

Nicoletta Fornara
Jithin Cheriyan
Asimina Mertzani (Eds.)

LNAI 14002

Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XVI

27th International Workshop, COINE 2023
London, UK, May 29, 2023
Revised Selected Papers

 Springer

Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence

14002

Founding Editor

Jörg Siekmann

Series Editors

Jaime G. Carbonell, *Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA*

Jörg Siekmann, *DFKI, Universität des Saarlandes, Saarbrücken, Germany*

Randy Goebel, *University of Alberta, Edmonton, Canada*

Yuzuru Tanaka, *Hokkaido University, Sapporo, Japan*

Wolfgang Wahlster, *DFKI, Berlin, Germany*

Zhi-Hua Zhou, *Nanjing University, Nanjing, China*

The series Lecture Notes in Artificial Intelligence (LNAI) was established in 1988 as a topical subseries of LNCS devoted to artificial intelligence.

The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Nicoletta Fornara · Jithin Cheriyan ·
Asimina Mertzani
Editors

Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XVI

27th International Workshop, COINE 2023
London, UK, May 29, 2023
Revised Selected Papers

Editors

Nicoletta Fornara 
Università della Svizzera Italiana
Lugano, Switzerland

Jithin Cheriyan
University of Otago
Dunedin, New Zealand

Asimina Mertzani 
Imperial College London
London, UK

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-031-49132-0 ISBN 978-3-031-49133-7 (eBook)
<https://doi.org/10.1007/978-3-031-49133-7>

LNCS Sublibrary: SL7 – Artificial Intelligence

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2023

Chapter “[Adding Preferences and Moral Values in an Agent-Based Simulation Framework for High-Performance Computing](#)” is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). For further details see license information in the chapter.

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Paper in this product is recyclable.

Preface

This volume collates selected revised papers presented at the 2023 edition of the Workshop Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems (COINE). Coordination, Organizations, Institutions, Norms and Ethics (COINE) are five key governance elements that regulate the functioning of open multi-agent systems. The goal of the COINE workshop series that began in 2006 is to bring together researchers in Autonomous Agents and Multi-Agent Systems (MAS) working on these five topics. The workshop focuses on both scientific and technological aspects of social coordination, organizational theory, artificial (electronic) institutions, and normative and ethical MAS.

The 27th edition of the COINE workshop, co-located with the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), was held on 29th May 2023. A total of 13 papers were submitted to the workshop and 11 were accepted after peer review (nine full research papers, one in the blue sky ideas track, and one short research paper). Each of these papers were reviewed by three Programme Committee members using a single-blind review method.

The papers were presented in three parts: (1) *Norms, Social Contracts, Institutions, and Privacy*; (2) *Studies on the Notion of Value*; (3) *Argumentation and Conventions*. About 40 participants attended the workshop. This workshop also featured an invited talk on Evolving normative systems: Allowing the system to adapt to changing needs from Marina De Vos, Department of Computer Science, University of Bath, UK. The abstract of the talk is included in this volume.

This volume contains 10 papers that are the extended and revised versions of the papers accepted at the workshop. The revisions made to the papers were each reviewed by one reviewer, and this formed the second round of peer review. We are confident this process has resulted in high-quality papers.

The workshop could not have taken place without the contribution of many people. We are very grateful to our invited speaker as well as to all the COINE 2023 participants who took part in the discussions. We thank all the members of the Program Committee (who are listed after this Preface) for their hard work, and the guidance offered by the COIN(E) Champions. We also thank EasyChair for the use of their conference management system. Thanks also goes to Springer Cham for publishing these post-proceedings.

Nicoletta Fornara
Jithin Cheriyan
Asimina Mertzani

Organization

Chairs

Nicoletta Fornara	Università della Svizzera Italiana, Switzerland
Jithin Cheriyian	University of Otago, New Zealand
Asimina Mertzani	Imperial College of London, UK

Program Committee

Nirav Ajmeri	University of Bristol, UK
Huib Aldewereld	HU University of Applied Sciences, Netherlands
Andrea Aler Tubella	Umeå University, Sweden
Olivier Boissier	École Nationale Supérieure des Mines de Saint-Étienne, France
Stefania Costantini	University of L'Aquila, Italy
Marina De Vos	University of Bath, UK
Frank Dignum	Umeå University, Sweden
Christopher Frantz	Norwegian University of Science and Technology, Norway
Maite Lopez-Sanchez	University of Barcelona, Spain
Eric Matson	Purdue University, USA
Andreas Morris-Martin	University of Bath, UK
Juan Carlos Nieves	Umeå University, Sweden
Pablo Noriega	IIIA-CSIC, Spain
Julian Padget	University of Bath, UK
Jeremy Pitt	Imperial College London, UK
Bastin Tony Roy Savarimuthu	University of Otago, New Zealand
Jaime Simão Sichman	Universidade de São Paulo, Brazil
Munindar Singh	North Carolina State University, USA
Javier Vázquez-Salceda	Universitat Politècnica de Catalunya, Spain
Harko Verhagen	Stockholm University, Sweden
Pinar Yolum	Utrecht University, Netherlands

Additional Reviewers

Emre Erdogan
Jessica Woodgate

Utrecht University, Netherlands
University of Bristol, UK

Invited Talk

Evolving Normative Systems: Allowing the System to Adapt to Changing Needs

Marina De Vos

Department of Computer Science
University of Bath
Bath, UK
M.D.Vos@bath.ac.uk

Abstract: In our complex human society, norms, policies, and laws act as the guiding principles that shape behaviour and interactions. These rules define the standards of conduct for individuals and prescribe consequences for adherence or violation. These principles also extend to socio-technical systems, where both human and software agents coexist. Within this realm, autonomous agents have the discretion to follow or deviate from established norms.

Traditionally, the normative system was created at design time and was immutable. More recently, acknowledging that norms need to change with the environment to remain relevant, the synthesis of norms at runtime is being studied more extensively. As socio-technical systems must adapt to serve their evolving purposes and reflect the needs of both participants and other stakeholders, the set of governing norms must evolve as well.

This talk elucidates the innovative Round-trip Engineering Framework, as proposed by Morris-Martin, De Vos, and Padget. This dynamic normative system is informed by the experiences of participating agents, allowing them to influence the evolution of norms governing the system at run time. This approach represents a crucial first step toward achieving self-governance in socio-technical systems through explicit and adaptable norms.

Our presentation will demonstrate the practicality of this framework, utilising the normative specification language INSTAL, and leveraging XHAIL, a symbolic Machine Learning system, to facilitate norm revision and adaptation. To conclude, we delve into current challenges and explore potential avenues for addressing them.

Biography Marina De Vos is a Senior Lecturer/Associate Professor in artificial intelligence and director of training for the UKRI Centre for Doctoral Training in Accountable, Responsible and Transparent AI at the University of Bath. Marina's research interests lie in automated human reasoning to allow better access to specialist knowledge, explainable artificial intelligence methods and modelling the behaviour of autonomous systems. Some application areas of her work are normative systems, legal reasoning, automated

music composition and assessing building damage after earthquakes. In her work on normative multiagent systems in the field of COINE, Marina currently explores systems that can automatically evolve through both external and internal stimuli and different ways of explaining the (normative) decisions by agents. Recently, she started looking at how to combine normative and value based systems. She has twice had the privilege of serving as a COINE programme chair.

Contents

Norms, Social Contracts, Institutions, and Privacy

PACCART: Reinforcing Trust in Multiuser Privacy Agreement Systems	3
<i>Daan Di Scala and Pinar Yolum</i>	
Generalising Axelrod's Metanorms Game Through the Use of Explicit Domain-Specific Norms	21
<i>Abira Sengupta, Stephen Cranefield, and Jeremy Pitt</i>	
Incentivising Participation with Exclusionary Sanctions (Full)	37
<i>Buster Blackledge, Antonios Papaioikonomou, Matthew Scott, Asimina Mertzani, Noan Le Renard, Hazem Masoud, and Jeremy Pitt</i>	
Governing Agents on the Web: (Blue Sky Ideas)	55
<i>Victor Charpenay, Matteo Baldoni, Andrei Ciortea, Stephen Cranefield, Julian Padget, and Munindar P. Singh</i>	

Studies on the Notion of Value

Addressing the Value Alignment Problem Through Online Institutions	77
<i>Pablo Noriega, Harko Verhagen, Julian Padget, and Mark d'Inverno</i>	
Adding Preferences and Moral Values in an Agent-Based Simulation Framework for High-Performance Computing	95
<i>David Marin Gutierrez, Javier Vázquez-Salceda, Sergio Alvarez-Napagao, and Dmitry Gnatyshak</i>	
Social Value Orientation and Integral Emotions in Multi-Agent Systems	118
<i>Daniel E. Collins, Conor Houghton, and Nirav Ajmeri</i>	

Argumentation and Conventions

Towards Ethical Argumentative Persuasive Chatbots	141
<i>Caren Al Anaissy, Srdjan Vesic, and Nathalie Nevejans</i>	
Uncertain Machine Ethical Decisions Using Hypothetical Retrospection	161
<i>Simon Kolker, Louise Dennis, Ramon Fraga Pereira, and Mengwei Xu</i>	

Towards Convention-Based Game Strategies 182
Shuxian Pan and Carles Sierra

Author Index 197

Norms, Social Contracts, Institutions, and Privacy



PACCART: Reinforcing Trust in Multiuser Privacy Agreement Systems

Daan Di Scala^(✉)  and Pinar Yolum 

Utrecht University, Utrecht, The Netherlands
daandiscala@hotmail.com, p.yolum@uu.nl

Abstract. Collaborative systems, such as Online Social Networks and the Internet of Things, enable users to share privacy sensitive content. Content in these systems is often co-owned by multiple users with different privacy expectations, leading to possible multiuser privacy conflicts. In order to resolve these conflicts, various agreement mechanisms have been designed and agents that could participate in such mechanisms have been proposed. However, research shows that users hesitate to use software tools for managing their privacy. To remedy this, we argue that users should be supported by trustworthy agents that adhere to the following criteria: (i) concealment of privacy preferences, such that only necessary information is shared with others, (ii) equity of treatment, such that different kinds of users are supported equally, (iii) collaboration of users, such that a group of users can support each other in agreement and (iv) explainability of actions, such that users know why certain information about them was shared to reach a decision. Accordingly, this paper proposes PACCART, an open-source agent that satisfies these criteria. Our experiments over simulations and user study indicate that PACCART increases user trust significantly.

Keywords: Multiuser privacy · Trust · Equity

1 Introduction

Privacy is the right of individuals to keep personal information to themselves [31]. While many systems are built with configurations to enable users to exercise this right, managing privacy is still a difficult problem. Collaborative systems, such as Online Social Networks and Internet of Things, contain a vast amount of content that pertain to a single individual, making it difficult, if not impossible, for individuals to attend to each piece of content separately [20]. Recent research on privacy agents shows promising results on how agents can help with privacy, such as on detecting privacy violations [14], recommending sharing behavior [11, 26], and learning privacy preferences [16, 30]. An important aspect to consider is **co-owned** content, such that the content does not belong to a single individual (e.g., medical information), but pertains to multiple people (e.g., a group photo or co-edited document [10]). These co-owners of the content can and do have

conflicting desires about the usage of the content, leading to what is termed as **multiuser privacy conflicts (MPCs)** [23, 28].

Various decision-making techniques, such as auctions, negotiation, and argumentation have been employed to build systems to resolve MPCs. Simply put, each user that participates in these systems is represented by a privacy agent that knows its user’s privacy requirements. The agent participates in the decision-making system on behalf of its user. For auction-based systems, this means bidding on its user’s behalf or for argumentation-based systems, this would correspond generating arguments on behalf of its user. Through participation in this system, the agents decide if and how to share co-owned content by resolving conflicts. Experimental evaluations on these systems yield good performance results. However, it is also known that users have concerns when it comes to using software tools for managing various elements of their privacy [12, 27].

Many existing studies of collaborative systems indicate the importance of *trust* in making systems usable by individuals [5, 17]. We argue that to realize trust, the privacy agent of a user should satisfy the following properties:

Concealment: The privacy agent will know the privacy constraints of the user, either through elicitation or learning over time. When the agent is interacting with others to resolve conflicts, it should reveal as little as possible about these privacy constraints, since the privacy constraints themselves are private information. So, users would know that their privacy is safe with their agent [2, 17].

Equity: Different users have different privacy stances in terms of their motivation and knowledge. While some users would fight not to share a piece of content, others will be indifferent. Contrary to some of the existing work in AI that favors users with certain properties [19, 24], we do not want any user to be left behind. Ideally, the privacy agent should take the privacy stance of the user into account and be able to help different types of users as equally as possible; thereby creating equity [31, 33].

Collaboration: It is possible that a number of agents that participate in the same conflict resolution have similar privacy concerns or complementary information to support a particular privacy decision [32]. Their agents should be able to collaborate in groups.

Explainability: It is well-studied that often users do not trust privacy tools because of misconceptions [27]. One solution for this is to make the tools explicit to users. But, more importantly, if the agent itself can provide explanations as to why it has taken certain actions, then its user can understand and even configure the agent better for future interactions [9, 21].

Accordingly, this paper proposes a new Privacy Agent for Content Concealment in Argumentation to Reinforce Trust (PACCART). PACCART can conceal its user’s privacy requirements at different levels, while still resolving conflicts. By adapting to different privacy understandings of users, PACCART will provide equitable treatment. At the same time, PACCART will enable agents to work together towards a shared desired outcome. Finally, it will help its user understand the actions it is taking. To the best of our knowledge, this is the first

privacy agent that brings these desirable properties together. We made PACCART openly available.¹

The rest of this paper is organized as follows: Sect. 2 explains the necessary background theory on argumentation-based agreement systems. Section 3 formalizes the PACCART model. Section 4 describes our realization of the model and our experimental results. Section 5 discusses the user study and its results. Finally, Sect. 6 systematically compares our approach with related work and gives pointers for future directions.

2 Background

We advocate that for an agent to exhibit these four criteria, it is useful to be able to express the relations between privacy preferences in a semantic manner. Thus, as an underlying agreement system, we opt for argumentation as opposed to other decision-making mechanisms such as auctions or negotiation. Below, we review how a privacy agent would use argumentation theory and how by using a dialogical argumentation system it can resolve privacy disputes.

2.1 Argumentation Theory

Our agent model makes use of argumentation theory for its reasoning. We follow the **structured argumentation** formalism of ASPIC+ [22]. An ASPIC+ **argumentation** or **dispute** $d = \langle P, R, B, C \rangle$ consists of **premises** $P = P_o \cup P_n$ (ordinary premises P_o and necessary premises P_n), **rules** $R = R_s \cup R_d$ (strict rules R_s and defeasible rules R_d), **biases** $B = B_p \cup B_r$ (premise biases B_p and rule biases B_r) and Contraries C .

A dispute is held between two opposing agents, **proponent** a_p and **opponent** a_o . Agents have access to their **knowledge base** KB , which contains premises, rules and contraries. With this content, agents can form **arguments**. In order to win the dispute, agents are able to **attack** each other’s arguments and can **support** (or **defend**) their own arguments with subarguments in order to try to win the dispute [7]. In some cases an agent is also able to **forfeit**, giving up on winning the dispute. Arguments can be attacked on their **weak points**, which is any subargument that is either a consequent of a defeasible rule or any ordinary premise. **Useful** arguments are arguments that, when added to the dispute, successfully attack any opponent’s current arguments. Acceptability conditions of winning or losing are dependent on the chosen **semantics**. Baroni et al. [3] offer an overview of different semantics and their meaning, including **grounded**, **preferred**, **complete** and **stable** semantics.

2.2 Dispute Protocol

In order for an argumentation agent to be able to hold a dispute with other agents about a **subject**, it follows a communication protocol. The protocol allows

¹ <https://github.com/PACCART/PACCARTpaper>.

agents to **extend** the dispute, meaning that they take turns adding arguments from their knowledge base to the dispute in order to either defend or attack the dispute subject.

