Normative multiagent systems as a research area can be defined as the intersection of normative systems and multiagent systems. Norms govern much of our social lives, and therefore are considered as a key element of artificial agents that are expected to behave comparably to humans [Boella et al., 2006]. We adopt Singh's [Singh, 2013] representation of social norms in which norms are classified as five types: commitment, authorization, prohibition, sanction and power.

1.2 Motivational Example

Example 1 *Consider a ringer manager as a SIPA. The ringer manager installed on Alice's phone decides appropriate ringer modes (loud, silent, or vibrate) for incoming calls. Alice, the phone owner is the primary stakeholder of the SIPA. Bob, Alice's friend who calls Alice often, and Charlie and Dave, Alice's coworkers, who are in her vicinity, are some of the secondary stakeholders. Further, the ringer manager's capabilities influencing its social experience include (1) allowing Alice to be tele-reachable, (2) notifying the caller if Alice is not reachable, (3) enabling Alice to work uninterrupted, and (4) not annoying Alice's neighbors.*

Suppose that Bob calls Alice when she is in an important meeting with Charlie and Dave. Alice is *committed* (a social norm) to answering Bob's phone calls. Another *commitment* is to keep one's phone silent during important meetings. Alice's SIPA, understanding the norms and knowing that Bob's calls to Alice are generally casual, puts Alice's phone on silent for Bob's call and notifies Bob that Alice is in a meeting; later when Alice's meeting ends, Alice's SIPA reminds her to call Bob.

Should Alice's phone rings loudly during the meeting, privacy implications may follow [Murukannaiah et al., 2016; Solove, 2006]. A loud ring *intrudes* upon Alice's and other meeting attendees' privacy in that call violates the meeting attendees' reasonable expectation to be left alone. Further, it is likely that meeting attendees frown at Alice (*disapprobation*). If Alice answers the call, those overhearing Alice and Bob's conversation can gain knowledge about her and her interlocutor (*information leak*). If Bob's call were urgent, Bob's SIPA could communicate the urgency to Alice's SIPA, and Alice's SIPA could deliver a different social experience, e.g., set phone on vibrate to notify Alice of urgency and yet not annoy other meeting attendees. Should Alice's phone stays silent for Bob's urgent call, it may affect their relationships.

In the examples above, ringer manager SIPA makes nontrivial decisions influencing social experience of its stakeholders. Existing AOSE methods [Bresciani et al., 2004; Murukannaiah and Singh, 2014; Winikoff and Padgham, 2004] are good starting point to engineer personal agents, however these methods do not guide developers with systematic steps to represent and reason about such scenarios, and thus fall short in supporting agents that adapt to evolving social contexts at runtime.

Social norms inform personal agents about a set of reasonable actions in a social context [van Riemsdijk et al., 2015a]. Norm compliance in a social context is either achieved by (1) conveyance of norms, where SIPAs are made aware of norms by direct communication, or (2) via (positive and negative) sanctions, where personal agents learn norms in the form of which actions are appropriate in a context [Andrighetto et al., 2013].

Under certain circumstances, we (as humans) may deviate from norms. When we deviate, we may offer an explanation typically revealing the context of the deviation. Revealing context may lessen the burden of deviation, and may help us avert sanction resulting from the deviation. Deviations from norms often hint toward a different norm that is contextually relevant. For instance if Alice reveals to meeting attendees' that the call was from a sick friend who needs urgent care, the meeting attendees' (1) may not frown on Alice, and (2) may learn that although it is not appropriate to answer calls during meetings as it intrudes upon attendees' privacy, answering an emergency call is acceptable as it could ensure someone's well being or safety. An ability to reason about deviation context, and an understanding of values promoted or demoted by different actions could assist SIPAs in providing a pleasing social experience to its stakeholders.

We recognize three key challenges. One, understanding what constitutes a social experience, and how SIPA's actions influence the social experience and privacy of its stakeholders? When SIPAs satisfy or violate norms, they might share certain contextual information related to satisfaction or violation. Social experience largely depends on how SIPAs' stakeholders perceive shared information. Two, how and what contextual information should a SIPA disclose? When norms conflict, SIPAs must execute actions that promote richer social experience. Three, how can we develop decision support to recommend actions?

1.3 Research Questions

Based on aforementioned challenges and nuances in the example, we seek to probe the following research questions:

RQ 1. How can we model social intelligence in a SIPA such that it delivers a social experience but respects its stakeholders' privacy?

RQ 2. How can we enable a SIPA to share deviation contexts, and to reason about contexts shared by other SIPAs?

RQ 3. Does a SIPA's ability to reason about *values* each of its actions could promote or demote, help it in enriching the social experience delivered to its stakeholders?

1.4 Contributions and Organization

1.4.1 Arnor: Modeling Social Intelligence via Norms in Privacy-Aware Personal Agents

To address the research question of modeling social intelligence, we develop Arnor [Ajmeri et al., 2017], a systematic method AOSE method. It facilitates developers to model stakeholders' actions and expectations, and how these influence each other. Arnor employs Singh's [Singh, 2013] model of (social) norms to capture social requirements, and incorporates argumentation constructs [Bench-Capon and Dunne, 2007] for sharing decision rationale. Since, testing a SIPA's adaptability in all possible social contexts is logistically challenging and time consuming, Arnor also incorporates a SIPA simulation testbed. We rigorously evaluate Arnor via a developer study and a set of simulation experiments on the simulation testbed.

Developer Study

We hypothesize that the developers who follow Arnor (1) produce better models, (2) expend less time, (3) feel it is easier to develop a SIPA, and (4) expend less effort, than those who follow Xipho. We find that developers using Arnor spend less time and effort, and overall feel it is easier to engineer a SIPA using Arnor. No significant difference is found in the model quality.

Simulation Experiment

We hypothesize that SIPAs developed using Arnor (1) have better adaptability features, and (2) provide richer social experience, than SIPAs developed using Xipho. We measure social experience via norm compliance and sanction proportion measures. We find that SIPAs engineered using Arnor have greater adaptability correctness, similar norm compliance, and are prone to lesser sanctions.

Chapter 2 details Arnor, and discusses its evaluation.

1.4.2 Poros: Enhancing Social Experience via Context Sharing

As SIPAs act and interact, they need to be aware of their stakeholders' contexts, and how each stakeholder perceive their actions. To address the research question of sharing and reasoning about deviation contexts, we develop Poros. Poros is a framework which enables SIPAs to share deviation contexts with other agents, and provides SIPAs an ability to reason about contexts shared by other agents. This ability to share and reason about shared contexts assist SIPAs in learning contextually relevant norms, and thus help SIPAs act in a way that provides a pleasing social experience to their stakeholders. We evaluated Poros by social simulation experiments.

Simulation Experiment

We hypothesize that Poros SIPAs that share and reason about context learn contextually relevant norms, and thus provide a greater social experience than those that don't reason about context. We measure social experience through happiness and experience payoff metrics (defined in Chapter 3), and find that Poros SIPAs provide higher happiness, and experience payoff.

Chapter 3 describes Poros, and the simulation experiments.

1.4.3 Research plan for Valar and Gimli: Enriching Social Experience with Values

Goal.

The goal is to enhance a SIPA's reasoning capability by making them aware of values each of their actions promote or demote.

Research question.

Does the awareness of *values* each action could promote or demote assist a SIPA in enriching the social experience it delivers to its stakeholders?

Background and Motivation.

Each action the Poros SIPAs execute promote or demote certain *values* [Pasotti et al., 2016]. For instance, a callee's action of answering an urgent phone call during a meeting may promote *safety* of the caller, but demote *privacy* of the meeting attendees'. To reason about such values, a SIPA could employ a value-based argumentation framework [Bench-Capon, 2003]. Being aware of values, and an ability to reason about these values could help a SIPA make an informed decision which aligns with its stakeholders' preferences, and thus essentially promote greater social experience in situations where norms conflict.

Hypothesis.

We hypothesize that a SIPA with a capability to reason about values each of its action promotes or demotes, provides a richer social experience than a SIPA without such a capability.

Research plan.

We plan to develop Valar, a framework that makes SIPAs aware of the values each of their actions could promote or demote. We envisage Gimli to work in tandem with Valar, and provide a decision-support system to SIPAs.

Valar. We will conduct a crowdsourced experiment wherein crowdworkers will answer questions in an immersive setting indicating their preferences for different available actions in situations where norms conflict, and the values each of these actions may promote or demote.

From the gathered data, we intend to identify (1) contextual factors (such as place and social relationships), and associated values (such as privacy, security, and trust) that could potentially influence decision making in a SIPA, (2) relative preference between the combinations of identified contextual factors and values, and (3) the tradeoffs between the contextual factors and values pairs.

To evaluate the proposed hypothesis, we will build different models based on the identified factors and values, and measure their goodness of fit.