Algorithm 1. Agent Dispute Extension Protocol

Require: Agents $A = \{a_p, a_o\}$, each with $KB = \langle P, R, C \rangle$

Ensure: Determine winner of dispute d

```

1:  $a \leftarrow a_p$ 
2: while  $d$  is not forfeited do
3:   if  $a$  can extend  $d$  then
4:      $a$  extends  $d$ 
5:     if  $a$  is  $a_p$  then
6:        $a \leftarrow a_o$ 
7:     else
8:        $a \leftarrow a_p$ 
9:     end if
10:  else
11:     $a$  forfeits  $d$ 
12:  end if
13: end while

```

Argumentation systems like PriArg [13] utilize this kind of extension protocol, as denoted in Algorithm 1. According to the extension protocol, if an agent is able to extend the dispute, it does so. An agent extends the dispute by adding any sufficient argument from its knowledge base. Therefore, as soon as an agent is unable to extend the dispute any further, it forfeits the dispute.

The winner of a dispute is determined by evaluating the outcome according to grounded semantics. This way the **burden of proof** initially lies on the proponent of the dispute, after which agents take turns by extending the dispute until one of them wins. This is done because the agent that initializes the dispute has something to gain by defending the subject.

3 Model

The PACCART agent consists of a base component, which works similarly to agents in the PriArg system, as it communicates with other agents through a dialogical argumentation framework that follows the same Dispute Extension Protocol, as defined in Sect. 2.2. Following this, four components will be introduced on top of the workings of the base component.

3.1 Concealment Component

In the case of argumentation over privacy issues, the information to be concealed consists of all information that a user’s agent can hold in its knowledge base,

including those that pertain to the user’s privacy preferences. We make a distinction between content that is revealed during a dispute and content that is not (yet) revealed, by keeping track of concealed rules R_c and premises P_c .

We make a distinction between content in Agent A ’s knowledge base KB that is concealed and content that is not, by keeping track of different sets throughout the dispute. At the initialization stage of the dispute, agents have not yet shared any content with each other, which means that all content is still concealed ($R_c = R$ and $P_c = P$). While the dispute develops, each time an agent shares content with another agent to extend the dispute, that content is revealed and therefore removed from the set of concealed content (if r or p is revealed: $R_c \leftarrow R_c \setminus r$ or $P_c \leftarrow P_c \setminus p$).

We formalize PACCART’s concealment component by providing it the ability to adopt a **privacy behavior**, consisting of three concealing aspects: **Scope**, **Division** and **Dedication**.

Scope: At each point in the dispute, if possible, an agent extends the dispute by adding one or more arguments (Algorithm 1, Step 4). The amount of useful arguments (as defined in Sect. 2.1) that an agent considers to add at any point of time to the dispute, is called its scope. An agent without any focused scope would add all available useful arguments at once. An agent with a focused scope is able to carefully select a smaller set of arguments, and locally gains control over the amount of the added (and therefore revealed) content. The larger the scope of an agent, the more content is added at each step in the dispute.

Division: Not all information is equally important. To be able to denote this, KB . To achieve this, we split the sets of contents into **set-families** [4] of content. These subgroups can then be ordered to the likings of the agent. This entails splitting the knowledge base into ordered subgroups of different groups of conceal-worthy content. Therefore, based on the original knowledge base $KB = \{P, R, C\}$, we propose an **ordered subdivided knowledge base (OSKB)**, which includes the following ordered tuples of set-families:

- An ordered tuple of premises $O_P = \langle P_1, \dots, P_n \rangle$
- An ordered tuple of rules $O_R = \langle R_1, \dots, R_n \rangle$

The relation between these ordered set-families F_X and the sets X (with $X = P, R$) all follow the same properties:

- $\bigcup F_X = X$
- $\bigcap F_X = \emptyset$
- $\forall y \in Y, \forall z \in Z ((Y \subseteq F_X \wedge Z \subseteq F_X) \rightarrow (y = z \leftrightarrow Y = Z))$.

With the introduced *OSKB*, an agent can order their content based on its concealment preferences. We can therefore treat these two ordered tuples together as one totally ordered knowledge base, subdivided in what we call **dedication levels**, as follows: $L = \langle \{O_{P_1}, O_{R_1}\}, \dots, \{O_{P_n}, O_{R_n}\} \rangle$. Each level contains one or more premises and rules. The first level L_1 contains content at the top of the ordering of each of the *OSKB* tuples, which is the content that the agent is

the least concerned about revealing. The last level L_n contains content at the bottom of the ordering, indicating the content that the agent considers most important to conceal and therefore has to fully commit to winning the dispute in order to be willing to reveal these pieces of information. The **exhaustion** of an agent’s division aspect indicates the amount ordered subdivisions an agent makes. The more exhaustive content subdivision, the higher amount of levels an agent splits its *OSKB* up into.

An example of four different *OSKB* divisions is shown in Fig. 1, where an agent makes no subdivision of its *OSKB* (Fig. 1a), by adding all its arguments to Level 1. Another possibility is it divides its *OSKB* in half, with two levels (Fig. 1b). Furthermore, an agent can choose to divide its *OSKB* in all separate arguments, which yields four levels in this case (Fig. 1c). Note that with this approach, Fig. 1 shows an example of a level with just one premise (‘j’), as only one premise can suffice to form an argument. A final approach consists of an agent dividing its *OSKB* by subdividing all of its content (all rules and premises) over different levels, yielding ten levels in this case (Fig. 1d).

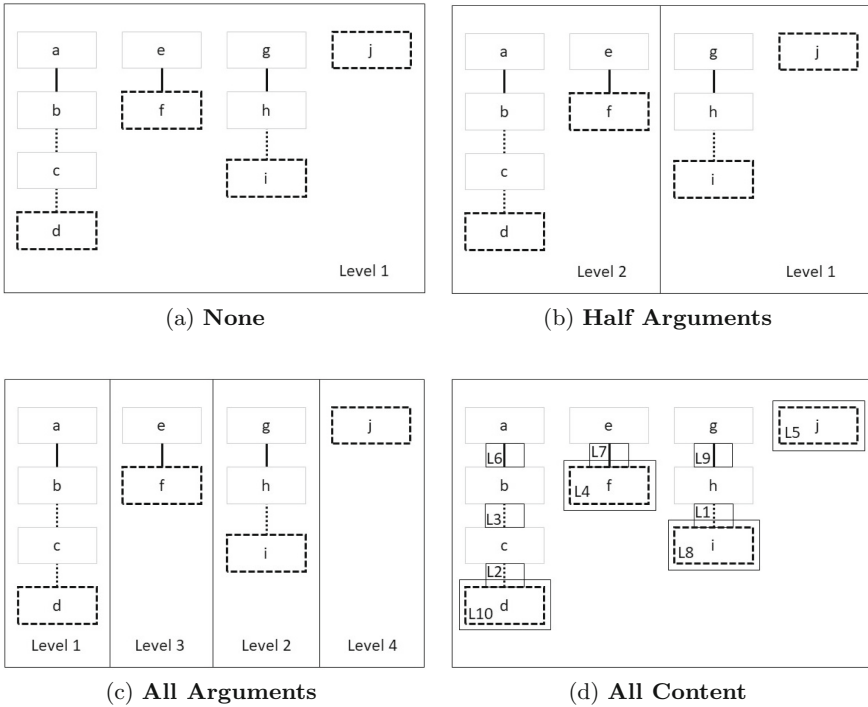


Fig. 1. Examples of different approaches of the PACCART agent’s division aspect. Four arguments consisting of ten pieces of content are divided up into different levels. Solid and dashed lines are for strict rules R_s and defeasible rules R_d , respectively.

Dedication: Agents that are able to divide their content into levels, can use this to their advantage. Such agents will initially only provide arguments if they can do so from their first level in their knowledge base. When all arguments in a first level have been depleted, the agents receive the option to either drop to a new level, therefore making a further argument privacy concession, or to forfeit the dispute. This gives agents the ability to weigh their decision to further dedicate to the argumentation. The amount of **willingness** to drop determines the agent’s dedication to continue the dispute. The more willing an agent is to drop dedication levels, the more it will use and therefore reveal the contents of its *OSKB*. This is calculated by whether a certain willingness Threshold θX with $X \in [0, 100]$ is met at the time of decision whether to commit further to the dispute. This means that an agent with $\theta 75$ has a 75% chance of dropping each level. This entails that the agent example of Fig. 1c has a $0.75^3 = 42.2\%$ chance to use the content of its final level (as it could drop three times until it reaches its fourth and final level of content), whereas agent example of Fig. 1d has a $0.75^9 = 7.5\%$ chance to fully commit its *OSKB*.

Any combination of all three concealing aspects maps to an agent’s **privacy type**. These privacy behaviors are in place for agents to further gain control over their content concealment during disputes, as well as influence their win rate.

3.2 Equity Component

Recall that we want our PACCART agent to be able to help different types of users to deliver on the equity aspect. On user’s privacy stances, we follow Dupree et al. [8], who determine a categorization based on stances regarding privacy along two dimensions. We define a user u with **knowledge** $k \in \{low, medium, high\}$ and **motivation** $m \in \{low, medium, high\}$. The degree of knowledge indicates the amount of awareness a user has about their privacy and the degree of general knowledge on privacy matters. The degree of motivation indicates the effort a user expends to protect their privacy and the degree of willingness to act on privacy matters. Each system user falls in one of five categories, also known as **privacy types**:

- **Fundamentalists:** high knowledge, high motivation
- **Lazy Experts:** high knowledge, low motivation
- **Technicians:** medium knowledge, high motivation
- **Amateurs:** medium knowledge, medium motivation
- **Marginally Concerned:** low knowledge, low motivation.

Dupree et al. determine the rate at which users fall into these categories: 3% of users are Fundamentalists, Lazy Experts 22%, Technicians 18%, Amateurs 34% and Marginally Concerned 23%. This is comparable to the categorical distributions of privacy types of earlier conducted researches [1, 6, 8, 18, 25, 29].

We define PACCART agents that adapt to the knowledge and motivation of the users’ privacy type as **Personalized** agents, whereas we consider **indifferent** agents to be not personalized and therefore have an unfocused scope and

make no distinction between the importance of content in their *KB*. In order for personalized agents to be considered equitable, they should adhere to the following equity properties, based on earlier research on equity [31, 33]:

- EP1:** The knowledge and motivation of a user is considered and utilized to the fullest extent by their personalized agent.
- EP2:** A personalized agent outperforms an indifferent agent.
- EP3:** There are no performance outliers between personalized agents; no personalized agent heavily over- or underperforms compared to others.

EP1 is important because the strengths of the user should be taken into account by their agent. The privacy stance of a user should not be ignored, as this would be unfair towards users that are heavily engaged in protecting their privacy. In the same line, EP2 is important because the agents that are tailored towards a user should not perform worse than an agnostic, basic agent. Providing personalization should be beneficial for users, not disadvantageous. EP3 is important because in order to reach fair outcomes, it should not be the case that the privacy stance of a user exorbitantly influences the performance of their agent. It would e.g. be unfair towards unknowledgeable users if their agents would underperform by design.

In order to meet these properties, we introduce a mapping between users and agents. This way, both knowledge and motivation are used to determine the personalized agent’s privacy type. We determine a fitting mapping between users u to their agents a such that all users get mapped to the shortest scope, user knowledge is mapped to agent division and user motivation is mapped inversely to agent dedication. We will substantiate each mapping.

First, we assign all personalized agents to have a small scope, since a small scope is beneficial for all users, independent of privacy stance. When a user has a high privacy stance, they can let their agent subdivide its content in such a way that each piece of content is thoroughly protected. This would mean that the agent already has a small amount of content to choose from, so for a high privacy user the scope has only a little positive impact. However, for users who do not have a lot of knowledge or motivation to bring to the dispute, a small scope is also the best fit as it protects as much content as possible.

Secondly, we map a user’s knowledge to their agent’s division, because of the degree of user knowledge should correspond with the amount of useful subdivisions of their agent’s *OSKB* levels. This means that the higher the user’s knowledge, the higher the agent’s content dividing. Someone with a high knowledge could benefit from an agent with a high capability of dividing its knowledge base content. This would allow users to provide their agent with their preferences in detail. This is in line with EP1. Similarly, mapping a low knowledge to a low *OSKB* division would also be useful. This is because users with low knowledge have little relevant preference divisions to make in their agent’s knowledge base.

Thirdly, we map a user’s motivation inversely to their agent’s dedication, because the amount of motivation of a user should correspond to the dedication of its agent to conceal content (in favor of winning disputes). This means that

Table 1. All three concealing aspects of indifferent PACCART agent and personalized PACCART agents that are matched with representative agents for different user privacy types.

Privacy type	Scope	Division	Dedication
<i>Indifferent</i>	All	None	$\theta 100$
Fundamentalist	Shortest	AllContent	$\theta 25$
Technician	Shortest	AllArgs	$\theta 25$
Amateur	Shortest	AllArgs	$\theta 50$
Lazy expert	Shortest	AllContent	$\theta 75$
M.Concerned	Shortest	HalfArgs	$\theta 75$

the higher the user’s motivation, the lower the agent’s dedication. Users that are highly motivated to protect their data would rather have their agent drop as little levels as possible, even if it would require taking (social) losses. Similarly, users that prefer not to act on privacy matters would want their agents to perform well when it comes to winning disputes, but would not mind agents revealing information to do so. This is also in line with EP1. This mapping results in five personalized agents, one representative for each user type, as noted in Table 1. This table also includes an indifferent agent.

3.3 Additional Usability Components

In addition to the Concealment and Equity components two usability measures are taken. A Collaboration component is introduced to support both sides of the dispute to be represented by multiple agents. This is achieved by introducing the notion of teams such that the set of agents A in the protocol now consists of $A = \{T_p, T_o\}$ to support both a proponent team $T_p = \{a_{p1}, \dots, a_{pn}\}$ and opponent team $T_o = \{a_{o1}, \dots, a_{on}\}$. In order to extend a dispute each team of PACCART agents continuously selects one of its agents to extend. A team forfeits when none of its agents can extend the dispute any further. This component allows for multiple PACCART agents to cooperate on a common goal of defending/attacking a privacy related subject. This means that agents can add content from their own *OSKB* to the dispute when other agents in their team fail to do so.

Furthermore, an Explainability component is introduced to give users insights to the working of their agent. The semantic nature of PACCART allows us to produce both textual and visual output. PACCART can provide textual output by considering outcomes and providing feedback to the user. Based on this, it is able to give different kinds of feedback, with a range of detail. It can notify users on a summary (e.g., “*I have won 56% of today’s disputes and managed to conceal 73% of your content*”) or it can give detailed advice on possible actions to be taken to improve its performance (such as listing possible weak points in its arguments for the user to improve upon). Furthermore, PACCART can

provide visual output by showing its user images of the Structured Argumentation Framework [22] of final disputes. This gives users a visual overview of (counter)arguments and possible weak points in their content. This component allows users of PACCART to better understand its inner workings and performance.

4 Experimental Results

The PACCART agent and the experimental setup are implemented as a C# program. For the sake of reproducibility, we make this program and experiments open source, along with examples and schematic overviews of the PACCART agent workings.

4.1 Dataset Generation

We implement a system that generates datasets of disputes according to four parameters. The **disputeAmount** parameter indicates the amount of generated unique disputes. A higher input value indicates a larger set of disputes, therefore less prone to outliers. The **disputeSize** parameter controls the amount of arguments that the dispute can contain. A higher input value indicates larger disputes with more content. The **maxArgumentSize** parameter dictates the maximum amount of subarguments that each argument can consist of. A higher input value indicates larger arguments with more content and therefore more attackable weak points. Finally, **maxBranches** is used to control the maximum amount of attacks that each weak point can have, indicating a branching choice in the dispute. A higher input value indicates more options for both agents.

By tuning these parameters, we are able to generate dispute datasets of various shapes and sizes, which makes for exhaustive possibilities for testing functionalities of PACCART. After preliminary analysis of variables, we generate a dispute dataset based on the default parameter settings (`disputeAmount = 200`, `disputeSize = 20`, `maxArgumentSize = 10`, `maxBranches = 2`).

4.2 Experiment 1: Effect of Privacy Behaviors

Setting The goal of the first experiment is to test the performance of PACCART agents. Agent performance is evaluated on two metrics, average concealment C_{avg} and average win rate W_{avg} . We hypothesize the following:

- H1:** A smaller scope leads to both increased concealment and increased win rate.
- H2:** More exhaustive division leads to increased concealment and decreased win rate.
- H3:** A higher dedication leads to decreased concealment and increased win rate.

We determine four or five conditions for each of the three privacy behavior aspects, to test the range of PACCART’s concealing behaviors. For the scope, we include selecting the **Shortest** or **Longest** arguments, as well as a **Random** argument or **All** possible arguments. For the division, we follow the examples of Fig. 1 and include conditions where **None** of the content is split, where the *OSKB* is split into two groups of arguments (**HalfArgs**), split into all separate levels of arguments (**AllArgs**) or a subdivision where each level contains a single piece of content (**AllContent**). The dedication conditions consist of an increasing threshold θ , with $\theta \in \{0, 25, 50, 75, 100\}$ that should be met in order to drop to a new level. These conditions yield 80 possible privacy types. Each of these 80 predetermined agents are set up against all other agents, and simulations are run on the 200 disputes of our dataset. This means that the experiment is run on 16,000 disputes for 80 agent set-ups, totaling in 1,280,000 simulated disputes. For each of the disputes, both agents are evaluated as a proponent, as well as opponent of the dispute, to ensure equal chances of winning.

Results Figures 2 and 3 depict the performance of the 80 different agent privacy behavior types, across all three concealing aspects.

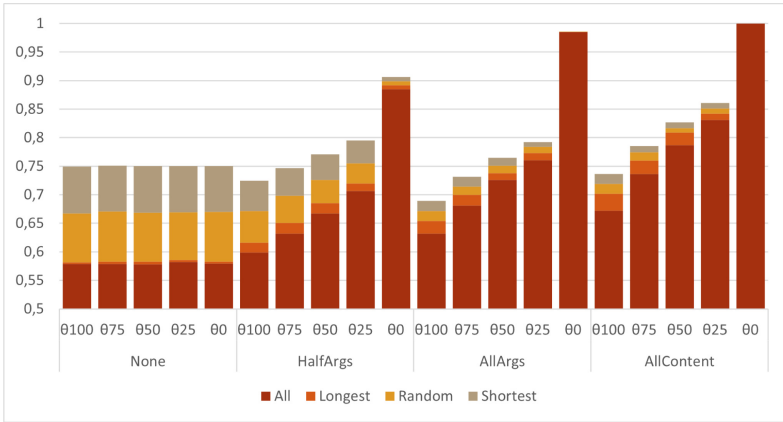


Fig. 2. Average Concealment C_{avg} results for all PACCART privacy types.

Scope: We observe from Figs. 2 and 3 that the scope of an agent has a significant effect on its performance. Both the average win rate W_{avg} and average concealment C_{avg} increase with a smaller scope. We conclude that a smaller scope has a strictly positive impact. This confirms hypothesis H1.

Dividing: All of the **None** dividing aspect results are equal, independent of dropping willingness. This means that not dividing the *OSKB* negates the effect of the agent’s dedication. This is an expected outcome, which happens because there is no division made of the knowledge base so there are no levels for the

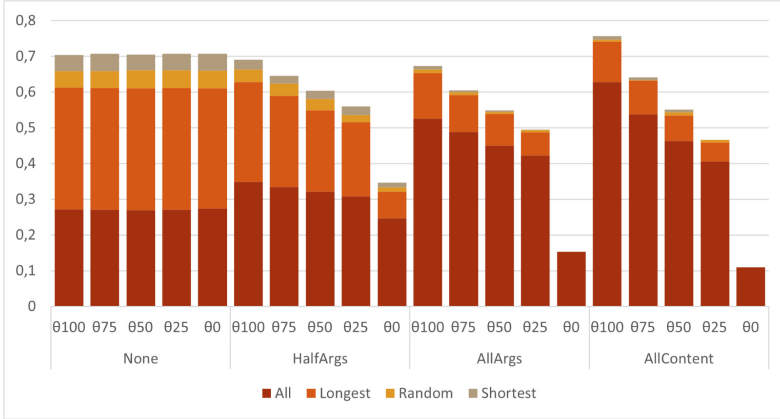


Fig. 3. Average Win Rate W_{avg} results for all PACCART privacy types.

agent to drop between, even if it would be willing. Outside of this behavior, an upward trend is noticeable in all cases for average concealment, as well as a downward trend in all cases for the win rate, with more exhaustive dividing. This confirms hypothesis H2.

Dedication: When looking at the dedication aspect, we observe an upward trend in all cases for average concealment C_{avg} , as well as a downward trend in all cases for win rate W_{avg} , with less willing dedication. This is a similar trend as with the dividing aspect of the privacy behavior. This confirms hypothesis H3. Furthermore, Fig. 3 shows a significant drop in win rate from θ_{25} to θ_0 , while the improvement in concealment is disproportional. This shows that it is beneficial for an agent to be at least somewhat willing to commit to the dispute. Based on these results, we conclude the following observation:

Observation 1. *PACCART's concealment component allows users to keep information private, while also giving them the choice of a trade-off between winning disputes and further protection of information.*

4.3 Experiment 2: Effect of User-Agent Mapping in Realistic Setting

Setting The goal of the second experiment is to evaluate the mappings between agents and users by simulating disputes for each personalized agent in a realistic setting. The results of this mapping will determine whether EP2 and EP3 are met, which means that PACCART is an equitable agent. Therefore, based on this mapping, we further hypothesize:

H4: Equity property EP2 is met under a mapping where personalized agents are assigned the smallest possible scope.

H5: Equity property EP3 is met under a mapping where personalized agents are assigned a fitting trade-off between division and dedication.

We create a set of opponents according to data of distribution of real life user population as given by Dupree et al. This opponent set therefore contains three Fundamentalist agents, 22 Lazy Expert agents, 18 Technician agents, 34 Amateur agents and 23 Marginally Concerned agents. We call this set of 100 agents the **Model Population Set MPS**. The MPS is in place because in a practical scenario it is less likely that an MPC occurs between Fundamentalists’ agents, as between Marginally Concerned users’ agents.²

This means that six agents (one indifferent agent and all five personalized agents) compete 100 times against each of the personalized agents, and simulations are run on 200 disputes on the dispute dataset. Overall, the experiment is run on 20.000 disputes for six agent set-ups. Furthermore, agents are again tested twice for all disputes, both as proponent and opponent of the subject, to ensure equal chances of winning.

Results The results of the second experiment can be seen in Fig. 4. Again, performance is measured by concealment C_{avg} and win rate W_{avg} . As shown in Fig. 4, the indifferent agent performs much worse than the personalized agents on both metrics (only 0.185 for win rate and 0.660 for concealment). This confirms hypothesis H4.

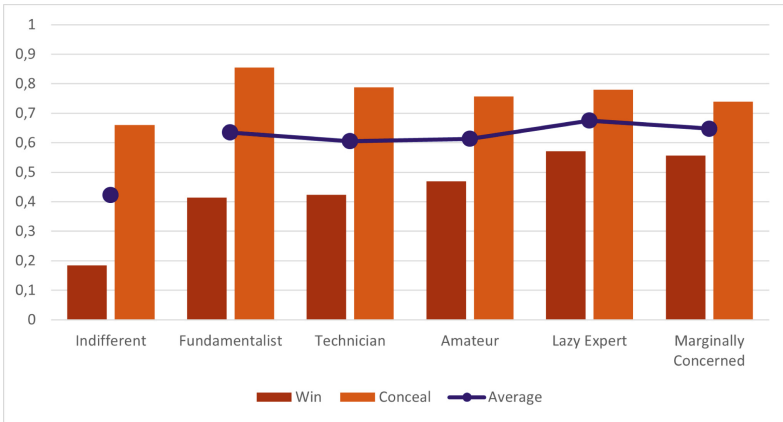


Fig. 4. Average win rate W_{avg} and Average Concealment C_{avg} for indifferent agent and personalized agents in MPS. Averages between W_{avg} and C_{avg} are indicated with a line.

Furthermore, the averages of all personalized agents range between 0.6 and 0.7. This means that although some personalized agents are better at winning

² An additional experiment is performed to evaluate the MPS, placing all agents in a non-distributed setting, which yielded similar results.

or concealing, the overall performance leads to an equitable situation where no users are victimized by the agent’s workings. This confirms hypothesis H5. It is worth noting that an interesting trend occurs between personalized agents, where the Fundamentalist representative’s agent (with the highest privacy stance) wins the least and conceals the most, while the Marginally Concerned representative’s agent wins the most and conceals the least. This trade-off shows how the different privacy stances influence the results. Based on these results, we conclude the following observation:

Observation 2. *PACCART’s equity component allows for a well-matched personalization for users of various privacy stances. While personalized PACCART agents overall perform relatively well, a consistent trade-off between win rate and concealment shows that no user is disadvantaged.*

5 User Study

We further conduct a user study to understand what components of PACCART lead to user trust.

5.1 Setting

We design a survey in two parts. The first part of the survey has questions on the privacy stance of participants, in order to assess their privacy type. We deliberately use existing questions from the literature to ensure compatibility: three questions used by Westin et al. (e.g., “*How much do you agree with the statement ‘Most businesses handle the personal information they collect about consumers in a proper and confidential way.’?*”) [15] to determine the knowledge of participants on privacy and 10 questions on statements about privacy from the study of Dupree et al., to determine the motivation of participants on privacy (e.g., “*How strongly do you identify yourself with the statement ‘I would rather choose being social over privacy.’?*”). As validation and to mitigate response bias, we also ask participants directly to self-assess their own knowledge and motivation (e.g., “*How much do you know about digital privacy issues?*”). These questions are all answered on a Likert scale. The full questionnaire is also made openly available.

The second part of the survey has questions on the various components of PACCART as a personal assistant. This part starts with an example scenario. Then a set of questions follows in which participants are asked to rate their perceived trust of such personal assistants on a Likert scale (1 = Strongly Distrust, 5 = Strongly Trust). The first question is on the participants’ initial thoughts of trust on the PACCART base component (an explanation followed by “*How much would you trust to use such a privacy assistant?*”). Then, each separate PACCART component is explained separately and addressed as a question. Afterwards, the participants are asked to rate the agent with all components combined

(the base component with all four additional components). Finally, the participants are asked to reconsider their thoughts on the base component. This gives the participants a chance to reflect on their initial thoughts.

The survey is distributed through Qualtrics, an online, secure cloud-based, survey tool. Data is automatically and anonymously recorded through Qualtrics, in accordance with GDPR requirements. The survey is preceded by filling out a consent form. To ensure correctness and clarity, we first perform a small pilot study. Afterwards, the survey is distributed online for the user study.

5.2 Results

Data was collected from 117 voluntary participants in the user study. Based on validation questions and completion requirements, 12 survey responses are filtered out. Out of the remaining 105 participants, eight participants self-assessed as Fundamentalists, 20 participants as Lazy Experts, 22 as Technicians, 31 as Amateurs and 24 as Marginally Concerned users. This is in line with the distributions by Dupree et al. [8].

We report the mean (M) and standard deviation (SD) of the results, as well as significance through t-tests (P). The results indicate that the initial consideration of the PACCART base component is fairly neutral ($M = 2.857$, $SD = 1.023$), slightly leaning towards distrust. The trust ratings given by participants are higher than the initial consideration for both Concealment ($M = 2.943$, $SD = 0.979$) as well as Equity ($M = 3.171$, $SD = 1.069$). There is a significantly ($P < .001$) positive increase of trust of the combined agent ($M = 3.467$, $SD = 0.974$) compared to the initial consideration of the base component. Even more so, when asked to reevaluate the trustworthiness of the agent, the average trust rating significantly ($P < .001$) drops ($M = 2.362$, $SD = 0.982$) compared to the combined agent.

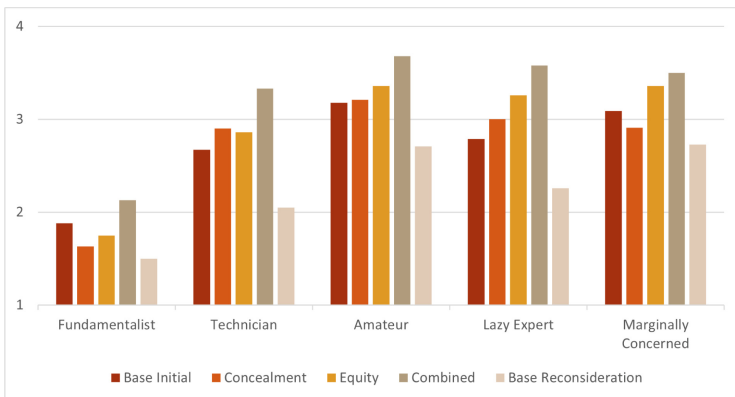


Fig. 5. Average survey study ratings of participant trust on a Likert scale (1 = Strongly Distrust, 2 = Distrust, 3 = Neutral, 4 = Trust, 5 = Strongly Trust), divided by privacy type.

The results are split on each of the privacy type categories, as shown in Fig. 5. The graph shows the average trust ratings by users with different privacy types. From this graph, we can observe the effect of different stances on privacy on agent trust scores. These results show that the lower the privacy stance, the higher the overall trust in the agent. Fundamentalists participants’ highest mean trust rating is a 2.130, whereas Marginally Concerned participants’ lowest mean trust rating is a 2.727. This is in line with our expectations about the privacy types and therefore an indication that the privacy stance assessment part of the survey works as intended.

A further noteworthy observation is that for all of the individual privacy types the reconsideration is rated lower than the initial thoughts on the base component. This indicates that after having read an explanation on what possible components could improve upon the base, participants independently of their privacy type assess the base component to be less trustworthy. When comparing the base component with the total combined agent, trust significantly increases for all user types ($P < 0.001$) except for Fundamentalists. While the results do indicate an increase of trust for Fundamentalists, the results are not significant ($P = 0.18$), which is expected because of the naturally low occurrence of users with this high privacy stance. These results strongly indicate that overall, the principles of PACCART and its components increase the indicated trust of users of all privacy stances.

6 Conclusion

We introduced PACCART, which helps users preserve privacy by enabling automated privacy argumentation. PACCART aims to induce trust by increasing content concealment, providing equitable personalizations, enabling multiagent team-based collaboration and explaining its actions through feedback. The agent is designed to be general and is made publicly available as an open-source program together with the dispute dataset generation system, so that they can be used for research as well as in practical applications, such as team collaboration tools (e.g., MS Teams) where co-owned data is shared abundantly and privacy disputes need to be resolved.

Future research could further investigate what improvements the system needs for its proper use in open systems, for example setting up standardization of the use of *OSKB*’s and determining what information is shared beforehand to increase privacy for both parties involved. Another research avenue would be to close the feedback loop between users and the agent to further increase trust. When users get prompted that their agent lost a dispute because of the lack of arguments, the user could respond by taking action to help and improve the agent fit to its user. Furthermore, introducing mutual feedback opens new possibilities for machine learning approaches. Now, there exists a mapping between users and their personalized agents, which could be changed into the agent learning the preferences of the user instead. Weights could be given to the importance

of dedication to win certain disputes, or concealing specific levels of content. The inclusion of reinforcement learning could be an important additional step towards robust and well-adjusted argumentation based privacy assistants.