Gimli. We will utilize the gathered data from the crowdsourcing experiment to identify the features (contextual factors and values) that influence the decision making in situations where norms conflict, and build classifiers to recommend actions. To evaluate Gimli, we will compute the recommender's precision, recall, and accuracy measures by comparing the recommendation with the actual crowdsourced response.

Success Criteria.

Valar. Goodness of fit for models containing values are high.

Gimli. Accuracy of the recommendations generated are high.

1.4.4 Status and Tentative Timeline

Table 1.1 shows the status of our contributions, and a tentative timeline for completion.

Table 1.1 Proposed plan.

	Task	Status	Timeline
1.	Arnor	Complete	
2.	Poros	Almost Complete	Aug 2017–Sep 2017
3a.	Valar	Ideation	Sep 2017–May 2018
3b.	Gimli	Ideation	Sep 2017–May 2018

CHAPTER

2

MODELING SOCIAL INTELLIGENCE VIA NORMS TO ENGINEER PRIVACY-AWARE PERSONAL AGENTS

2.1 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 [Chopra and Singh, 2016; Kafalı et al., 2016].

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.

To understand the nuances in modeling social intelligence in SIPAs, let us revisit the example in Chapter 1.

Example 2 *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 3 *Bob calls Alice when she is in an important meeting with Charlie and Dave.*

In Example 3, should Alice's phone ring loud during the meeting, privacy implications may follow [Murukannaiah et al., 2016; Solove, 2006]. 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 4 *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 [Bresciani et al., 2004] actor models and Gaia [Wooldridge et al., 2000] 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 [Alechina et al., 2016; Criado and Such, 2016]. Van Riemsdijk [van Riemsdijk et al., 2015b] 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 [Andrighetto et al., 2013]. 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 [Singh, 2013] model of (social) norms to capture social requirements, and incorporates argumentation constructs [Bench-Capon and Dunne, 2007] 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 [Sen and Airiau, 2007]) styles.

Section 2.2 briefly describes the background works on which we build. Section 2.3 describes Arnor in detail. Section 2.4 describes our developer and simulation studies, and Section 2.5 presents our results. Section 2.6 discusses related work, threats to validity, and concludes with important future directions.

2.2 Background

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

2.2.1 Tropos and Xipho

Tropos [Bresciani et al., 2004] 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 2.1 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.

Xipho [Murukannaiah and Singh, 2014] 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.

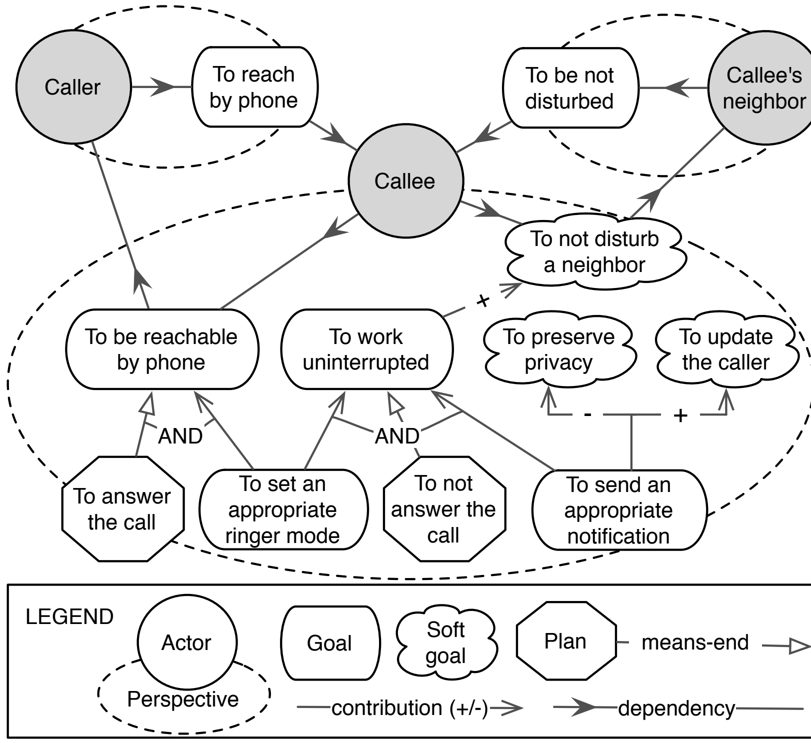

Figure 2.1 A Tropos system-as-is model of the ringer manager, expanding the callee's perspective [Murukannaiah and Singh, 2014].

2.2.2 Norms and Sanctions

A norm as understood here [Singh, 2013] 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:

$$\mathsf{N}(\textsc{subject}, \textsc{object}, \mathit{antecedent}, \mathit{consequent})$$

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 [Nardin et al., 2016]. 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* ∧ *ring* = *loud*, feedback = *scowl*).

2.3 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 2.2 shows a conceptual model of Arnor. Table 2.1 provides an overview.

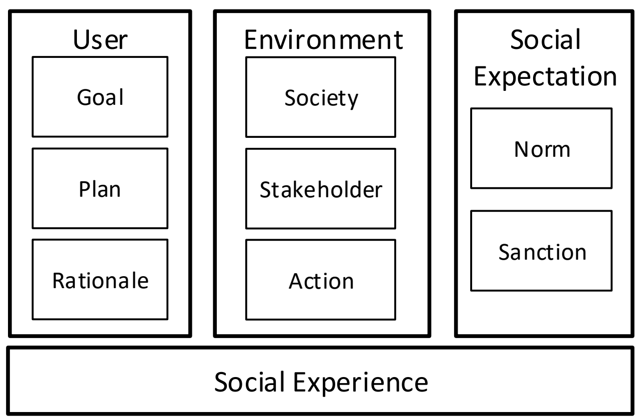

Figure 2.2 Arnor's conceptual model schematically.

2.3.1 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 2.1, we create the goal model for the ringer manager SIPA described in Examples 2–4 and Figure 2.1.

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.

2.3.2 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, 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.

2.3.3 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}: $\mathsf{C}(\text{PHONE-USER}, \text{CALLER}, \text{call},$
  $\text{ring} = \textit{silent} \wedge \textit{notify})$
- 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.
  $C_{meeting}$: $\mathsf{C}(\text{PHONE-USER}, \text{COWORKERS}, \text{call},$
  $\text{ring} = \textit{silent} \vee \text{ring} = \textit{vibrate})$

Sanctions. The associated sanctions are as below:

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

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.,
$\textit{conflict}(C_{caller}, C_{meeting})$.

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*).

2.3.4 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 [Bench-Capon and Dunne, 2007], and make use of argumentation schemes such as *arguments from consequences*, and *arguments from popular opinion* [Walton et al., 2008]. 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.

2.4 Evaluation

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

2.4.1 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 2–4 and Figure 2.1.

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.

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.

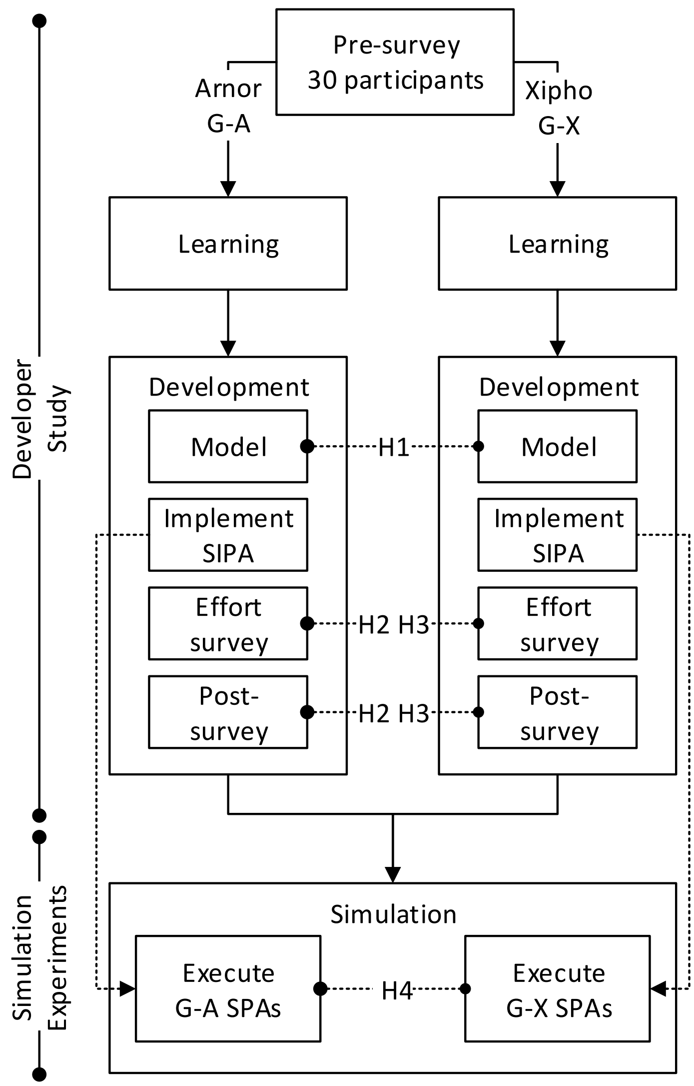

Figure 2.3 Experimental Design.

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.

2.4.2 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.

2.5 Results

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

2.5.1 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 2.4 and 2.5 shows the boxplots for time and effort expended by participants using Arnor and Xipho to develop the social ringer SIPA.

Difficulty of development. The participants using Arnor found it easier to develop SIPAs with Arnor, compared to participants using Xipho. Figure 2.6 shows the difficulty of development boxplots.

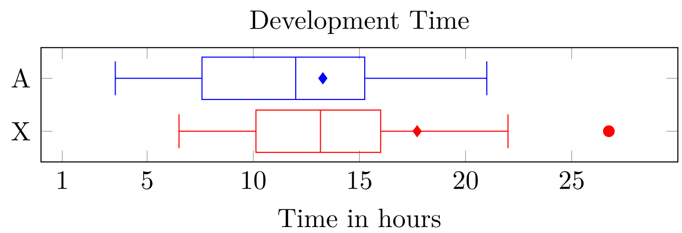

Figure 2.4 Arnor vs. Xipho's development time in hours as reported in the work session surveys.

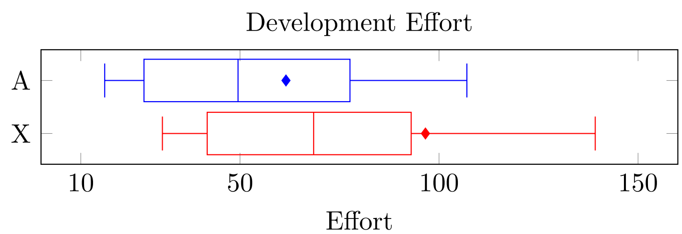

Figure 2.5 Arnor vs. Xipho's development effort as reported in the work session surveys.

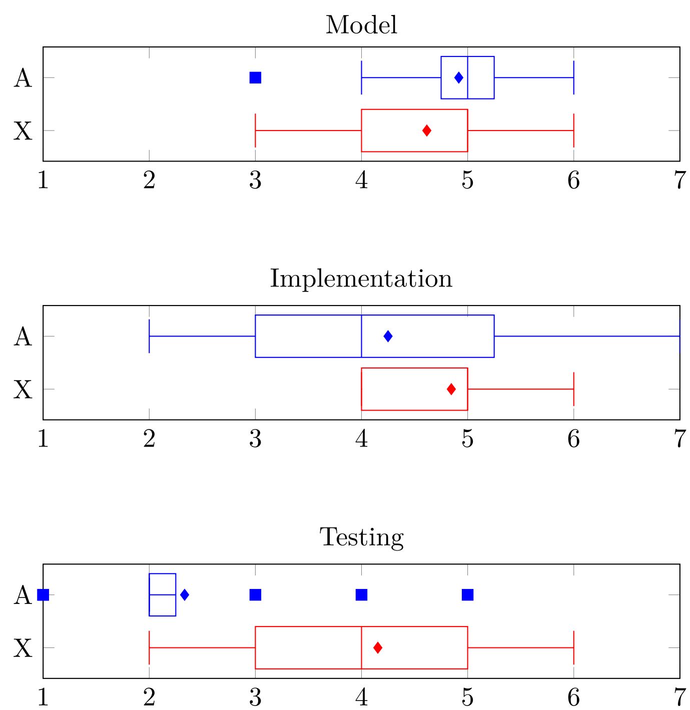

Figure 2.6 Arnor vs. Xipho's difficulty of development on a Likert scale of 1 (very easy) to 7 (very difficult).

2.5.2 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.

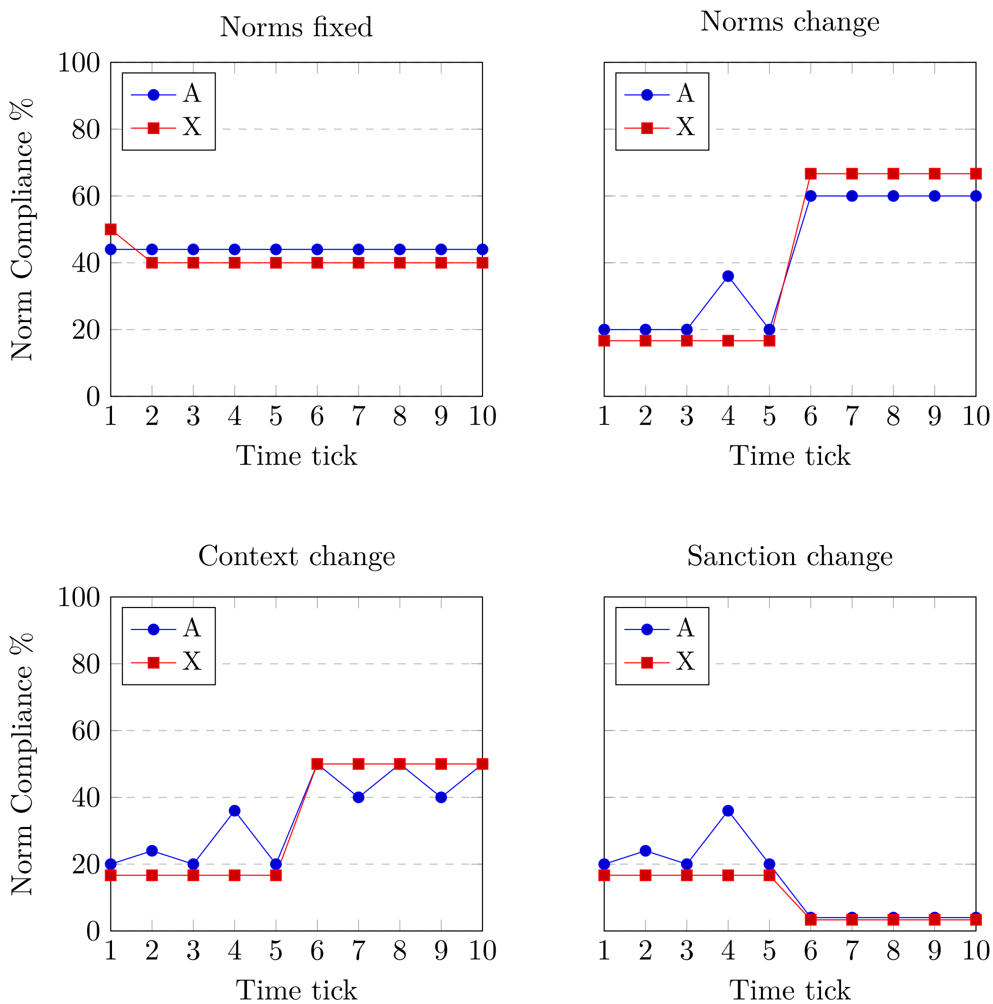

Figure 2.7 Arnor vs. Xipho's norm compliance.

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 2.7 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 2.8 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