References

1. Ackerman, M.S., Cranor, L.F., Reagle, J.: Privacy in e-commerce: examining user scenarios and privacy preferences. In: Proceedings of the 1st ACM Conference on Electronic Commerce, pp. 1–8 (1999)
2. Addo, I.D., Ahamed, S.I., Yau, S.S., Buduru, A.: A reference architecture for improving security and privacy in internet of things applications. In: IEEE International Conference on Mobile Services, pp. 108–115. IEEE (2014)
3. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. *Knowl. Eng. Rev.* **26**(4), 365–410 (2011)
4. Brualdi, R.A.: *Introductory Combinatorics*. Pearson Education India (1977)
5. Colnago, J., Feng, Y., Palanivel, T., Pearman, S., Ung, M., Acquisti, A., Cranor, L.F., Sadeh, N.: Informing the design of a personalized privacy assistant for the internet of things. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2020)
6. Consolvo, S., Smith, I.E., Matthews, T., LaMarca, A., Tabert, J., Powledge, P.: Location disclosure to social relations: why, when, & what people want to share. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 81–90 (2005)
7. Cyras, K., Toni, F.: ABA+: assumption-based argumentation with preferences. In: Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning, pp. 553–556 (2016)
8. Dupree, J.L., Devries, R., Berry, D.M., Lank, E.: Privacy personas: Clustering users via attitudes and behaviors toward security practices. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 5228–5239 (2016)
9. Fazzinga, B., Galassi, A., Torroni, P.: An argumentative dialogue system for covid-19 vaccine information. In: International Conference on Logic and Argumentation, pp. 477–485. Springer (2021)
10. Fogues, R.L., Murukannaiah, P.K., Such, J.M., Singh, M.P.: Sharing policies in multiuser privacy scenarios: incorporating context, preferences, and arguments in decision making. *ACM Trans. Comput-Hum. Interact. (TOCHI)* **24**(1), 1–29 (2017)
11. Fogues, R.L., Murukannaiah, P.K., Such, J.M., Singh, M.P.: Sosharp: recommending sharing policies in multiuser privacy scenarios. *IEEE Internet Comput.* **21**(6), 28–36 (2017)
12. Jin, H., Guo, B., Roychoudhury, R., Yao, Y., Kumar, S., Agarwal, Y., Hong, J.I.: Exploring the needs of users for supporting privacy-protective behaviors in smart homes. In: CHI Conference on Human Factors in Computing Systems, pp. 1–19 (2022)
13. Kökciyan, N., Yaglikci, N., Yolum, P.: An argumentation approach for resolving privacy disputes in online social networks. *ACM Trans. Internet Technol.* **17**(3), 1–22 (2017)
14. Kökciyan, N., Yolum, P.: Priguard: a semantic approach to detect privacy violations in online social networks. *IEEE Trans. Knowl. Data Eng.* **28**(10), 2724–2737 (2016)

15. Kumaraguru, P., Cranor, L.F.: Privacy indexes: a survey of Westin's studies. Carnegie Mellon University, School of Computer Science, Institute for Software Research International (2005)
16. Kurtan, A.C., Yolum, P.: Assisting humans in privacy management: an agent-based approach. *Auton. Agent. Multi-Agent Syst.* **35**(1), 1–33 (2021)
17. Maple, C.: Security and privacy in the internet of things. *J. Cyber Policy* **2**(2), 155–184 (2017)
18. Maus, B., Olsson, C.M., Salvi, D.: Privacy personas for IoT-based health research: A privacy calculus approach. *Frontiers in digital health*, p. 187 (2021)
19. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. *ACM Comput. Surv. (CSUR)* **54**(6), 1–35 (2021)
20. Mester, Y., Kökciyan, N., Yolum, P.: Negotiating privacy constraints in online social networks. In: *International Workshop on Multiagent Foundations of Social Computing*, pp. 112–129. Springer (2015)
21. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019)
22. Modgil, S., Prakken, H.: The aspic+ framework for structured argumentation: a tutorial. *Argum. Comput.* **5**(1), 31–62 (2014)
23. Mosca, F., Such, J.M., McBurney, P.: Towards a value-driven explainable agent for collective privacy. In: *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1937–1939 (2020)
24. O'neil, C.: *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown (2017)
25. Sheehan, K.B.: Toward a typology of internet users and online privacy concerns. *Inf. Soc.* **18**(1), 21–32 (2002)
26. Squicciarini, A.C., Novelli, A., Lin, D., Caragea, C., Zhong, H.: From tag to protect: A tag-driven policy recommender system for image sharing. In: *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pp. 337–33709. IEEE (2017)
27. Story, P., Smullen, D., Yao, Y., Acquisti, A., Cranor, L.F., Sadeh, N., Schaub, F.: Awareness, adoption, and misconceptions of web privacy tools. *Proc. Privacy Enhanc. Technol.* **2021**(3), 308–333 (2021)
28. Such, J.M., Porter, J., Preibusch, S., Joinson, A.: Photo privacy conflicts in social media: A large-scale empirical study. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 3821–3832 (2017)
29. Taylor, H.: Most people are “privacy pragmatists” who, while concerned about privacy, will sometimes trade it off for other benefits. *The Harris Poll* **17**(19), 44 (2003)
30. Tonge, A., Caragea, C.: Image privacy prediction using deep neural networks. *ACM Trans. Web (TWEB)* **14**(2), 1–32 (2020)
31. Ulusoy, O., Yolum, P.: Panola: a personal assistant for supporting users in preserving privacy. *ACM Trans. Internet Technol.* **22**(1), 1–32 (2021)
32. Wishart, R., Corapi, D., Marinovic, S., Sloman, M.: Collaborative privacy policy authoring in a social networking context. In: *2010 IEEE International Symposium on Policies for Distributed Systems and Networks*, pp. 1–8. IEEE (2010)
33. Woodgate, J., Ajmeri, N.: Macro ethics for governing equitable sociotechnical systems. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1824–1828 (2022)



Generalising Axelrod's Metanorms Game Through the Use of Explicit Domain-Specific Norms

Abira Sengupta¹, Stephen Cranefield¹, and Jeremy Pitt²

¹ University of Otago, Dunedin, New Zealand
abira.sengupta@postgrad.otago.ac.nz, stephen.cranefield@otago.ac.nz

² Imperial College London, London, England
j.pitt@imperial.ac.uk

Abstract. Achieving social order in societies of self-interested autonomous agents is a difficult problem due to lack of trust in the actions of others and the temptation to seek rewards at the expense of others. In human society, social norms play a strong role in fostering cooperative behaviour—as long as the value of cooperation and the cost of defection are understood by a large proportion of society. Prior work has shown the importance of both norms and metanorms (requiring punishment of defection) to produce and maintain norm-compliant behaviour in a society, e.g. as in Axelrod's approach of learning of individual behavioural characteristics of boldness and vengefulness. However, much of this work (including Axelrod's) uses simplified simulation scenarios in which norms are implicit in the code or are represented as simple bit strings, which limits the practical application of these methods for agents that interact across a range of real-world scenarios with complex norms. This work presents a generalisation of Axelrod's approach in which norms are explicitly represented and agents can choose their actions after performing *what-if* reasoning using a version of the event calculus that tracks the creation, fulfilment and violation of expectations. This approach allows agents to continually learn and apply their boldness and vengefulness parameters across multiple scenarios with differing norms. The approach is illustrated using Axelrod's scenario as well as a social dilemma from the behavioural game theory literature.

Keywords: Norms · Metanorms game · Expectation event calculus

1 Introduction

The conflict between social benefit and an individual's self-interest is a central challenge in all social relationships as individuals may put their own interests ahead of those of the society as a whole, often leading to a suboptimal outcome for all—a situation known as a *social dilemma*.

Understanding how societies can solve this conflict and achieve cooperation toward the collective good is essential. In a fishing scenario, for example, it is profitable for a fisherman to catch as many fish as possible, but if everyone is selfishly doing the same thing, the fishery will eventually run out of fish.

The question of why people often do cooperate with others remains, despite the fact that individuals may benefit more from defecting than from cooperating. The classical game theory literature from the last few decades models social dilemmas using payoff matrices or trees and solution concepts such as the Nash equilibrium, while making the assumption that all agents are perfectly rational and well-informed. However, this becomes intractable to reason about for a large number of agents.

Human society suggests that cooperation can occur due to internal and social motivations such as altruism, rational expectations (e.g. focal points), social choice mechanisms (e.g. voting and bargaining) and social norms [9].

One body of prior work has focused on the role of norms, providing evidence that social norms play an important role in fostering cooperation. Social norms imply that members of society should comply with prescribed behaviour while avoiding proscribed behaviours [15]. Bicchieri explains why agents adhere to social norms, claiming that a social norm emerges as a result of our expectations of others and beliefs about their expectations [4]. Axelrod uses an evolutionary computing approach to show how agents adopt normative behaviour after learning individual parameters of boldness and vengefulness, where their boldness represents their propensity to violate norms and vengefulness represents their inclination to punish others for violating norms [3]. However, his approach is based on an implicit representation of norms. The norms themselves play no role in his simulation. Instead the agents have hard-coded logic for using the boldness and vengefulness parameters to inform decisions about whether to cooperate or defect and whether to punish defectors and agents that observe defections but do not punish the defectors. This limits its practical use for agents that interact in a variety of real-world situations with a range of different norms.

In this work, we propose a generalisation of Axelrod’s method where norms are represented explicitly and agents can choose their course of action after engaging in *what-if* reasoning to compare the normative outcomes of alternative actions. This approach is significant because it enables agents to continuously learn and apply their boldness and vengefulness parameters across a variety of scenarios with various norms.

The following is the structure of the paper. Axelrod’s norms and metanorms games are discussed in Sect. 2. Section 3 emphasises the use of explicit norms to encode Axelrod’s mechanism. Section 4 depicts the results of generalisation of Axelrod’s norms and metanorms games, as well as the use of boldness and vengefulness in other scenarios. The prior event calculus models of norms are described in Sect. 5. Section 6 concludes the paper.

2 Background of Axelrod’s Model

Axelrod states that “the extent to how often a given type of action is a norm depends on just how often the action is taken and just how often someone is punished for not taking it”. To understand how cooperation emerges from norms, he developed a game in which players learn the parameters of boldness and vengefulness over generations of the population and can choose to deviate from the norms and metanorms, receiving punishment for their violations [3].

2.1 The Norms Game

Axelrod’s norms game follows an evolutionary model in which successful agent strategies propagate over generations. A strategy is a pair of values representing the agent’s boldness and vengefulness. Each agent has the option of defecting by violating a norm, and there is a chance of being observed by other agents with the probability S , which is drawn individually for each agent from a uniform distribution. Each of the agents has two decisions to make (Fig. 1a).

- Agents must decide whether to cooperate or defect based on their boldness value (B). A defecting agent (when $S < B$) receives a Temptation payoff ($T = 3$) while other agents receive a Hurt payoff ($H = -1$). If an agent decides to cooperate, no one’s payoff will change as a result.
- If an agent observes others defecting (as determined by the S value), the agent decides whether to punish those defectors based on its vengefulness (V) (a probability of punishment). Punishers incur an enforcement cost ($E = -2$) every time they punish ($P = -9$) a defector.

Axelrod simulated the norms game five times with 100 generations of 20 agents.¹ Between generations, the utilities of each agent are used to evolve the population of agents. Agents with scores greater than the average population score plus one standard deviation are reproduced twice in a new generation. Agents with a score less than the average population score minus one standard deviation are not reproduced. Other agents are only reproduced once.² The initial values of B and V are chosen at random from a uniform distribution of eight values ranging from $0/7$ to $7/7$, with the numerator represented as a 3 bit string. During reproduction each bit has a 1% chance of being flipped as a mutation.

¹ Axelrod used five runs of a hundred generations to simulate the norms and metanorms games. However, we follow the recommendation of [7] and use 100 runs.

² Axelrod does not state how he maintains a fixed population size after applying these reproduction rules. We follow the approach of [7] involving random sampling when the new population is too large, and random replication when the population is too small.

2.2 The Metanorms Game

Axelrod found that norms alone were not sufficient to sustain norm compliance in society. He therefore introduced a *metanorm* to reinforce the practice of punishing defectors. The metanorms game includes punishment for those agents who do not punish defectors after observing them defect (Fig. 1b). Metapunishers incur a meta-enforcement cost ($E' = -2$) every time they metapunish ($P' = -9$).

3 Generalising Axelrod’s Metanorms Game Using Explicit Norms

To generalise Axelrod’s metanorms game, we provide an explicit representation of norms and a mechanism that can compare alternative actions to determine which will lead to a norm violation. The expectation event calculus (EEC) [5], a discrete event calculus extension, provides this capability.

3.1 The Expectation Event Calculus

The event calculus (EC) consists of a set of predicates that are used to encode information about the occurrence of events and dynamic properties of the state of the world (known as fluents), as well as a set of axioms that interrelate these predicates [14]. This logical language supports various types of reasoning. In this work, we use it for *temporal projection*. This takes as input a narrative of events that are known to occur (expressed using *happensAt*(E, T), where E is an event and T is a time point) and a domain-specific set of clauses defining the conditions under which events will *initiate* and *terminate* fluents (expressed using the predicates *initiates*(E, F, T) and *terminates*(E, F, T)). The EC axioms are then used to infer what fluents hold at each time point. By default, fluents are assumed to have *inertia*, i.e. they hold until explicitly terminated by an event.

The EC, in general, assumes that time is dense, and time points are ordered using explicit ‘<’ constraints. In this work, we use the *discrete event calculus* (DEC), which assumes that time points are discrete and identified by integers [11].

The expectation event calculus (EEC) [5] is an extension of the DEC that includes the concepts of expectation, fulfilment, and violation. Expectations are constraints on the future, expressed in a form of linear temporal logic, that the agent wishes to monitor. Expectations are free from inertia and instead are automatically *progressed* from one state to the next, which means they are partially evaluated and re-expressed in terms of the next time point. During progression, if they evaluate to true or false, a fulfilment or violation is generated.

Figure 2 illustrates temporal projection in the EEC. In addition to the standard features of the DEC, there are two special kinds of fluents: *exp_rule* and *exp*. A conditional rule to create expectations is expressed by an *exp_rule*($Cond, Exp$) fluent. Here, $Cond$ is a condition on the past and/or present, while Exp represents the future expectation. Exp will be expected if $Cond$ holds, in which case

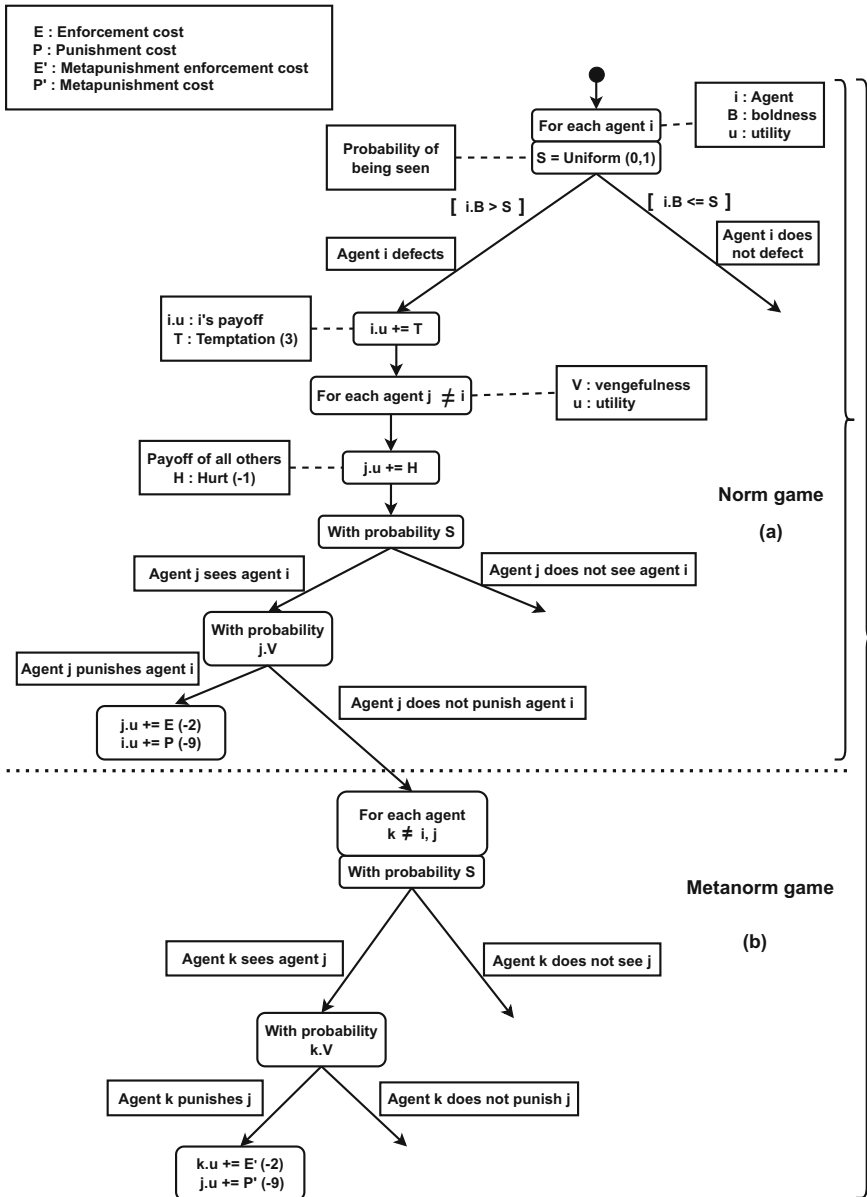


Fig. 1. a In Axelrod's **norms game**, agent i will defect if bold enough; otherwise, agent i will cooperate. Another agent j will punish i if the defection is observed and agent j is vengeful enough. **b** **The metanorms game** adds the possibility of metapunishment of agent j by another agent k . This occurs if j sees a defection from i , j does not punish i , this lack of punishment is observed by k and k is vengeful enough.