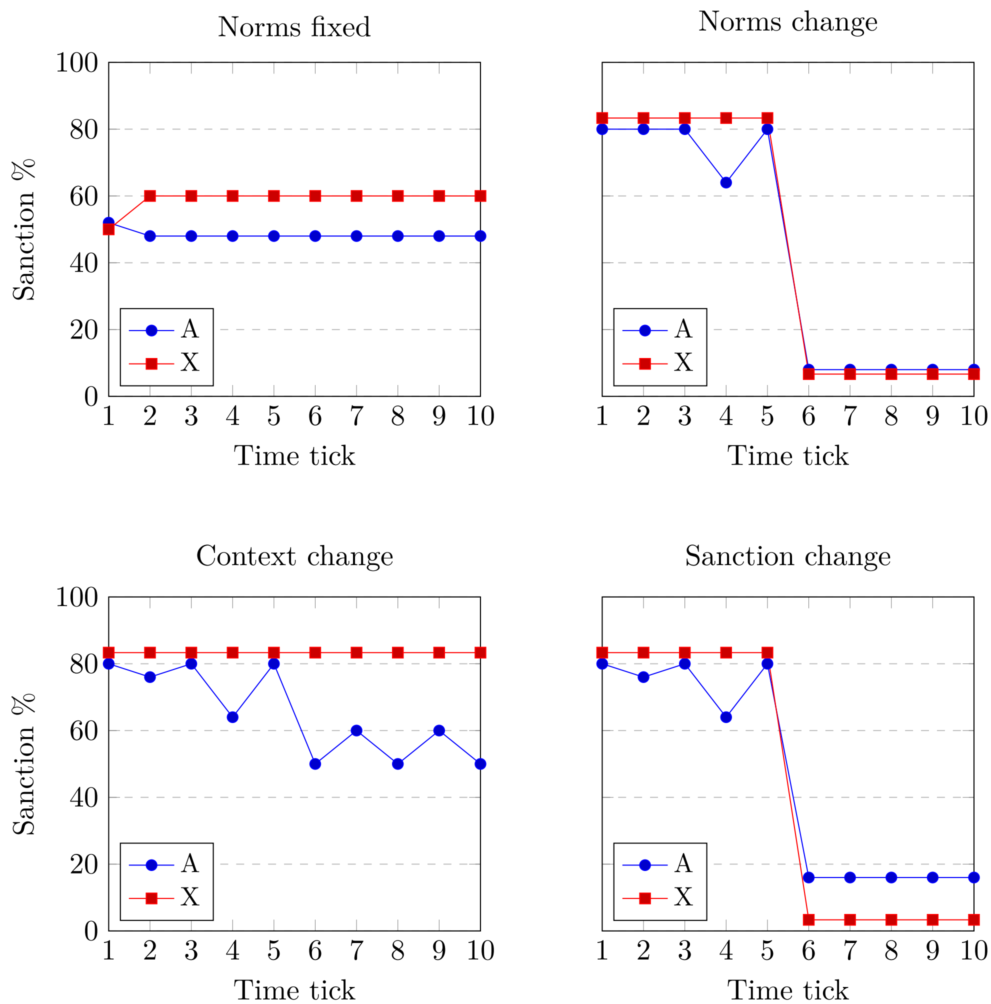

Figure 2.8 Arnor vs. Xipho's sanction proportion.

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.

2.5.3 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.

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.

2.6 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.

2.6.1 Related Works

Ali et al. [2013] 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. [2006] 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 [2011] attempt to resolve conflicts through reinforcement learning. Mashayekhi et al. [2016] propose a hybrid mechanism to monitor interactions and recommend norms to resolve conflicts. Mihaylov et al. [2014] study convergence and propose a decentralized approach based on strategies in game theory. Villatoro et al. [2013] introduce social instruments to facilitate norm emergence via social learning. Yu et al. [2013] 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. [2016] 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. [2015a] 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. [Aldewereld et al., 2016] present a formalism for group norms, and provide mechanisms to reason about these norms. Ajmeri et al. [2016] 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.

2.6.2 Future Directions

Ferreira et al. [2013] propose a computational model for emotional agents that considers norms, social relations, roles and socially acceptable behaviors in a given context. Sollenberger and Singh [2011] introduce Kokomo to develop affective applications, and provide a middleware for building such applications. Incorporating an affective [Sollenberger and Singh, 2011] 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.

Fogués et al. [2017] 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.

Table 2.1 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 telereachable*, 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 *telereachable* 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 *telereachable* 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	

CHAPTER

3

ENHANCING SOCIAL EXPERIENCE IN NORMATIVE AGENTS VIA CONTEXT SHARING

3.1 Introduction

Social *norms* provide a robust means to regulate interactions in a human society. Our everyday actions tend to *comply* with one or more social norms. For example, *not answering a phone call during a meeting* and *not talking loudly in a public library* are typically expected behaviors that accord with social norms. However, under certain circumstances, we *deviate* from the expected behavior and act not in accordance with the social norms of the society. For instance, *stepping out of a meeting to answer a phone call* may deviate from a norm. The ability to deviate from norms is important—it is what makes us autonomous. We may *sanction* each other based on how we are interacting. In particular, negative sanctions in response to deviations are a means for establishing norms. Such sanctions serve as the key basis for having a norm in the first place. For example, when a meeting attendee's phone rings loudly, *scowl* on other attendees' faces is a hint toward a norm of *keeping one's phone on silent during meetings*.

When we deviate from a norm, we may offer an explanation, typically describing the circumstance or *context* in which we deviated. First, revealing context may soften the burden of a deviation and help us avert negative sanctions. Suppose Alice reveals to meeting attendees that a call she received was from a sick family member. Here, the meeting attendees may not negatively sanction Alice. A deviation from a norm may also result in a positive sanction. For instance, a physician who reveals a patient's private data to save the patient's life would receive a positive sanction for violating a privacy norm. Whether a deviation leads to a positive or negative sanction largely depends on how others perceive the context of the deviation.

Second, context may help refine the underlying norms from which we might be deviating. For example, Alice's revelation may help refine the norm from *not answering a phone call during a meeting* to *not answering a phone call during a meeting, unless the call is urgent*. In essence, the contexts associated with deviations and any ensuing sanctions help characterize the boundaries of the norms in play.

We seek to engineer an artificial agent society in which personal agents [Murukannaiah and Singh, 2014] act and interact on behalf of human users. We imagine that personal agents intend to respect social norms. Tambe et al. [2008] note from their experience of deploying personal agents that complying with social norms is critical in providing a rich experience to users. In their vision of socially adaptive agents, van Riemsdijk et al. [2015b] argue for the need of personal agents to learn norms from experience and reason about norms at runtime. We adapt Ajmeri et al.'s [Ajmeri et al., 2017] term, socially intelligent personal agents (SIPAs), for agents who act in accordance with social norms.

A SIPA is autonomous and may deviate from norms. A distinguishing feature of SIPAs is that when a SIPA deviates from a norm, it shares the then context with (one or more) other SIPAs. We conjecture that sharing context promotes SIPAs to deliver a rich social experience. In this regard, we consider two research questions:

RQ 1. Does a SIPA's ability to reason about shared context help the SIPA in learning contextually relevant social norms?

RQ 2. Does a SIPA's ability to learn contextually relevant social norms help the SIPA in delivering a rich social experience to its users?

Contribution.

To answer the research questions above, we conceptualize Poros, a framework wherein SIPAs learn social norms via sharing context. Poros includes a conceptual model and mechanisms through which SIPAs interact and learn norms. We evaluate Poros via a social simulation inspired by real scenarios. Our results show that SIPAs in Poros that share context more accurately learn relevant social norms and provide a better social experience than SIPAs that learn norms solely via sanctions.

Organization.

Section 3.2 reviews related works. Section 3.3 describes Poros. Section 3.4 details our social simulation. Section 3.5 describes the simulation experiments and results. Section 3.6 concludes with a discussion and future directions.

3.2 Related Work

3.2.1 Normative Systems

Researchers have endeavored to design normative systems. Singh [2013] proposes a computational representation of norms to engineer governance in sociotechnical systems (STSs). We adopt this framework, in which norms are classified as five types: commitment, authorization, prohibition, sanction and power. A norm is directed from a subject (stakeholder) to an object (stakeholder), 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). Chopra and Singh [2016] introduce Interaction-Oriented Software Engineering (IOSE) as a paradigm expressly suited to capturing the social basis of STSs via norms, in order to compensate the lack of support of social processes in existing approaches. Hao et al. [2016] propose a formal method to design normative systems. Ajmeri et al. [2017] propose Arnor,an agent-oriented software engineering method to model social intelligence in personal agents. They suggest that personal agents which can understand and learn from the intricacies of social norms, deviations and associated arguments provide a rich social experience to their users. These works provide worthy insights from a SIPA modeling standpoint. Poros is novel in the way it helps SIPAs learn social norms by sharing and reasoning about deviation contexts.

3.2.2 Other Concepts

3.2.2.1 Norm Conflicts and Compliance

In normative systems, agents are autonomous. They "can decide whether to follow the explicitly represented norms, and the normative systems specify how and in which extent the agents can modify the norms [Boella et al., 2006]." Agents may face the conflicts of multiple applicable norms, or conflicts between norms and their own goals. López et al. [2006] proposes a formal framework that describes how agents reason about norms in a normative system. They discuss the reason that agents comply with norms and how they work to satisfy their own goals. van Riemsdijk et al. [2015a] develop a norm compliance framework to design socially adaptive agents. Aldewereld et al. [2016] present a formalism and mechanism to comply with group norms.

Sugawara [2011] uses reinforcement learning to resolve norm conflicts. They give insights into how agents make decisions in our framework.

3.2.2.2 Norm Emergence and Evolution

Norms are dynamic and may emerge and evolve in the interactions of agents [Savarimuthu et al., 2009]. Boella et al. [2009a] propose a normative framework to evaluate and classify normative system change. We take these insights into consideration during our simulation. We observe the evolution of norms and reason about the different changes of norms in variant scenarios. Mashayekhi et al. [2016] propose Silk, a hybrid mechanism for norm emergence and conflict resolution in sociotechnical systems. Villatoro et al. [2013] present social instruments as a way to assist norm emergence. Yu et al. [2013] suggest using collective learning compared to pairwise learning for norm emergence. Poros differs from these works in supporting sharing and reasoning about contextual information to facilitate learning of contextually relevant norms.