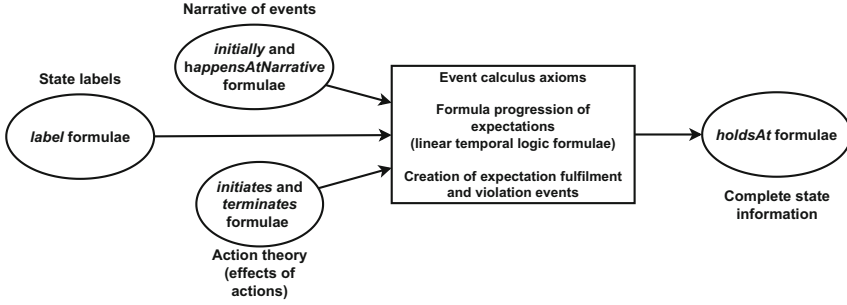


Fig. 2. Overview of reasoning in the expectation event calculus (EEC) [5].

an $exp(Exp)$ fluent is created. In our implementation of the EEC, the condition can test for fluents holding, the occurrence of events (expressed using $happ(E)$) and the presence of a symbolic label L in a state (using the expression $@L$). Complex expressions involving conjunctions and linear temporal logic operators such as *next*, *eventually*, *always* and *never* can also be used. Labels are associated with time points using $label(L, T)$ declarations, and are not required to be unique. To distinguish between basic events in the narrative and the inferred fulfilment and violation events, we use the predicate $happensAtNarrative$ to declare the narrative events.

In contrast to the earlier approach of CraneField [5], we represent fulfilments and violations as events rather than fluents, denoted $fulf(Cond, Exp, T, Res)$ and $viol(Cond, Exp, T, Res)$, where $Cond$ and Exp are the condition and expectation of an expectation rule that was triggered at time T to create the expectation, and Res is the residual expectation (after being progressed zero or more times since its creation) at the time of its fulfilment or violation.³

Our EEC implementation includes a *what-if* predicate that accepts two alternative event lists (E_1 and E_2) as arguments and infers the fluents that would hold and the events (including violation and fulfilment events) that would occur if E_1 or, alternatively, E_2 occurred at the current time point. It returns the fluents and events that would occur if the events in E_1 are performed but not those in E_2 , and those that would occur if the events in E_2 occur but not those in E_1 . This can be used as a basic form of look-ahead to assist an agent in deciding between two alternative sets of actions. In particular, in this work we consider options that are singleton lists and use the *what-if* predicate to compare which (if any) of two actions will cause one or more expectation violations.

3.2 Modelling Axelrod’s Scenario with the EEC

We model time as a repeated cycle of steps and associate an EEC label with each step. We use the event calculus *initiates* and *terminates* clauses to define the

³ There is also an extended version of the exp fluent with these four arguments—the version used in this paper has only the residual expectation as its argument.

effects of events that update an agent's S value, give payoff to an agent as the outcome of all agents' cooperate or defect actions, and punish and metapunish agents.

We use the EEC within a simulation platform [6] that integrates Repast Symphony [12] with the EEC through queries to SWI Prolog. This includes an institutional model in which agents take on roles by asserting to the EEC narrative that certain institution-related events have occurred, such as joining an institution and adding a role. Each role has an associated set of conditional rules. A rule engine⁴ is run at the start of selected simulation steps when agents must choose an action and these role-specific rules recommend the actions that are relevant to the agent's current role based on queries to the EEC, e.g. to check the current step's label and the fluents that currently hold. Then the agent can run scenario-specific code to select one of the actions to perform.⁵

In contrast to Axelrod's implicit representation of a norm and metanorm, our explicit representation of a norm implies that three norms are required. In the metanorms game, each action choice is governed by a norm. As there are three choice-points, there are three norms that we model using *exp_rule* fluents.

First-Order Norm

```
initially(
  exp_rule(member(A, society),
    never(happ(defect(A))))).
```

This *initially* clause creates an expectation rule (*exp_rule*) expressing the *first-order norm*, which states that no defection should occur for any agent who is a member of the society.

As the first-order norm described above is likely to be insufficient to motivate selfish agents to follow the norm and cooperate with others, a *second-order norm* is required.

Second-Order Norm

```
initially(
  exp_rule(and([sawViolation(B,A,R,_),
    pl(contains_term(defect(A), R))]),
    happ(punish(B,A)))).
```

The *second-order norm* states that if the *first-order norm* is violated by an agent, another agent who observes the violation should punish the first-order norm defector. The *pl* term in the rule's condition indicates a goal to be evaluated using Prolog.

⁴ <https://github.com/maxant/rules>.

⁵ At present we assume there are no more than two relevant actions.

This second-order norm is triggered by a *sawViolation* fluent, which is created when a violation of the first-order norm occurs and a defector is observed. The following *initiates* clause creates this fluent.

```
initiates(viol(_,_ ,ResidualExp),
          sawViolation(B,A,ResidualExp,T),
          T) :-
    responsible(ResidualExp, A),
    agent(B),
    B \== A,
    holdsAt(s(A,S), T),
    random(R),
    R < S.
```

The condition for this clause first determines which agent is responsible for the unfulfilled expectation, then generates a possible observer different from the violator and compares that agent's *S* value with a random number to determine whether or not the violation has been observed.

In our application, the violated expectation will include an instantiation of one of the event terms *defect(A)*, *punish(A)* or *metapunish(A)*, and we can use these to identify the responsible agent. We therefore define the *responsible* predicate in Prolog as follows.

```
responsible(Expectation, A) :-
    contains_term(defect(A), Expectation).

responsible(Expectation, A) :-
    contains_term(punish(A,_), Expectation).

responsible(Expectation, A) :-
    contains_term(metapunish(A,_), Expectation).
```

To encourage the punishment of second-order norm violators, a *third-order norm*, is required.

Third-Order Norm

```
initially(
    exp_rule(and([sawViolation(B,A,R,_),
                 pl((contains_term(punish(A,C), R), B \== C))]),
             happ(metapunish(B,A)))).
```

According to this EEC rule, observer agents are expected to metapunish the violators of the second-order norm when the violating agents fail to punish the first-order norm defector after observing their defection.

Figure 3a, b illustrate the differences between our implementations of the metanorms game with implicit and explicit norms. Figure 3a makes hard-coded

Table 1. Roles and their possible actions

<i>Step</i>	<i>Role</i>	<i>Possible actions</i>
Cooperate or defect	Temptation role	Cooperate or defect
Punishment	Possible punisher role	Punish or do not punish
Metapunishment	Possible punisher role	Metapunish or do not metapunish

action choices, following Axelrod’s algorithm. However, in Fig. 3b, whenever there is an action choice to be made, the two action choices are compared using *what-if* reasoning that is informed by one of the three norms. For the decision between cooperation or defection, if only one of the action choices will cause norm violation, the agent’s boldness parameter is used to decide whether the violating option (cooperation) is chosen. For decisions between (meta)punishment or no punishment, the vengefulness parameter (V) is used: the violating option is chosen with probability $1 - V$.

The box labelled “Cooperate or defect” indicates a time step in which an agent must decide whether to cooperate or defect based on whether it has a high boldness value (Fig. 3a). In Fig. 3b, the first-order norm’s *exp_rule* expressing the first-order will have been created in the initial time step and triggered once for each agent, resulting in *exp* fluents stating that each agent should never defect. Therefore, an agent can use the *what-if* mechanism to compare the outcomes of the two alternative actions, cooperate or defect. If the agent’s boldness value exceeds S , the agent will violate the first-order norm by defecting; otherwise, the agent will cooperate.

Figure 3a shows how the punishment step of the implicit norm simulation hard-codes the decision to punish each observed defector with a probability given by an agent’s vengefulness parameter. In contrast, the explicit norm representation simulation cycle in Fig. 3b shows how *what-if* reasoning informed by the explicit second-order norm detects that failure to punish will cause a norm violation. That norm-violating option is then chosen with probability $1 - V$.

In the metapunishment step of the implicit norm simulation, if an agent chose not to punish an observed defector in the punishment step, then every agent with sufficiently high vengefulness will metapunish that agent (Fig. 3(a)). However, when using explicit norms, (Fig. 3b), *what-if* reasoning detects that failure to metapunish will cause a violation of the explicit third-order norm, and this option will be chosen with probability $1 - V$.

Figure 4 depicts in more detail our use of explicit norms with agents that are aware of norm violations. We have three norms represented by *exp_rule* fluents, which are triggered at different time steps. After triggering, each *exp_rule* fluent creates an expectation. The EEC *initially* clauses generated the *exp_rule* fluents for the first-order norm (N_1), second-order norm (N_2), and third-order norm (N_3). Each agent has two roles: *temptation role* and *possible punisher role*. Table 1 shows what actions an agent can take in the simulation when assigned to a specific role for each step. The *temptation role* specifies that an agent can

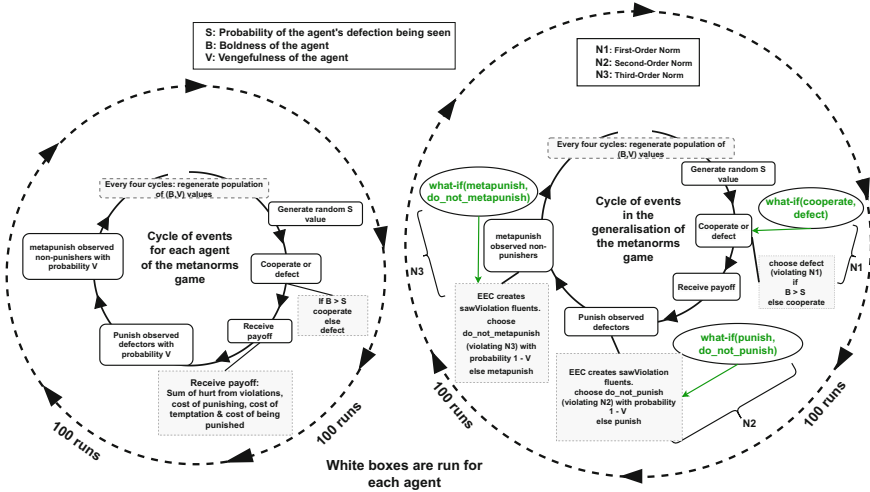


Fig. 3. The distinction between implicit and explicit metanorms. The cycle of events for Axelrod’s metanorms game is shown in **a** its original form with implicit norms, and **b** in our generalised form with explicit norms.

choose to cooperate or defect at the *cooperate or defect* step. At the *punishment* step an agent with the *possible punisher* role can choose to punish or do not punish, and an agent with the *possible punisher* role can choose to metapunish or not to metapunish at the *metapunishment* step. At the initial time step of the simulation, both roles (*temptation role* and *possible punisher role*) are activated for each agent.

The EEC *what-if* predicate is used to consider two options: cooperate or defect, punish or do not punish, metapunish or do not metapunish (depending on the current step in the simulation cycle), and determine whether one option produces a violation while the other does not. The non-violating option is then chosen (or a random choice if there is no violation). If both options result in a violation, the cost of each violation is calculated (using domain-specific knowledge) and the less costly option is chosen. If the costs are the same, a random selection is made.

At the simulation’s final step, regenerate.population, successful agents are replicated and mutated to form a new generation of the same size [7]. In this simulation, folded outlined arrows represent iteration: one for the three norms and their corresponding expectations within one generation, and the other for 100 generations of simulation.

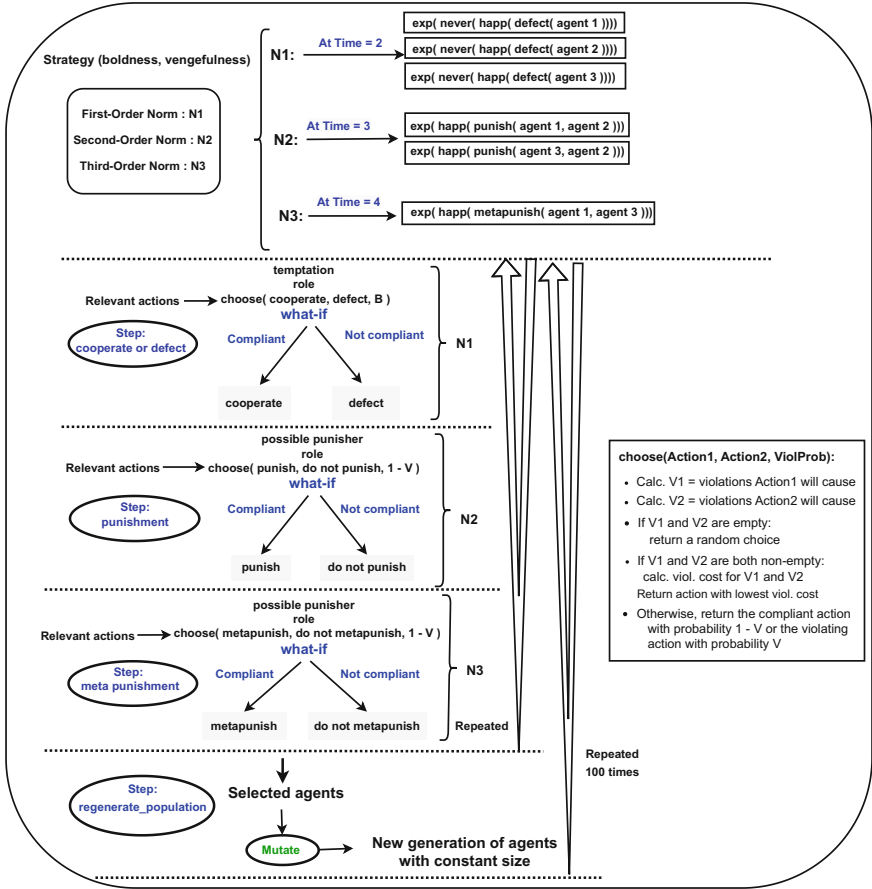


Fig. 4. The generalisation of Axelrod's approach in which norms are explicitly represented and agents can choose their actions based on *what-if* reasoning using expectation event calculus, which tracks the creation, fulfilment, and violation of expectations. For illustration, in the top section, we assume there are three agents. All agents are expected never to defect under the first-order norm, but we assume that agent 2 chooses to defect. According to the second-order norm, agents 1 and 3 are expected to punish agent 2. Then, according to the third-order norm, if agent 1 notices that agent 3 did not punish agent 2, agent 1 should punish agent 3.