3.2.2.3 Deviation and Sanctions

As aforementioned, an agent in a normative system can decide whether to comply with or deviate from a norm. The notion of sanction, which can either be negative or positive, is associated with the reaction of other agents to this decision. Nardin et al. [2016] develop a sanction typology and introduce a conceptual sanctioning process model to promote governance in STSs. Previous works adopt sanctions as a way to promote norm compliance. Recent works explore combining norm communication with sanctions to promote cooperation [Andrighetto et al., 2013]. van Riemsdijk et al. [2015b] emphasize the understanding of norm violation as an important challenge for designing socially adaptive agents. To understand a deviation, it is necessary to understand the context in which a deviation occurs, a problem that Poros addresses.

3.2.3 Context and Context Sharing

Ali et al. [2013] emphasize on context in software engineering and analyze the detection of inconsistency and conflicts in contextual goal models. Software systems have contextual requirements and their actions to reach them can in turn change the context. In Xipho, Murukannaiah and Singh [2014] state the importance of context in designing personal agents. Ajmeri et al. [2017] extend Xipho with social norms. Personal agents are subject to different norms in different context, so they have to be context-aware to be able to reason about norms. Poros examines the effect of context sharing among agents after norm deviations. However, context sharing could raise privacy or security concerns, such as those in location-based services. We focus on the improved social experience of context sharing.

3.3 The Poros Framework

Poros comprises of a conceptual model and mechanisms via which SIPAs interact.

3.3.1 Conceptual Model

Figure 3.1 shows a conceptual model of a society Poros aims to engineer. The society consists of stakeholders, a social structure, and SIPAs acting on behalf of stakeholders.

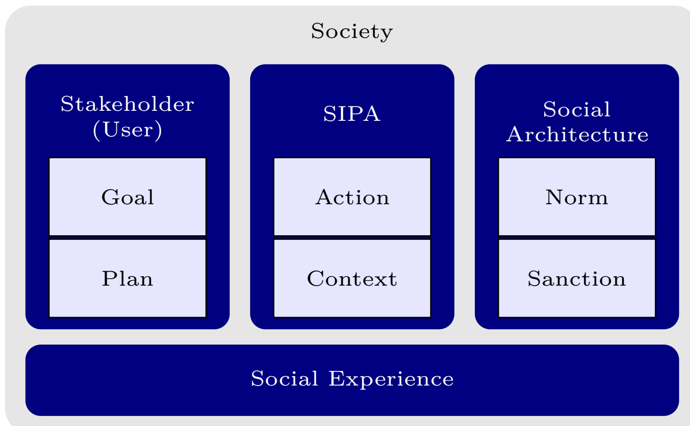

Figure 3.1 A conceptual model of the Poros society.

The stakeholders are users. Each stakeholder has goals and plans.

- A *goal* is a set of desirable states. A goal is associated with one or more stakeholder; each stakeholder may have multiple goals.
- A *plan* is a set of actions that can bring about a goal. A plan is associated with a user and potentially helps bring about one or more goals of that user.

The social architecture of a society consists of its norms and the sanctions that ensure compliance and noncompliance of norms.

- A *norm* is a tuple of {subject, object, antecedent, consequent} [Singh, 2013]. Norms characterize the social architecture that promotes prosocial behavior.
- A *sanction* is a set of actions a stakeholder may take for deviation of a norm by another stakeholder. Depending on how agents perceive deviation, it results in *sanctions* that are positive or negative [Nardin et al., 2016].

A SIPA acts and interacts on behalf of a stakeholder and is aware of the social architecture of the society.

- An *action* is a step a SIPA takes to execute its stakeholder's plan, thereby bringing about the corresponding goal. An action may satisfy or violate a norm. SIPAs in the society can observe each other's actions.
- A *context* is the circumstance under which a SIPA takes an action [Dey, 2001]. Thus, a context may characterize the circumstance in which a norm is satisfied or violated. A SIPA may share the context in which it takes an action with other SIPAs.

Experience captures which or how many goals of its stakeholders a SIPA helps to bring about.

Social experience captures which goals that relate to norms are promoted by a SIPA. It also relates to how a SIPA's stakeholders perceive a norm satisfaction or violation, and the sanctions they apply. Our objective is to promote each SIPA's actions that maximize the social experience.

3.3.2 Interaction and Learning

Algorithm 1 shows how SIPAs interact and learn in Poros. Further, each SIPA, to bring about its goals, identifies plans and an associated set of actions for each goal. If multiple plans can achieve a goal, the SIPA selects the plan that yields the larger social experience (based on its history), or the first viable plan. The actions a SIPA performs as part of a plan are observed by all other SIPAs. As SIPAs execute plans, they share the associated context with other SIPAs.

When other SIPAs observe a SIPA's actions and receive its shared context, they sanction it. Each SIPA stores the goals, plans, associated context, and sanctions received in its history to facilitate future decision making.

3.3.3 Example SIPA: Ringer

We demonstrate Poros using an example SIPA called RINGER, whose conceptual model Figure 2.1 shows [Murukannaiah and Singh, 2014]. The *callee*, a stakeholder of RINGER, has the goals of *being reachable by phone*, *to work uninterrupted*, *to not disturb neighbors* and *to preserve privacy*. The *caller*, another stakeholder, has the goal *to reach the callee*. The *neighbor*, also a stakeholder, has a goal *to not be disturbed*. To provide a rich social experience to its stakeholders, the SIPA should be aware of its context, stakeholders' goals, associated plans, and the applicable norms.

Following Algorithm 1, the RINGER starts interacting with other agents, considering a fixed set of norms, such as *never answer a call during a meeting* or *always answer an urgent call from a family member*. As it acts and interacts with its users, it learns contextually relevant norms and appropriate actions.

Algorithm 2 describes decision making for an incoming phone call for a RINGER that has history. Here the *context* includes the caller-callee relationship, call urgency (casual or urgent),

Algorithm 1: Interaction and learning in Poros.

Input: N: norms
Input: G: goal
Input: P: plans
Input: c: context

$p_{exec} = \emptyset$;
$e_{max} = 0$;
foreach $p_i \in P \vdash (G \mid c)$ **do**
    h = hasHistory(G, p, c);
    if h **then**
        exp_i = predictExperience(p_i, c, h);
        if $exp_i > e_{max}$ **then**
            $exp_{max} = exp_i$;
            $p_{exec} = p_i$;
        end
    else
        $p_{exec} = p_i$;
    end
end
Perform plan p_{exec};
Share context $\{p_{exec}, c\}$;
Receive sanctions $\{s\}$;
Add history $\{g, p_{exec}, c, s\}$;

the place where the callee is, and the neighbor-callee relationship. Possible plans include answering or ignoring the incoming phone call. To make a choice, the RINGER predicts the experience for the two plans from its history and chooses the action that provides the higher social experience. Note that, to predict experience for each action, SIPA could employ any machine learning technique, such as linear regression, on the accumulated history. But machine learning is not the core focus here.

As the *callee*'s RINGER chooses an action, it follows Algorithm 3 to share the context with all *neighbors* and the *caller*. A SIPA identifies the norms that are satisfied or violated, and provides contextual arguments in favor of and against why it chose its action given norm satisfaction and violation. Ideally, RINGER should selectively reveal contexts to other stakeholders according to its goals and its human user's privacy attitude. However, for simplicity, we have RINGER share context with all stakeholders. The sharing of contexts can lead to additional benefits such as increase in trust between users. We do not model those in the current work.

When a RINGER instance receives the context shared by another RINGER instance, the receiver reasons about context as detailed in Algorithm 4. Consider the *caller*'s RINGER, for

Algorithm 2: $a \leftarrow \text{selectAction}(r_c,\ u,\ l,\ r_n,\ h)$.

Input: r_c: caller-callee relationship
Input: u: call urgency
Input: l: place
Input: r_n: neighbor-callee relationship
Input: h: History
Output: a: action

$exp_{answer} = \text{predictExperience}(h, \text{Answer}, r_c, l, r_n)$;
$exp_{ignore} = \text{predictExperience}(h, \text{Ignore}, r_c, l, r_n)$;
if $exp_{answer} > exp_{ignore}$ **then**
  $a = \text{Answer}$
else
  $a = \text{Ignore}$
end

example. When *callee*'s RINGER shares the context with the *caller*'s RINGER, the latter predicts an action according to its history of incoming calls. If the *caller*'s predicted action based on its history matches the *callee*'s observed action, the *caller* positively sanctions the *callee*; otherwise the *caller* perceives the *callee* as a deviant, and negatively sanctions it. The *neighbor*'s RINGER sanctions the *callee* similarly.

3.4 Simulation Model

We evaluate Poros via a simulation model. We use MASON [Luke et al., 2005] to build a simulation environment, henceforth the *ringer environment*, inspired from real-life settings where the RINGER can interact.