4 Experiments

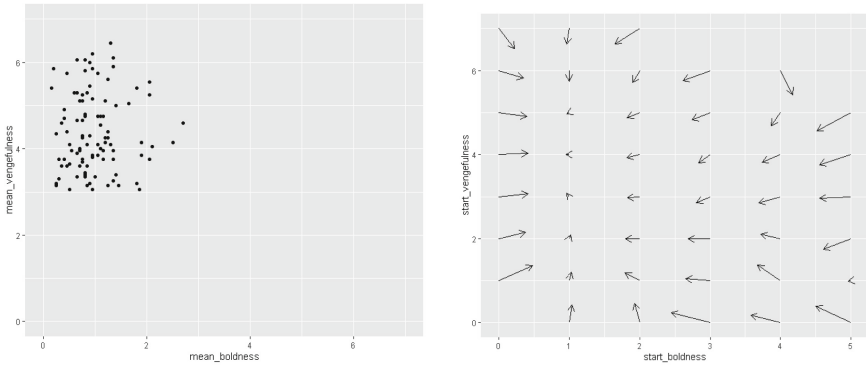


Fig. 5. a Scatter plot of mean boldness along x-axis wise and mean vengefulness along y-axis wise of generalisation of Axelrod’s study with all three norms. **b Vector plot representation** of mean boldness and vengefulness.

Experiment 1: Generalisation of Axelrod’s metanorms game This experiment illustrates that our generalised metanorms game with explicit norms generates punishment and metapunishment events that sustain norm-compliant behaviour. We used 20 agents, 100 generations and 100 runs. Figure 5a shows a scatter plot of the mean boldness and vengefulness values at the end of each run, and we observe that mean boldness is always low and mean vengefulness ranges from high to average.

Figure 5b depicts the vector representation of the same data set.⁶ Vectors show how boldness and vengefulness change across generations in the population. The results show that *what-if* reasoning with explicit norms causes the first-order norm to be largely upheld in the society due to low boldness being maintained.

Experiment 2: Using the boldness and vengefulness in another scenario Klein [10] introduced a scenario that we refer to as the plain-plateau scenario in our previous work [13]. The scenario depicts a society in which people have the option of living on a river plain with easy access to water, otherwise, they can live on a plateau. Flooding is a risk for river-plain residents. When the government has complete discretionary power, it is in the government’s best interests to compensate citizens whose homes have been flooded by taxing citizens who live on the plateau, creating a prisoner’s dilemma situation. In our previous work, we experimented with the use of social norm-based expectations to achieve coordination where citizen agents are hard-coded to prefer actions that will result in no violation.

⁶ Populations with similar average levels of boldness and vengefulness are grouped together to create each vector. The end point of each arrow shows the average levels of these features one generation later [3].

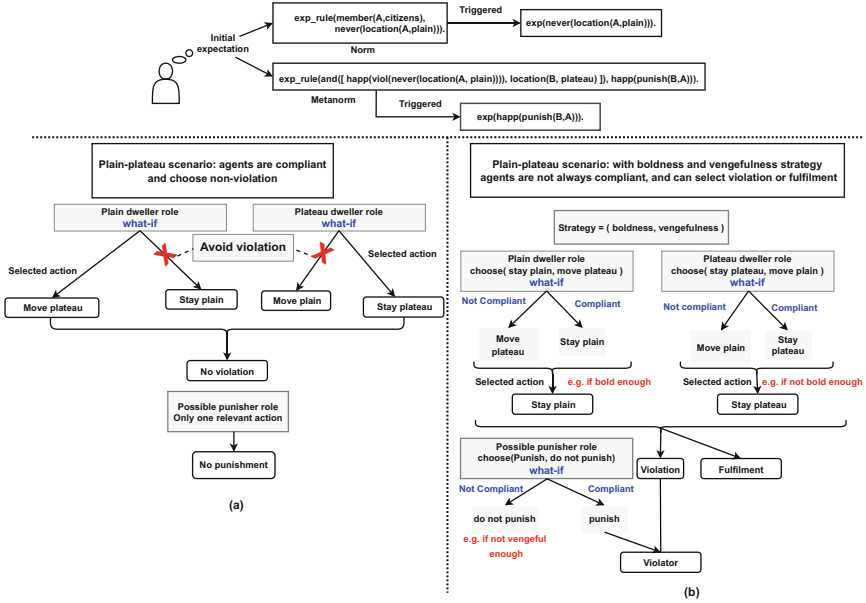


Fig. 6. a The plain-plateau scenario in which agents are hard-coded to always choose the non-violating actions. **b The plain-plateau scenario** as an application of our generalised metanorms game.

Figure 6a illustrates this prior work. Each agent has either a plain dweller or a plateau dweller role, and in each simulation cycle there are two choices: agents can stay on the plateau or move to the plain. In this scenario, we assume there exists a norm that no one should live in the plain and a metanorm stating that plateau dwellers should punish those who live on the plain.

Figure 6b illustrates the application of our generalisation of the metanorms game to this scenario.⁷ This simulates an agent finding itself in a new scenario after having already evolved its personality with respect to norms, and illustrates the generality of our approach using explicit norms and *what-if* reasoning. In the plain-dweller role, the EEC *what-if* predicate is used to consider two options: move to the plateau or stay in the plain; similarly, in the plateau-dweller role, the *what-if* mechanism is used to consider either move plain or stay plateau. When agents with high boldness in the plain-dweller role choose to stay in the plain, they violate the norm. When other vengeful agents observe violators, they punish them, unless insufficient vengefulness causes them to violate the metanorm.

We simulated a group of agents who encounter the plain-plateau scenario after evolving their boldness and vengefulness parameters in the Axelrod scenario. From the first run of Experiment 1, we randomly sampled six (boldness,

⁷ We do not include a third-order norm for the plain-plateau scenario as this was not part of Klein’s model [10].

vengefulness) pairs from personalities of the twenty agents at the end of that run. This resulted in six agents with a boldness of zero.⁸

We used the expectation-aware action-selection mechanism generalised version of Axelrod’s metanorms game with the input norms replaced with those shown at the top of Fig. 6. As before, boldness was used as the probability of choosing a violating action over a non-violating one, and $1 - \text{vengefulness}$ as the probability of choosing to violate an expectation to punish. As all six agents had a boldness of zero, they all opted to live on the plateau (which counts as cooperation in this scenario) whenever they had an opportunity to change their location. Therefore, there were no norm violations and no expectations to punish were created.

While this cooperative behaviour was not surprising given the lack of boldness of the simulated agents, this was an emergent outcome from the transfer of attitudes to normative behaviour learned in the previous scenario, a uniform action-selection mechanism that can be used across scenarios, and the ability to provide new norms as symbolic inputs to inform this mechanism.

5 Prior Event Calculus Models of Norms

This section of the paper reviews some research on the use of event calculus in autonomous agent reasoning to examine the effects of norms.

Alrawagfeh [1] suggests formalising prohibition and obligation norms using event calculus and offers a method for BDI agents to reason about their behaviour at runtime while taking into account the norms in effect at the time and previous actions. Norms are represented by EC rules that initiate fluents with special meanings. The introduced fluents represent punishments for breaking a prohibition norm or failing to fulfil obligation norms, or the rewards for fulfilling obligation norms. The normative reasoning strategy assists agents in selecting the most profitable plan by temporarily asserting to the event calculus the actions that each plan would generate and considering the punishments and/or rewards it would trigger.

In Alrawagfeh’s work, norms cannot be changed dynamically without changing the event calculus rule base, because they are defined by EC initiates clauses. In contrast, in our approach, EC rules can be instantiated automatically from *exp_rule* fluents, which can be changed dynamically by events.

Alrawagfeh has no representation of active norms, violations or fulfilments: only punishments and rewards. In our work, expectation creation, fulfilment, and violation are represented as events, and the *what-if* predicate compares alternative events to track expectation creation, fulfilment, and violation. We do not assume that rewards and/or punishments will always follow violations and fulfilments; these could be defined by separate *exp_rules* or EC initiates clauses.

Hashmi et al. [8] propose a number of new EC predicates to allow them to model different types of obligation that occur in legal norms. In particular, they

⁸ Run 1 of Experiment 2 ended with only one agent with a non-zero boldness value: 4/7.

introduce a *deontically holds at* predicate that ensures an obligation enters into force at the same time that the triggering event occurs. In contrast, our approach using the EEC does not necessitate the introduction of a new type of EC predicate in order to initiate a deontic predicate. An *exp_rule* or an expectation can be created with a standard initiates clause and an *exp* fluent is created by an *exp_rule* in the state where the condition of the rule becomes true. The EEC does, however, include additional axioms to handle the progression of expectations.

Alrawagfeh and Hashmi et al. both use standard EC, whereas we use discrete EC because this work involves discrete time simulations.

6 Conclusion and Future Work

In previous work [13], we used the EEC *what-if* mechanism for choosing actions in the presence of expectations. However, we assumed that all agents are compliant and will always choose a non-violating action if possible. The current work removes this assumption, but it also makes the following significant standalone contribution: it generalises Axelrod's metanorms game to use explicitly represented norms. This allows the metanorms game to be used across multiple scenarios.

Applying our generalised version of Axelrod's metanorms game to varying scenarios will require changing the mechanism for evolving boldness and vengefulness parameters. Strategy evolution through population regeneration is not realistic for agents that continually evolve their boldness and vengefulness as they move between different scenarios. Therefore, in future work we will investigate the use of a pairwise comparison approach where an agent may adopt another agent's strategy based on a comparison of their respective fitnesses, e.g. by using the Fermi process [2].

Acknowledgements. This work was supported by the Marsden Fund Council from New Zealand Government funding, managed by Royal Society Te Apārangi.

References

1. Alrawagfeh, W.: Norm representation and reasoning: a formalization in event calculus. In: International Conference on Principles and Practice of Multi-Agent Systems, pp. 5–20. Springer (2013)
2. Altrock, P.M., Traulsen, A.: Fixation times in evolutionary games under weak selection. *New J. Phys.* **11**(1), 013012 (2009)
3. Axelrod, R.: An evolutionary approach to norms. *Am. Polit. Sci. Rev.* **80**(4), 1095–1111 (1986)
4. Bicchieri, C.: *The Grammar of Society: The Nature and Dynamics of Social Norms*. Cambridge University Press (2005)
5. Cranefield, S.: Agents and expectations. In: International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems, pp. 234–255. Springer (2013)

6. Cranefield, S., Clark-Younger, H., Hay, G.: A collective action simulation platform. In: *Multi-Agent-Based Simulation XX: 20th International Workshop, MABS 2019, Montreal, QC, Canada, May 13, 2019, Revised Selected Papers 20*, pp. 69–80. Springer (2020)
7. Galan, J.M., Izquierdo, L.R.: Appearances can be deceiving: Lessons learned re-implementing Axelrod’s evolutionary approach to norms. *J. Artif. Soc. Soc. Simul.* **8**(3) (2005)
8. Hashmi, M., Governatori, G., Wynn, M.T.: Modeling obligations with event-calculus. In: *Rules on the Web. From Theory to Applications: 8th International Symposium, RuleML 2014, Co-located with the 21st European Conference on Artificial Intelligence, ECAI 2014, Prague, Czech Republic, August 18–20, 2014*, pp. 296–310. Springer (2014)
9. Holzinger, K.: The problems of collective action: A new approach. *MPI Collective Goods Preprint No. 2003/2*, SSRN (2003). <https://doi.org/10.2139/ssrn.399140>
10. Klein, D.B.: The microfoundations of rules vs. discretion. *Constit. Polit. Econ.* **1**(3), 1–19 (1990)
11. Mueller, E.T.: *Commonsense Reasoning*. Morgan Kaufmann (2006)
12. North, M.J., Collier, N.T., Ozik, J., Tatara, E.R., Macal, C.M., Bragen, M., Sydelko, P.: Complex adaptive systems modeling with Repast Simphony. *Complex Adapt. Syst. Model.* **1**(1), 3 (2013)
13. Sengupta, A., Cranefield, S., Pitt, J.: Solving social dilemmas by reasoning about expectations. In: *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIV*, pp. 143–159. Springer (2022)
14. Shanahan, M.: The event calculus explained. In: *Artificial Intelligence Today*, pp. 409–430. Springer (1999)
15. Thøgersen, J.: Social norms and cooperation in real-life social dilemmas. *J. Econ. Psychol.* **29**(4), 458–472 (2008)



Incentivising Participation with Exclusionary Sanctions (Full)

Buster Blackledge^(✉), Antonios Papaoikonomou, Matthew Scott, Asimina Mertzani, Noan Le Renard, Hazem Masoud, and Jeremy Pitt

Imperial College London, London SW7 2BX, UK
{bb922,ap3422,mss2518,s20,nml18,hm1822,jpitt}@ic.ac.uk

Abstract. Some cooperative survival situations require all members of a group to participate equally in collective action; however, if the only sanction for non-participatory free-riding is exclusion, then it can be ineffective, as exclusion is indistinguishable from non-participation. The question then is: how does a group, that can define a set of mutually agreed conventional rules, incentivise participation that supports collective survival when the only sanctioning instrument is exclusion. This problem is investigated in this paper through the design and implementation of a self-organising multi-agent simulator for an iterated cooperative survival game. A series of experiments, or ‘survival trials’, is run for three different sanctioning schemes: fixed-length, dynamic-length and graduated-length exclusion. Results show that graduated sanctions outperform the alternatives, which can be either too weak or too strong. We conclude by arguing that these results provide evidence for why graduated sanctions are the basis for one of the principles of self-governing institutions.

Keywords: Multi-agent system · Cooperative survival games · Collective action · Social contracts · Sanctions · Governance

1 Introduction

Collective action games are a type of game where a player’s decisions impact the welfare of the collective as a whole, and all players must work together for a common goal. In these games, there is often an element of cooperative survival, where individuals within the scenario must act in the interest of social welfare, despite their own self interest [18], in order to withstand a disaster. Often, with these games, it is the case that if one player dies, all players die.

These survival games can be considered as a form of extreme, high-stakes common pool of resources (CPR) problem, with the players themselves serving as the common pool. Here, the actions of individuals have consequences for all, such as with the ‘tragedy of the commons’ [9], where individuals have access to a shared resource that is susceptible to depletion if not properly managed. In this

problem, socially oriented traits are necessary to ensure the long-term survival and success of the group. As such, understanding cooperative survival strategies, and how to incentivise collectivism, is crucial for solving complex problems and achieving sustainability in a range of industries and applications.

For players to successfully traverse these game environments, predicated upon the scarcity of life-sustaining resources, they must embody a number of principles for the management of communities and the common resource in the form of self-governing institutions [19].

This paper investigates the dilemma in developing such an institution in the absence of any central authority. When the only form of legislature is the social construction of social contracts, and the only sanction for non-participation or breaking these contracts is exclusion, then it can be ineffective because exclusion is indistinguishable from non-participation. Moreover, in a high-stakes cooperative survival game, non-cooperation is beneficial in the short-term because the risk of instantaneously dying is eliminated; however it is detrimental to the common good as it increases the probability of the collective dying out sooner than if everyone participated.

The structure of this paper is as follows. We begin by summarising the background of this research in Sect. 2 by looking at elements of survival games, institutional power, sanctions and social contracts. Following this, we discuss the implementation of the game in Sect. 3, and the self-organising mechanisms used to solve it in Sect. 5. Subsequently, we conclude that a simulator is best used to solve this game, which is formalised in Sect. 6.

In order to examine the effectiveness of sanctioning in such a game, a set of experiments are designed in Sect. 7 which investigate the survivability of the collective under three different sanction designs: fixed-length, dynamic-length and graduated-length. Here, we conclude that by varying the duration of fixed-length sanctions, a point can be found where survivability is maximised. We also conclude that introducing dynamic and graduated sanctions solves the issue of poor survivability with very low- and high-duration sanction, with graduated sanctions permitting sanctions of effectively maximum length. The simulation platform was developed using GoLang and can be accessed at <https://github.com/antonyypap/SOMAS2022>.

2 Scenario and Background

For the purposes of this paper, we consider a social dilemma where a group of players start at the bottom of a pit, each level of which contains an enemy to be fought. The group must battle and defeat the enemy before they can ascend to the next level, however, any deaths incurred on the way reduce the group’s ability to defeat increasingly strong enemies. With each enemy defeated, players get access to a stash of *loot*, containing weapons, shields and potions, which can be divided amongst the group. Weapons are used to attack the enemy, shields are used to defend against the enemy, and potions are used to regenerate health.

Furthermore, this game is designed to be played in an economy of scarcity, meaning that the allocation of loot cannot fully satisfy all of the players' individual desires, leading to biased decision formations, reinforced by increased individual utility [4,8]. This condition sets the stage for Ostrom institutions to be formalised for solving a common pool of resources (CPR) game [21], within a norm governed society. These societies take into account the permissions and obligations of its members, as well as the possibility of a deviation from the expected action [1], creating a framework for: sanctions, forgiveness to inspire reconciliation and defiance to incite change [23].

These social norms can be formalised by social contracts, which specify the conditions under which these norms must be obeyed. It has been shown that it is always theoretically possible to design an optimal social contract for the moral imperative [6], although designing this contract is often not a trivial task [24].

As well as defining the conditions by which the social contract must be obeyed, the contract also defines the punishment for not doing so. The breaking of a contract often merits a *sanction* [20], which comes as a detriment to the disobedient actor involved. Such sanctions can vary drastically in severity, such as with their duration, so must be carefully constructed, since "unfair sanctions" [7] can have detrimental impacts on human co-operation. To prevent this, designing effective sanctions has seen a computational approach [2,17]. In this scenario where sanctions entail exclusion, a negative feedback loop is formed, where sanctioning a defector becomes detrimental to the collective. It is important to prevent free-riders from appropriating the shared resource yet refusing to fight (the risk-averse approach), however over-exclusion will leave them more susceptible to damage, thereby hindering the possibility of co-operative survival. Drawing on the Ancient Greek democratic procedure of Ostracism, which sought to banish tyrannical members of society, damage can be minimised by deposing any unjust institutions who punish defectors with biased sanctions [22].

Social choice theory unifies the relationship between a collection of individual preferences and the final decision of the community [5,14]. Should these individual preferences be influenced by weighted social knowledge predicated upon reputation, the resulting scenario is an economy of esteem [3], where this reputation is a non-tradeable commodity and cannot be influenced by ones starting position or wealth in a heterogeneous society.

There are various frameworks available to guide decision-making. One such framework is *Preference Utilitarianism*, a contemporary philosophy that seeks to maximize actions that serve the interests of all actors involved [10]. This differs from the conventional "greatest happiness" utilitarian principle [15], as it emphasizes the importance of recognizing the interests of others. In our case, due to an environment of scarcity, it is to be expected that players' personal preference will be to independently accumulate resources. This would entail a lack of recognition of the other, which is essential to morality and ethics [11]. However, despite their condition, we can hope that through knowledge aggregation and collective action, that they can embody preference utilitarianism, by acknowledging that their social network share the same collective interests.

3 Game Design

This game consists of two main phases—a *battle* phase and a *self-organisation* phase—which occur each *level* (l). The battle phase has each player combat the enemy, which runs iteratively until either the enemy is defeated, allowing the players to progress to the next stage, or the players lose (they are killed by the enemy), causing the game to end. If a battle round is victorious, players will progress to the self-organisation phase and subsequently move up a level. The game is completed when the final level is reached (all enemies have been defeated), resulting in a win, or all players have died, resulting in a loss. A compact formalisation of the system architecture is shown in Fig. 1, where “S.O. Phase” is an abbreviation for the *self-organisation* phase.

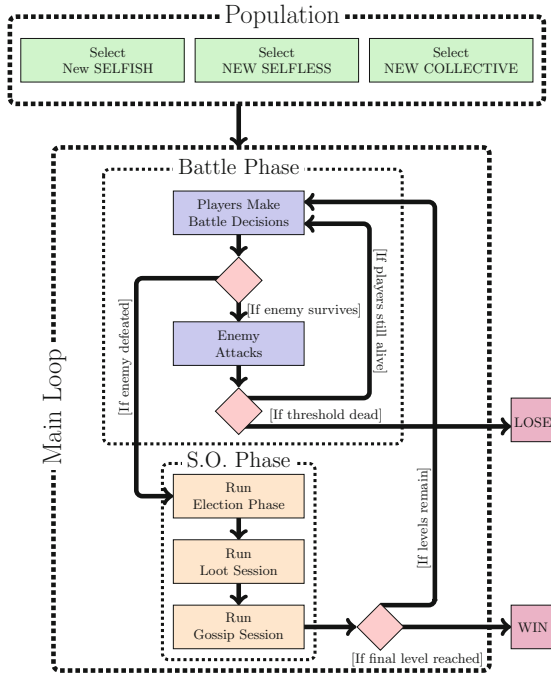


Fig. 1. System architecture for the co-operative survival game

3.1 Entities

We envision three main entities in this game: (1) the *Players*, (2) the *Enemy* and (3) the *Loot*. The *attributes* that these entities possess is shown in Table 1.

These entities are used throughout the game in both the *battle phase* and the *self-organisation* phase. We begin by introducing the *battle phase*, below.

Table 1. List of attributes for each entity

Player	Enemy	Loot
Health (HP)	Resilience (X)	Sword value (A_s)
Stamina (ST)	Damage potential (Y)	Shield value (D_s)
Attack (A)		Health potion (P_h)
Defence (D)		Stamina potion (P_s)

4 Battle Phase

In this section, we discuss the different stages in the battle phase, as well as the mathematics behind the enemy.

4.1 Game Stages

Within each battle round, players have three actions that they can perform: *fight*, *defend* and *cover*. Fighting deals damage to the enemy, defending absorbs damage from the enemy and cowering skips the fight round in order to regenerate *stamina* and *health*. *Stamina* is reduced whenever a fight action is performed by the corresponding attribute value, for example, performing a fight action reduces ST by A , so players may only perform an action so long as they have $ST \geq (A \vee D)$. Cowering requires no stamina to perform.

Equipping a loot item affects a player's attributes. *Potions* and *weapons* increase the corresponding attribute by their value. For example: equipping a *sword* increases A by A_s , whilst equipping (drinking) a *health potion* increases HP by P_h . The pit contains a set number of levels (L) and ends if all agents die, or they complete the final level. The rules for battle rounds are as follows:

Rule 1. An enemy dies if the aggregated attack value of the attacking players is above X at the end of a round.

Rule 2. A player dies if the damage received is higher than their remaining HP .

Rule 3. If the enemy is not defeated on a given battle turn, it attacks dealing $Y - \sum_i D_i$ (for all defending players) damage divided equally amongst all battling players.

Rule 4. If all players cover, they all receive equal damage of $\frac{Y}{N_A}$, where N_A represents the number of players alive on that level.

4.2 Enemy Formulation

Each player starts with a pre-determined level of health and stamina. Health is only depleted when damage is received from an enemies attack, whilst stamina

is depleted with every fight action performed. Both of these attributes are regenerated by their corresponding potion, or through the action of cowering, where agents recover 1% of their starting value.

The enemy attributes: Resilience, X (Eq. 1), and Damage Potential, Y (Eq. 2), are designed to linearly increase throughout the course of the game. They are also both dependent on: the starting health (HP) and stamina (ST) of the players, the number of players (N), as well as the total number of levels in the pit (L). This helps to maintain the difficulty of the game irrespective of the starting position.

$$X = \delta \frac{N * ST}{L} \sigma \quad (1)$$

$$Y = \delta \frac{N * (HP + ST)}{L} \sigma \quad (2)$$

δ represents a modifier in the range $[0.8, 1.2]$ to add non-determinism to the each separate calculation, σ denotes the linearly increasing scalar (Eq. 3).

$$\sigma = \frac{c}{L} + 0.5 \quad (3)$$

where c is the current level. As with the agent fight action values, the damage dealt by the enemy is scaled by a modifier in the range $[0.5, 1]$ applied to the damage potential.

The values given to swords, shields and potions are dependent on the strength of the defeated enemy. Their equations are not included for simplicity. Finally the total quantity of loot dropped is dictated by a pre-determined percentage of N_{init} , the initial number of players in the game, to ensure an economy of scarcity.

5 Self-organisation Phase

Following the conclusion of a victorious battle round, the game continues with four, successive self-organisation stages:

1. Players may *Gossip* to perform knowledge distribution and aggregation.
2. A vote of No-Confidence is cast to depose the current *Chair* if successful.
3. Elections are held to select a community *Chair*.
4. Players vote on proposed Social Norm contracts to select a new Social Norm.

Participation in these stages is optional for all players, and each stage is discussed in the following sections:

5.1 Gossip, Governance and Social Contracts

Exchanging *gossip* messages is the self-organising mechanism used for knowledge aggregation. This stage of the self-organisation phase allows players to share information about other players by sending a message to a discrete set of recipients. Players then have the ability to update their social perceptions based on this information, however, we consider *false gossip* to be outside of the scope.

The next self-organisational stage allows players to elect a leader, called the *Chair*, who gains *Institutional Powers* as follows: The chair can select and broadcast proposals for voting, and impose sanctions on defectors from the social contract by denying them access to the common pool resource. These powers allow a chair to introduce a bias towards their trusted agents.

Each player has the opportunity to submit themselves for consideration. If elected, their tenure lasts for a maximum of 30 levels, however, at the end of each level, their rule is subject to a no-confidence vote. If a majority is reached, the chair is deposed and a new leader is chosen, this makes leadership strategies a balance between bias and maintaining popularity. Introducing reigns that persist over multiple levels combats the initial transient behaviour within norm governed systems, where the effect of new rulers is not felt of the first few iterations [13].

Social contracts provide a set of mutually agreed conditions under which players must perform certain actions. Each player has the opportunity to create and submit a potential contract, known as a *proposal*, to the elected chair. Each *Battle Contract* contains all four player attributes: *HP*, *ST*, *A* and *D*, and an associated value for each, as well as a specified action: *attack* or *defend*. This value represents a threshold, with any attribute value over this deeming it ‘active’. If all attributes are ‘active’, a player is obligated to perform the action specified in the contract.

Once a proposal is accepted and the contract is created, players can calculate their required battle action. However, should this action not be in their self-interest, they have the capability of disobeying the contract at the cost of a sanction and being labeled as a ‘defector’ which may have social implications of a reduction in *reputation*, introduced in Sect. 6.

5.2 Sanctions

Sanctions, introduced in Sect. 5.1, deny players access to the common pool resource for a number of levels. Without access, players have no capability of increasing their *A* or *D* attributes, as no loot can be obtained. This only leaves agents with the capacity to replenish *HP* and *ST* by cowering, an action that could incur further sanctions.

The purpose of these sanctions is to limit ‘wasteful’ access to the common pool. Players that regularly appropriate from the common pool, however choose to cower, make little use of the items that they obtain. Intuitively, these items would be better given to players that more often comply with the fight contracts, as they will get immediate use out of it. This is especially important given the way that the enemy’s damage and health scale according to Eqs. 1 and 2, as the

longer an item is held, the less effective it will be at either dealing or mitigating damage.

The key dilemma with sanctioning is that this exclusion creates a negative feedback loop, as sanctioned players reduce the collective’s total potential damage output; it is better to have multiple attackers on a single turn and arm them with shields as well. Having multiple attackers increases the chance of defeating the enemy in a single turn, thus mitigating further damage, and supplying players with a shield ensures that defending the enemy’s attacks is easier. At the same time, these players must be trusted that they will use their items effectively, as covering will mean that they are effectively wasted.

We propose three different sanctioning mechanisms for affecting the duration that players are excluded from the common pool:

Fixed-Length Sanctions The simplest of methods is a fixed length sanction. In this method, any defectors serve a fixed-length sanction of $l \ll L$ levels on the interval $[0,.)$. We formalise this, and all subsequent sanctions, by introducing the term δl , which represents the change in duration of each successive sanction. Naturally, for fixed-length sanctions:

$$\delta l = 0 \tag{4}$$

Dynamic-Length Sanctions Dynamic sanctions build upon the fixed length method. Chairs are now given the choice of increasing or decreasing the sanction, by a maximum of a single level, depending on the defectors HP value. In theory, vulnerable agents would receive smaller sanction severity, thereby increasing the probability of a high average health in the community and increasing the expected utility of a weak player. This method aims to combat the self-defeating feedback loop by showing leniency to weak players.

We consider the HP of the collective, HP_c to be normally distributed, and subsequently calculate its mean and variance. This gives the distribution:

$$HP_c \sim \mathcal{N}(\mu_{HP}, \sigma_{HP}^2) \tag{5}$$

Which influences the change in sanction length, δl , according to Eq. 6

$$\delta l = \begin{cases} +1, & HP \geq \mu_{HP} + \sigma_{HP} \\ -1, & HP \leq \mu_{HP} - \sigma_{HP} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

and ensures that only players at least one standard deviation above or below the mean have varied sanctions. This is in keeping with trying to ‘counteract’ the negative feedback of sanctions by trying to equalise the accessibility of the common pool.

Graduated-Length Sanctions The final method of determining sanction length is inspired by Ostrom’s principles. Graduated sanctions require increasing the sanction length with each repeat defection, up to a maximum sanction level. This method aims to heavily punish repeat offenders, causing them to change to a more collective strategy, whilst mitigating detriment to the collective as a sanction of length 6, say, may better be served in installments of 1, 2 and 3 successive sanctions. We formalise this sanction in Eq. 7

$$\delta l = +1 \tag{7}$$

To tackle the scenario designed in this section, a multi-agent system is adopted, where independent agents act as players within the game.

6 Agent Implementation

In this section, we present the inspiration behind the agent strategy, as well as the primary mechanisms that govern the agent’s behaviour. The basis of the agent behaviour is encapsulated in its *reputation*, which is used as the metric by which opinions are formed, trust is built, and sanctions are applied Table 2.

Table 2. Parameters used in agent design

Parameter	Range
Reputation (R)	[0, 100]
Social capital (SC)	[0, 100]
Trusted social Network (TSN)	[Agent]

In this table are the four principal aspects of the agent design. *Reputation* considers the *needs* and *productivity* of an agent. The evolution of an agent’s health and stamina define its *needs*, where a low S and HP imply that the agent is participating in battle (by not cowering), and therefore results in a reputation increase. The agent’s decision history over a single level of battle phases is used to determine its *productivity*, where a high fight-to-cower ratio rewards an agent with a reputation increase. Finally, when calculating reputation, an agent considers the opinions of the social network.

Agents send a *gossip* message which consists of a set of (other) agents and their respective reputation scores. Using these *gossip* messages, an agent can perform a weighted average of the information received to update their reputation of the other players. In this scenario, the weights are determined by the communicating agent’s level of *trust*.

Social capital rises and falls with the rate of contact between two agents. Continued communication results in an increase in SC , which in turn increases the likelihood of communication in the next gossip session.

Agents can develop opinions about their counterparts using *Trust*, a metric calculated by summing each agent's *SC* and *R*, which is then used to determine the player's *Trusted Social Network (TSN)*. For each agent the (*TSN*) is defined as the set of agents with a *trust* score above a certain threshold.

Trust not only influences how agents weight opinions, but also influences the election of the *chair*. The theory that high *reputation* entails high productivity and therefore a level of expertise allows agents to vote for chairs that they believe will be highly effective.

6.1 Preferential Utilitarianism

At the heart of the agent design is a codification of *preference utilitarianism*. To implement this ideology, we use *trust* to split all agents within the environment into two categories: the first being the *TSN*, introduced in Sect. 6, and the second being the remaining agents. With this network defined, we can summarise this ideology with respect to the cooperative survival game as the choice of actions that maximise the interests of the self, and the *TSN*, whilst endeavouring to satisfice all other agents involved.