3.4.1 The Ringer Environment

The ringer environment contains shared places, as Figure 3.2 shows. Each place is a location such as home, party, meeting, library, and emergency room understood in conceptual terms [Murukannaiah and Singh, 2012]. Corresponding to each place, we define social circles such as family, friends, and colleagues. Each agent belongs to one or more of these circles. The agents keep moving from place to place. Network topology is random but each agent is part of three circles: friend, family and colleague.

Actions.

The agents in the ringer environment perform the following actions depending upon their roles:

Algorithm 3: $c \leftarrow \text{shareContext}(N,\ a,\ r_c,\ u,\ l,\ r_n)$.

Input: N: norms
Input: a: action
Input: r_c: caller-callee relationship
Input: u: call urgency
Input: l: place
Input: r_n: neighbor-callee relationship
Output: c: context arguments

foreach $n_i \in N$ **do**
  if $isSatisfied(n_i, a)$ **then**
    Add ContextArgInFav $\leftarrow (n_i,\ a,\ r_c,\ l,\ r_n)$;
  end
  if $isViolated(n_i, a)$ **then**
    Add ContextArgInOpp $\leftarrow (n_i,\ a,\ r_c,\ l,\ r_n)$;
  end
end
if $ContextArgInFav \neq \emptyset \vee ContextArgInOpp \neq \emptyset$ **then**
  $c = \{\text{ContextArgInFav, ContextArgInOpp}\}$;
end

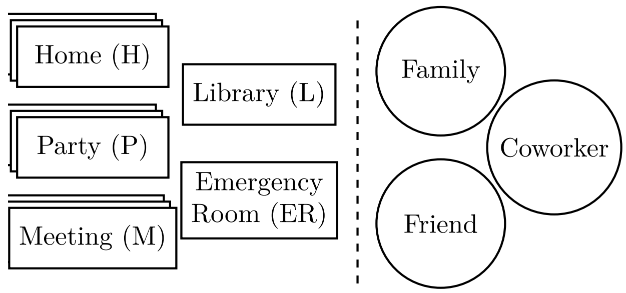

Figure 3.2 Places and social circles in the ringer environment.

- A caller initiates an urgent or a casual phone call.
- A callee answers or ignores a phone call.
- A caller and neighbors respectively sanction a callee for answering or ignoring phone calls.
- A callee shares context for answering or ignoring phone calls.
- A caller and a neighbor reason about context.

Norms.

Each place and each circle has predefined norms, as defined in Table 3.1.

Algorithm 4: $s \leftarrow$ reasonAboutContext(a, r_c, u, l, r_n, h).

Input: a: action
Input: r_c: caller-callee relationship
Input: u: call urgency
Input: l: place
Input: r_n: neighbor-callee relationship
Input: h: History
Output: s: sanction

a_e = selectAction(r_c, u, l, r_n, h);
if $a = a_e$ **then**
    $s = s_p$: positive sanction;
else
    $s = s_n$: negative sanction;
end

Table 3.1 Norms for answering calls based on the place and the caller's social circle.

Place	**Answer**
Home (H)	✓
Party (P)	✓
Meeting (M)	✗
Library (L)	✗
Emergency (ER)	✓

Circle	**Call Type**	
	Casual	Urgent
Family	✓	✓
Friend	✓	✓
Coworker	✓	✓
Stranger	✗	✓

Payoffs.

For each phone call, based on the callee's action of answering or not answering, the caller, callee, and neighbors perceive a fixed payoff as shown in Tables 3.2, 3.3 and 3.4.

Table 3.2 Payoff for the callee.

Caller's Relation	**Callee's Response**	**Call Type**	
		Casual	Urgent
Family, Friend, or Coworker	✓	0.50	1.00
	✗	0.00	–0.50
Stranger	✓	0.00	0.50
	✗	0.25	–0.25

Table 3.3 Payoff for the caller.

Callee's Response	**Call Type**	
	Casual	Urgent
✓	0.50	1.00
✗	–0.50	–1.00

Table 3.4 Payoff for the neighbors.

Callee's Response	**Place**				
	H	P	M	L	ER
✓	0.67	–0.33	–1.00	–1.00	1.00
✗	–0.33	0.67	1.00	1.00	–1.00

3.4.2 Agent Types

Fixed agents act according to the fixed set of norms listed in Table 3.1. If the norms conflict, the agents toss a fair coin to choose between alternative actions. If Fixed agents perceive an action as deviation, they sanction the deviant.

Sanctioning agents learn social norms from sanctions [Andrighetto et al., 2013]. In the ringer environment, these agents start as Fixed agents. They continue to record the call history, and associated sanctions. Once they have gained confidence about their history of sanctions, they base their actions on the learning based on history. That is, as callees, when norms conflict, they select the action that provides a higher payoff, computed according to Tables 3.2–3.4. As callers and neighbors, these agents sanction callees as per fixed norms listed in Table 3.1.

Poros agents learn social norms by sharing and reasoning about context. In the ringer environment, Poros agents start as Fixed agents following fixed norms listed in Table 3.1. As callees, they share context, i.e., reveal the caller's relationship and the call's urgency, to their neighbors, and reveal their place and their neighbors' relationships to the caller. As neighbors or callers, they first understand a callee's context and decide what action they would have performed were they in the same situation, and sanction accordingly. Poros agents use payoffs as listed in Table 3.5.

In the simulation, we employ a linear regression model over history to predict sanctions by stakeholders, and thus social experience.

Table 3.5 Payoff for the neighbors based on reasoning about the shared context by the callee.

Callee's Response	**Neighbor's Expectation**	**Place**				
		H	M	ER	P	L
✓	✓	0.67	1	1	0.67	1
✓	✗	–0.33	–1	–1	–0.33	–1
✗	✓	–0.33	–1	–1	–0.33	–1
✗	✗	0.67	1	1	0.67	1

3.5 Experiments and Results

We evaluate our research questions via three experiments in which we simulate the agents described in Section 3.4.2 on the ringer environment. We run each simulation for $3,000$ steps and compute the following metrics.

Happiness measures the proportion of agents that perceive actions as norm compliant.

Experience-payoff measures the social experience delivered by an agent. It is computed by aggregating payoffs for all stakeholders according to the payoff Tables 3.2, 3.3, 3.4 and 3.5.

To answer *RQ 1*, we consider the following hypotheses:

H1$_{alt}$ Poros agents yield *happiness* greater than that yielded by Fixed agents.
H1$_{null}$ Poros agents yield no significant gain in *happiness* over Fixed agents.
H2$_{alt}$ Poros agents yield *happiness* greater than that yielded by Sanctioning agents.
H2$_{null}$ Poros agents yield no significant gain in *happiness* over Sanctioning agents.

To answer *RQ 2*, we consider the following hypotheses:

H3$_{alt}$ Poros agents yield *social experience* greater than that yield by Fixed agents.
H3$_{null}$ Poros agents yield no significant gain in *social experience* over Fixed agents.
H4$_{alt}$ Poros agents yield *social experience* greater than that yield by Sanctioning agents.

H4$_{null}$ Poros agents yield no significant gain in *social experience* over Sanctioning agents.

To test the significance of the hypotheses, we apply the common two-tailed paired t-test.

3.5.1 Pragmatic Agents and Varying Network Size

We simulate agents described in Section 3.4.2 on four types of network, specifically, large and small networks with dense or sparse connectivity as Table 3.6 describes. The agents in this experiment are pragmatic, and try to achieve a high average experience-payoff for all callees, callers, and neighbors. We now describe our results.

Table 3.6 Characteristics of network types studied.

Network Type	Agents	Circles		
		Home	Meeting	Party
Large-Dense	1,000	20	20	20
Large-Sparse	1,000	100	100	100
Small-Dense	250	5	5	5
Small-Sparse	250	25	25	25

Fixed agents. behave according to the fixed norms, their behaviors do not change over time. As expected, we observe no significant change in social experience throughout the simulation. The average social experience was found to be between 0.53 and 0.56, and the happiness to be about 52% for the four network types.

Sanctioning agents. Sanctioning agents first learn and then act as they have learned. As expected, at around step $1,000$ we see a rise in social experience offered by Sanctioning compared to Fixed agents. The rise is gradual as the agents start to learn. For the first 200 steps, the average social experience is the same as Fixed agents. It later stabilizes between 1.11 and 1.21 for all four networks. The happiness values were between 61.2 and 63.7%.

Poros agents. Poros agents first learn, and then act according to the learning. At around step $1,000$, as agents acquire confidence, we see a significant increase in social experience offered by Poros agents. It stabilizes between 2.14 and 2.19 for the different networks. Happiness was found be significantly higher between 82.0 and 83.2%. For the first 200 steps, Poros agents yield the same average social experience as Fixed and Sanctioning agents.

Happiness and experience payoffs offered by Poros agents are significantly greater than those offered by Fixed and Sanctioning agents; thus the four null hypotheses: H1$_{null}$, H2$_{null}$, H3$_{null}$, and H4$_{null}$, are rejected. Figure 3.3 show the experience payoff plots indicating the results are consistent across the four network types. Table 3.7 summarizes the findings of the experiment with pragmatic agents.

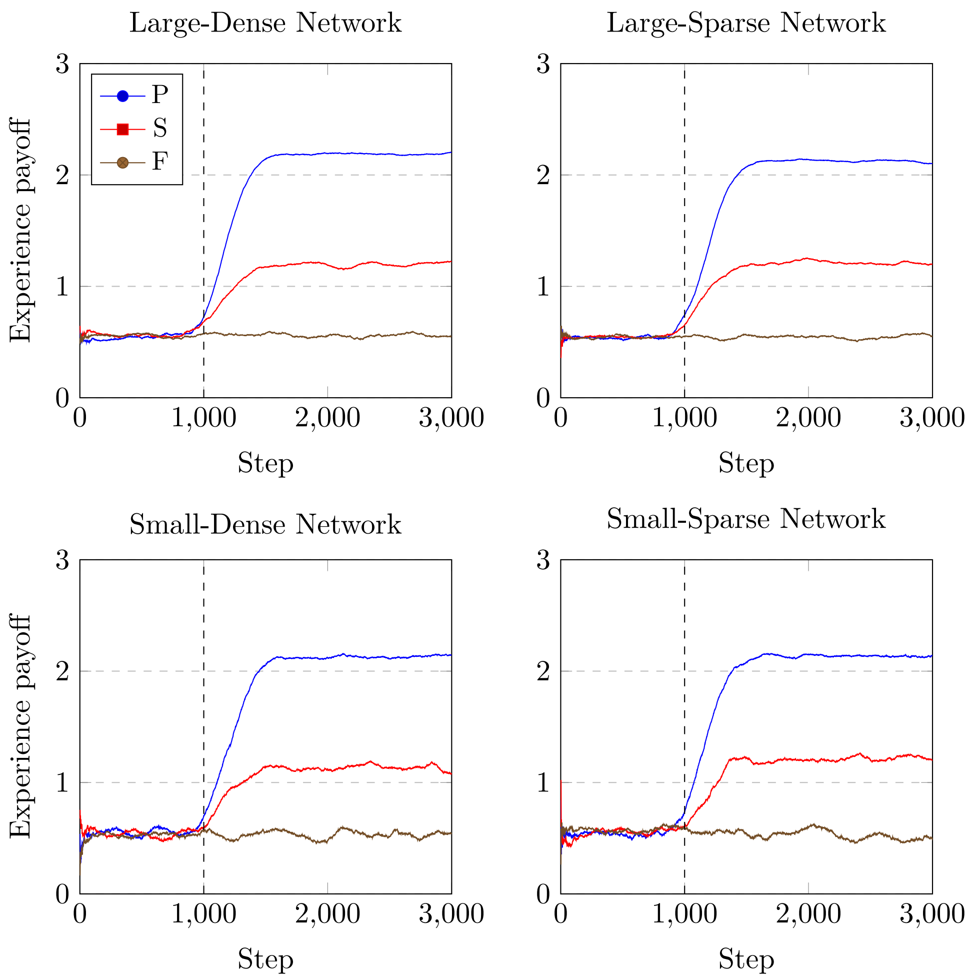

Figure 3.3 Experience payoff per phone call for a window size of 200 steps on different network sizes and densities.

3.5.2 Experiment with Considerate Agents

We experiment with considerate agents who consider payoffs only for their neighbors when deciding the actions to perform when norms conflict. These agents continue to sanction based on their history.

Table 3.7 Empirical results on the effectiveness of Poros agents.

Large-Dense

Agent Type	Experience$^{\#}$	Happiness$^{\#\#}$	$p^{\#\#\#}$
Fixed	0.56	52.7%	< 0.01
Sanctioning	1.21	63.5%	< 0.01
Poros	2.19	83.2%	–

Large-Sparse

Agent Type	Experience$^{\#}$	Happiness$^{\#\#}$	$p^{\#\#\#}$
Fixed	0.55	52.5%	< 0.01
Sanctioning	1.21	63.5%	< 0.01
Poros	2.19	83.2%	–

Small-Dense

Agent Type	Experience$^{\#}$	Happiness$^{\#\#}$	$p^{\#\#\#}$
Fixed	0.53	52.1%	< 0.01
Sanctioning	1.11	61.2%	< 0.01
Poros	2.14	82.0%	–

Small-Sparse

Agent Type	Experience$^{\#}$	Happiness$^{\#\#}$	$p^{\#\#\#}$
Fixed	0.54	52.5%	< 0.01
Sanctioning	1.22	63.7%	< 0.01
Poros	2.14	82.1%	–

$^{\#}$ Stabilized value of social experience; $^{\#\#}$ Stabilized value of happiness; $^{\#\#\#}$ Two-tailed paired t-test

Results.

Figure 3.4 shows the experience payoffs for considerate Sanctioning and Poros agents in a Small-Dense network. The average experience payoff drops for Sanctioning and Poros agents after they have gained enough confidence. We attribute this decline to the fact that these agents value the neighbors' experience payoff more than their own, and thus start to ignore calls that they should have answered. Poros agents offer higher experience payoff than Sanctioning agents because the stakeholders give smaller negative sanctions when they reason about context. The results for the other three network types are similar.

3.5.3 Experiment with Selfish Agents

Selfish agents consider only their payoffs when performing actions. Selfish agents may not always sanction others. They sanction a deviant based on their history.

Results.

Figure 3.4 shows the experience payoff plot for selfish Fixed and Poros agents in a Small-Dense network. The plots are similar to those in the experiment with pragmatic agents, but with slightly lower stabilized values. Here, agents tend to answer all calls, which benefits both caller and callee most of the time. We observe similar results for the other three networks.

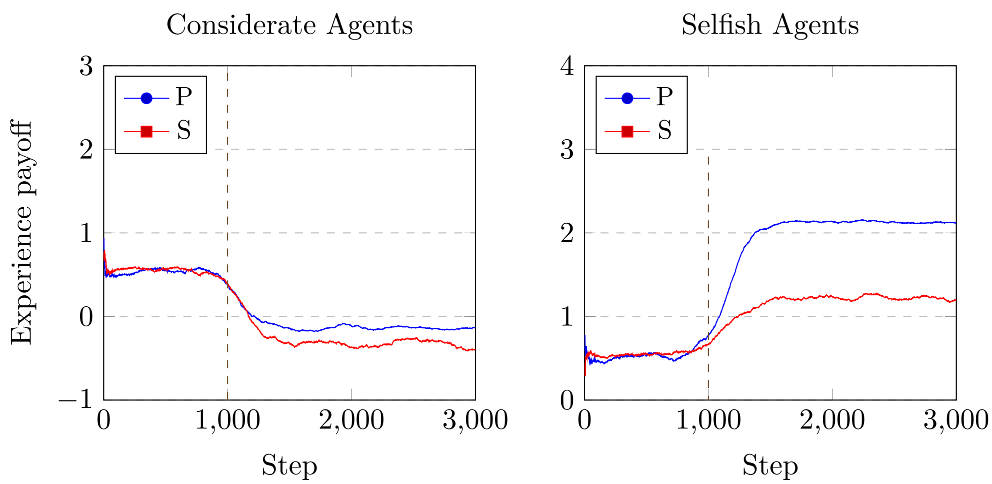

Figure 3.4 Experience payoffs (averaged over a window size of 200 steps) yield by considerate and selfish agents in a Small-Dense network.

3.6 Conclusion and Future Works

Poros is a framework in which SIPAs learn contextually relevant social norms by offering and reasoning shared contexts. Via a series of simulation experiments, we find that Poros agents accurately learn contextually norms compared to agents relying on fixed norms or learning norms via sanctions. In the experiments, we also compute social metrics and find that Poros agents deliver significantly higher happiness and experience payoff than other agents. These findings are stable under changes to network size and characteristics of agents.

Incorporating affect in relation to norms [Ferreira et al., 2013] is an interesting future direction. Another direction is to crowdsource data about real users' information sharing preferences and attitudes, and build machine learning techniques in a SIPA for recommending policies to its users.

Acknowledgments

We thank the US Department of Defense for support through the Science of Security Lablet at NC State University.

BIBLIOGRAPHY

Nirav Ajmeri, Jiaming Jiang, Rada Chirkova, Jon Doyle, and Munindar P. Singh. Coco: Runtime reasoning about conflicting commitments. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 17–23, New York, 2016.

Nirav Ajmeri, Pradeep K. Murukannaiah, Hui Guo, and Munindar P. Singh. Arnor: Modeling social intelligence via norms to engineer privacy-aware personal agents. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 230–238, São Paulo, May 2017. IFAAMAS.

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

Natasha Alechina, Joseph Y. Halpern, Ian A. Kash, and Brian Logan. Decentralised norm monitoring in open multi-agent systems: (extended abstract). In *Proceedings of the 15th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1399–1400, Singapore, 2016. IFAAMAS.

Raian Ali, Fabiano Dalpiaz, and Paolo Giorgini. Reasoning with contextual requirements: Detecting inconsistency and conflicts. *Information and Software Technology*, 55(1):35–57, 2013.

Giulia Andrighetto, Jordi Brandts, Rosaria Conte, Jordi Sabater-Mir, Hector Solaz, and Daniel Villatoro. Punish and voice: Punishment enhances cooperation when combined with norm-signalling. *PLoS ONE*, 8(6):1–8, 06 2013. doi: 10.1371/journal.pone.0064941.

Trevor J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, July 2007. ISSN 0004-3702.

Guido Boella, Leendert van der Torre, and Harko Verhagen. Introduction to normative multi-agent systems. *Computational & Mathematical Organization Theory*, 12(2):71–79, 2006. ISSN 1572-9346. doi: 10.1007/s10588-006-9537-7. URL `http://dx.doi.org/10.1007/s10588-006-9537-7`.

Guido Boella, Gabriella Pigozzi, and Leendert van der Torre. Normative framework for normative system change. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 169–176, Budapest, 2009a. International Foundation for Autonomous Agents and Multiagent Systems.

Guido Boella, Gabriella Pigozzi, and Leendert van der Torre. Normative systems in computer science - ten guidelines for normative multiagent systems. In Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, editors, *Normative Multi-Agent Systems*, number 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009b. Schloss Dagstuhl

- Leibniz-Zentrum fuer Informatik, Germany. URL http://drops.dagstuhl.de/opus/volltexte/2009/1902.

Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multiagent Systems*, 8(3):203–236, May 2004.

Amit K. Chopra and Munindar P. Singh. From social machines to social protocols: Software engineering foundations for sociotechnical systems. In *Proceedings of the 25th International World Wide Web Conference*, pages 903–914, Montréal, April 2016. ACM.

Natalia Criado and Jose M. Such. Selective norm monitoring. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 208–214, New York, 2016.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1): 4–7, January 2001. ISSN 1617-4909.

Nuno Ferreira, Samuel Mascarenhas, Ana Paiva, Gennaro Di Tosto, Frank Dignum, John McBreen, Nick Degens, Gert Jan Hofstede, Giulia Andrighetto, and Rosaria Conte. An agent model for the appraisal of normative events based in in-group and out-group relations. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1220–1226, Bellevue, 2013. AAAI Press.

Ricard López Fogués, Pradeep K. Murukannaiah, Jose M. Such, and Munindar P. Singh. Understanding sharing policies in multiparty scenarios: Incorporating context, preferences, and arguments in decision making. *ACM Transactions on Computer-Human Interaction*, 24(1): 5:1–5:29, March 2017.

Jianye Hao, Eunsuk Kang, Jun Sun, and Daniel Jackson. Designing minimal effective normative systems with the help of lightweight formal methods. In *Proceedings of the 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*, pages 50–60, Seattle, 2016. ACM.

Özgür Kafalı, Nirav Ajmeri, and Munindar P. Singh. Revani: Revising and verifying normative specifications for privacy. *IEEE Intelligent Systems*, 31(5):8–15, September 2016.

Fabiola López Y López, Michael Luck, and Mark D'Inverno. A normative framework for agent-based systems. *Comput. Math. Organ. Theory*, 12(2-3):227–250, October 2006. ISSN 1381-298X. doi: 10.1007/s10588-006-9545-7. URL http://dx.doi.org/10.1007/s10588-006-9545-7.

Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. *Simulation: Transactions of the Society for Modeling and Simulation International*, 81(7):517–527, July 2005.

Mehdi Mashayekhi, Hongying Du, George F. List, and Munindar P. Singh. Silk: A simulation study of regulating open normative multiagent systems. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 373–379, New York, 2016. AAAI Press.

John-Jules Ch. Meyer and Roel J. Wieringa, editors. *Deontic Logic in Computer Science: Normative System Specification*. Wiley, Chichester, United Kingdom, 1993.

Mihail Mihaylov, Karl Tuyls, and Ann Nowé. A decentralized approach for convention emergence in multi-agent systems. *Autonomous Agents and Multiagent Systems*, 28(5):749–778, 2014.

Pradeep K. Murukannaiah and Munindar P. Singh. Platys Social: Relating shared places and private social circles. *IEEE Internet Computing*, 16(3):53–59, May 2012.

Pradeep K. Murukannaiah and Munindar P. Singh. Xipho: Extending Tropos to engineer context-aware personal agents. In *Proceedings of the 14th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 309–316, Paris, May 2014. IFAAMAS.

Pradeep K. Murukannaiah, Nirav Ajmeri, and Munindar P. Singh. Engineering privacy in social applications. *IEEE Internet Computing*, 20(2):72–76, Mar 2016. ISSN 1089-7801. doi: 10.1109/MIC.2016.30.

Luis G. Nardin, Tina Balke-Visser, Nirav Ajmeri, Anup K. Kalia, Jaime S. Sichman, and Munindar P. Singh. Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *The Knowledge Engineering Review (KER)*, 31: 142–166, March 2016. ISSN 1469-8005. doi: 10.1017/S0269888916000023. URL `http://journals.cambridge.org/article_S0269888916000023`.

Pietro Pasotti, M. Birna van Riemsdijk, and Catholijn M. Jonker. Representing human habits: Towards a habit support agent. In *Proceedings of the 10th International workshop on Normative Multiagent Systems (NorMAS)*, LNCS. Springer, 2016. To appear.

Iyad Rahwan, Thomas Juan, and Leon Sterling. Integrating social modelling and agent interaction through goal-oriented analysis. *Computer Systems Science and Engineering*, 21(2), 2006.

Bastin Tony Roy Savarimuthu, Stephen Cranefield, Martin K. Purvis, and Maryam A. Purvis. Norm emergence in agent societies formed by dynamically changing networks. *Web Intelligence and Agent Systems: An International Journal*, 7(3):223–232, 2009.

Sandip Sen and Stéphane Airiau. Emergence of norms through social learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1507–1512, Hyderabad, 2007. Morgan Kaufmann Publishers Inc.

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

Rhys Smith and Jianhua Shao. Privacy and e-commerce: A consumer-centric perspective. *Electronic Commerce Research*, 7(2):89–116, 2007.

Derek J. Sollenberger and Munindar P. Singh. Kokomo: An empirically evaluated methodology for affective applications. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 293–300, Taipei, May 2011. IFAAMAS.

Daniel J. Solove. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3): 477–564, 2006. ISSN 00419907. URL `http://www.jstor.org/stable/40041279`.

Sarah Spiekermann and Lorrie Faith Cranor. Engineering privacy. *IEEE Transactions on Software Engineering (TSE)*, 35(1):67–82, January 2009.

Toshiharu Sugawara. Emergence and stability of social conventions in conflict situations. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 371–378, Barcelona, 2011. AAAI Press.

Milind Tambe, Emma Bowring, Jonathan P. Pearce, Pradeep Varakantham, Paul Scerri, and David V. Pynadath. Electric Elves: What went wrong and why. *AI Magazine*, 29(2):23, 2008.

M. Birna van Riemsdijk, Louise Dennis, Michael Fisher, and Koen V. Hindriks. A semantic framework for socially adaptive agents: Towards strong norm compliance. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 423–432, Istanbul, May 2015a. IFAAMAS.

M. Birna van Riemsdijk, Catholijn M. Jonker, and Victor Lesser. Creating socially adaptive electronic partners: Interaction, reasoning and ethical challenges. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1201–1206, Istanbul, 2015b. IFAAMAS.

Daniel Villatoro, Jordi Sabater-Mir, and Sandip 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, April 2013. ISSN 1556-4665. doi: 10.1145/2451248.2451250. URL `http://doi.acm.org/10.1145/2451248.2451250`.

Douglas Walton, Chris Reed, and Fabrizio Macagno. *Argumentation Schemes*. Cambridge University Press, 2008. ISBN 9780521897907.

Alan F. Westin. Privacy and freedom, atheneum. *New York*, 1967.

Alan F. Westin. Social and political dimensions of privacy. *Journal of Social Issues*, 59(2): 431–453, 2003.

Michael Winikoff and Lin Padgham. *Developing Intelligent Agent Systems: A Practical Guide*. Wiley, Chichester, UK, 2004. ISBN 0470861207.

Michael Wooldridge, Nicholas R. Jennings, and David Kinny. The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, September 2000. doi: 10.1023/A:1010071910869.

Chao Yu, Minjie Zhang, Fenghui Ren, and Xudong Luo. Emergence of social norms through collective learning in networked agent societies. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 475–482, Saint Paul, 2013. International Foundation for Autonomous Agents and Multiagent Systems.