Therefore, this agent must be adaptable to the changing circumstances of the social network by creating a dynamically scaling network of agent-to-agent relationships, where actions, disobedience, and peer-to-peer communication influence reputation scores and determine the strength of these relationships.

We envision a direct parallel to Ostrom's perspective on the benefits of non-centralised governance, which in the context of multi-agent systems, may only function in the presence of ubiquitous common knowledge [23]. Therefore, an agent strategy that simultaneously acts to both improve the common knowledge and learn from it, creates a sense of positive feedback.

6.2 Leadership Strategy, Evaluation And Sanctions

When voting on which agent will occupy the position of *Chair*, introduced in Sect. 5.1, this agent considers a weighted sum of the applicants' *Reputation* and *Social Capital* values. This list of scores is then sorted in descending order to yield a preference order.

The No-Confidence vote mechanism considers social norm disobedience among the *TSN*. An agent measures the performance of the institution by measuring the number of defections in the *TSN*, should this proportion be above a given threshold, the agent votes to depose the chair.

For determining whether to sanction defecting agents, the chair normalises the reputation score of the defecting agent with respect to the reputation of all other agents. If this value is above a given threshold, the agent is sanctioned with the mechanism in use.

6.3 Reputation, Sanctions And Loot Allocation

We introduce the concept *Expected Utility* [16], which is the total amount of utility the agent can produce from an allocated item, to inform loot allocation

at the end of the level. As per Sect. 5.1, agents with a higher reputation are more likely to follow through with fighting, while those with a lower reputation are more likely to renege. Hence, a leader uses the probability $P(U)_i$, of an agent i using an item with a value V_j , to calculate the expected utility $E[V]_i$ of giving agent i the weapon as:

$$E[V]_i = P(U)_i * \sum_j V_j \quad (8)$$

To maximise the expected utility gained by an agent, it is optimal to give the higher valued items to the agents which are more likely to use it (and hence adhere with social contracts). Through using reputation as a naïve indicator of an agent’s likelihood of adherence, the leader sanctions non-compliant agents to maximise the utility of more reliable agents. The non-sanctioned agents are then sorted according to reputation and iteratively given their requested items to ensure the distribution of all loot, as any discarded item has zero effective value. This creates the summation term in Eq. 8, where multiple pieces of loot can be allocated to one agent if the number of looting agents is less than the number of items in the loot pool. In reference to Tarantino’s *True Romance*, “it’s better to have a gun and not need it, than to need a gun and not have it.

6.4 Social Contracts and Collective Actions

Each agent has the opportunity to submit a proposal containing the rules of a potential social contract. According to preferential utilitarianism, this proposal must be designed to maximise the utility of the agent and their *TSN*, whilst ensuring community survival. To achieve this, Eq. 9 shows how the threshold for a single agent attribute is calculated, according to the above principals.

$$HP_{Threshold} = HP + (0.2 * \delta_1) + (0.1 * \delta_2) \quad (9)$$

where δ_1 and δ_2 represent the difference between the average collective health and the agent’s health, and the average collective health and average *TSN* health respectively. In line with preferential utilitarianism, the weight of personal-to-group state divergence is weighted twice as strongly. This allows the agent to create a proposal resulting in, for example, an attack decision for themselves, whilst modifying the threshold slightly to ensure weak members of the *TSN* are allowed to cover.

7 Experimental Design and Results

With the overarching dilemma of encouraging participation when the only possible sanctioning mechanism is exclusion, we investigate the three categories of sanction introduced in Sect. 5.2 to establish which method is the most effective for optimising the survivability of the collective.

We assess the survivability by considering the average level reached by the agents, in a simulator comprising 60 levels ($L = 60$), with 30 agents of each type:

Selfless, *Collective*, and *Selfish*. This yields a total of 90 agents, where all agents are given starting *HP* and *ST* values of 1000 and 2000 respectively.

For each of the test simulations, a parameter sweep of sanction lengths is conducted to produce a line graph showing survivability against sanction length, with each data point averaged across 30 iterations. Each sanction mechanism is simulated with both persistent and non-persistent sanctions to examine the difference between seamless transitions of power and chairs who actively suppress the decisions of their predecessors. We also note that an additional resilience and potential damage multiplier is applied to *X* and *Y*, respectively, to ensure that the game is not trivially winnable across all sanction lengths.

7.1 Fixed Length Sanctions

From the results in Fig. 2a, we see a parabola that peaks at $l = 4$, for the persistent sanctions and $l = 2$ for the non-persistent sanctions. We reason that the trajectory of this figure follows the intuition of the sanctioning mechanism. A 0-length sanction is insufficient in restricting the common pool from free-riders, who will ‘waste’ the utility of weapons by choosing inaction, resulting in less damage than would otherwise be achieved by prioritising reliable agents and, across multiple turns, less survivability.

A similar result is found with longer duration sanctions of $l \geq 7$, where the over-exclusion of agents results in equally low survivability in the persistent case. With such long sanctions, it is impossible to effectively arm agents with the swords and shields needed to survive, so the net damage and defence ‘potential’ of the collective is reduced. Therefore, fewer agents are capable of effectively attacking and/or defending, so defeating the enemy becomes increasingly more challenging. This, again, across multiple turns, results in less survivability.

Intuitively, there is a maximum reached in between these two extremes, where a trade-off between the over-exclusion and under-restriction of the common-pool is achieved. In the persistent case, it is with a sanction duration of $l = 4$, which enables non-compliant agents to be prevented from ‘wasting’ the high-utility loot, while still enabling them to be sufficiently equipped to remain alive.

A disparity between the persistent and non-persistent sanctions can also be seen in Fig. 2a. We suggest that this is due to the frequency of *Chair* re-elections causing sanctions to effectively be ‘forgiven’. For example, a sanction of $l = 7$ may be interrupted after three turns due to a change in *Chair*, resulting in the agent effectively serving an $l = 3$ sanction. It is likely the case that the re-election period is shorter than the sanction duration. This results in the survivability achieved from longer duration, non-persistent sanctions being similar to the survivability of the lower duration, persistent case (a difference of at most eight levels). The rate of survivability decrease is also much slower for non-persistent sanctions.

7.2 Dynamic Length Sanctions

In Fig. 2b, where the x-axis denotes the initial sanction length to which Eq. 6 will be applied, a similar parabolic curve to the one described in Sect. 7.1 can be observed, where an increasing initial duration is followed, $l > 6$, by detrimental effects to the accomplishment of the common goal. Once again, the peak values at $l = 1$ and $l = 4$ dictate the optimal starting points according to this strategy. A noticeable difference can be found on $l = 0$, where there is a dramatic increase in the average level reached. We reason that this is due to the dynamics allowing an increase in sanction length to $l = 1$, resulting in the increase of equipment in the hands of high utility agents.

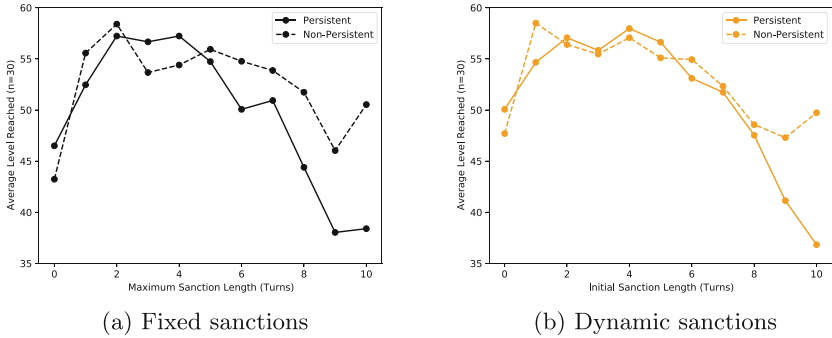


Fig. 2. Persistent (solid) and non-persistent (dotted), fixed-length (left) and dynamic (right) sanctions

It is worth noting that both persistent and non-persistent strategies seem to converge to the same results for $2 \leq l \leq 8$. However, the non-persistent sanctions show a global maximum at $l = 1$. A similar divergence between the two approaches can be seen for $l > 8$ as in Fig. 2a, mainly credited to the forgiving nature of sanctions in non-persistent transitions of power.

When directly comparing them to the results in Fig. 2a, we can notice that this adaptive approach to sanctioning, modified according to each agents state, shows an improvement on average performance for each sanctioning length. We theorise that this performance increase is due to the increased leniency given to vulnerable agents. Giving these weak agents the opportunity to allocate equipment increases their capabilities, thus increasing the total utility of the collective.

7.3 Graduated Length Sanctions

Following the improvement to the fixed-length sanctions made by the dynamic-length sanctions, we introduce a third and final mechanism of graduated-length sanctions, inspired by Ostrom.

Figure 3a deviates from Fig. 2a and b in its trajectory as the sanction length tends to $l = 10$. Here, the survivability trends upwards until a peak at $l = 5$, where it plateaus. This is unlike the previous experiments, where a longer sanction duration was detrimental to the collective.

We reason that this behaviour arises, as agents are never capable of reaching the upper sanction bound of $l \geq 5$, yet are able to effectively serve it in instalments. Reaching an upper bound of, say, $l = 5$ implies that an agent has been sanctioned for a total of ten turns prior to this, effectively serving an $l = 10$ sanction. However, these sanctions are not necessarily served consecutively. Therefore, agents are permitted to access the common-pool to increase their attack and defence, ensuring that the total utility of the collective increases. This removes the possibility of ‘useless’ agents, who are incapable of attacking or defending, as it is more likely that every agent has at least one piece of equipment to use.

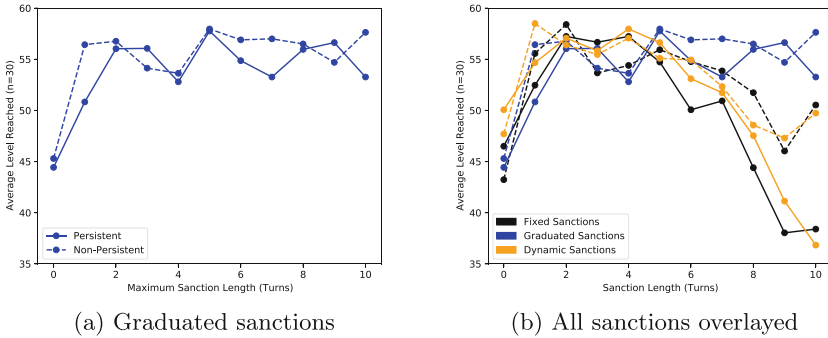


Fig. 3. Persistent (solid) and non-persistent (dotted), graduated (left) and all (right) sanctions

The lower bound of this plot at $l = 0$ trivially mirrors the behaviour in Fig. 2a, as a maximum length graduated sanction of $l = 0$ is identical to a fixed length sanction of the same duration, with the same issues with survivability.

It is also possible that agents are more incentivised to participate with this sanctioning system, as the plateauing behaviour implies that sanctions of high length are never reached. Graduating sanctions would allow agents to allocate equipment in between sanctions of increasing length, increasing their individual utility whilst also allowing time for an adjusted strategy to take hold.

Ultimately, we see this experiment as unifying the knowledge from experiments reported in Sects. 7.1 and 7.2. We have established that a sanction of duration $l = 4$ is effective, however detracting from the collective is detrimental to survivability. Lenient sanctions allow for less-compliant agents to appropriate from the common-pool leading to wasted utility, however it is still important to provide them with the basic means of survival in the event that they may fight in the future, evidenced by the steep decline in survivability in Fig. 2a and b. Therefore, we see graduated sanctions as a form of ‘trade-off’, where a lenient

sanction allows for less reliable agents to arm themselves from an early stage, yet be prevented from wasting utility as the game progresses as harsher sanctions are implemented.

7.4 Summary of Experiments

From experiments in Sects. 7.1 and 7.2, in Fig. 2a and b we get an inverted parabola for both fixed- and dynamic-length sanctioned. In both cases, an optimum value is reached at $l \approx 4$. Whilst the high-duration performance of both sanction types is similar, the low-duration ($l \leq 2$) is improved for the dynamic-length sanctions.

By introducing graduated-length sanctions in Sect. 7.3, Fig. 3a shows how the high-duration ($l \geq 5$) behaviour is improved, resulting in a plateau instead of a decreasing curve. The performance of $l \leq 2$ is unaffected, however, graduated-sanctions with a maximum length of $l \leq 2$ are effectively identical to fixed length sanctions of the same duration, due to the under-restriction issue.

Figure 2a and b show that the disparity between the survivability of persistent and non-persistent sanctions grows as the sanction length increases. This trend is eliminated with graduated-length sanctions, however, in Fig. 3a.

Ultimately, there appears to be three key ‘regions’ of sanction duration: under-restriction ($l \leq 3$), optimal ($l = 4$) and over-exclusion ($l \geq 5$), which can be seen from Fig. 3b. If the optimal sanction duration is chosen, all methods are equally as effective. If the sanctions are under-restrictive, however, then it is best to chose dynamic-length sanctions and if the sanction are over-exclusive then it is best to choose graduated-length sanctions.

8 Discussion and Further Work

8.1 Discussion

In economics, the Laffer curve has been proposed as showing a theoretical relationship between taxation and revenue [12]. It is argued that with 0% percent taxation, revenue is zero, whilst at 100% taxation, revenue is also zero, as there would be no incentive to work. Therefore, there must be some point in between for the level of taxation which maximises revenue.

By analogy, the same situation appears here: with zero sanctioning, there is no incentive to participate because free-riding is the risk-averse choice; but the ultimate sanction (permanent exclusion) is equally harmful to the collective, since by applying this sanction there will be no one left to participate. It is tempting to postulate that, as with the Laffer curve, there must be some fixed-length sanction duration which maximises the incentive to participate.

However, just as the Laffer curve does not warrant the assertion that cutting taxes increases revenue, starting from a fixed-length sanction and cutting it, as with dynamic-length sanctions, does not solve the problem either. It turns out that graduated sanctions perform best, and there are a variety of reasons for this:

including caution (in case of errors and possible appeals); the scope for agents to evaluate opportunity costs, and work out they would be better off participating; and the problem that for *one-shot* wicked problems like common-pool resource sustainability or high-stages cooperative survival, it is simply not possible to run multiple *in vivo* survival trials to find the optimal sanction duration for this particular problem.

We have also seen that, in this type of negative-feedback scenario, it may *not* be effective to have a seamless transition of power between elected chairs. Reneging on the sanctions introduced by one’s predecessor may be integral for achieving greater survivability of the collective. This is due to a ‘fresh’ chair giving offenders a clean-slate, reducing the length of the sanction. Should this period be sufficiently small, it reduces long sanctions to effective levels.

8.2 Further Work

Building on top of the notions of *trust* and *reputation* discussed, we could also explore the nature of posthumous reputation, where agents are conscious of their reputation after their death. This would allow an investigation into any heroic agents, who embody Ambassador Spock’s philosophy that “the needs of the many outweigh the needs to the few”.

As well as this, we have seen that different sanctions perform well at different lengths. Therefore, a ‘mixed-strategy’ sanction that combines multiple principles could improve survivability across all sanction lengths. The effect of introducing P2P trading, not restricted by the sanctioning process, could also be explored.

9 Summary and Conclusion

In this paper, we have specified an innovative, co-operative survival game where players are incentivised to participate to maximise collective survival, however the only possible punishment for non-compliance is exclusionary sanctions. This creates a ‘negative-feedback loop’. To solve this game, we have developed and specified a self-organising, multi-agent system that facilitates message passing, governance and social contract creation as self-organising mechanisms.

We have investigated three possible techniques for sanctioning non-compliant players: fixed-, dynamic- and graduated-length sanctions, which we assess using a series of survival trial experiments that investigate how the sanction duration for each of the different techniques impacts the survivability of the collective.

We have shown that fixed-length sanctions are feasible, so long as they are carefully tuned to prevent over-exclusion and under-restriction, as the performance is likened to a *Laffer Curve*. We then expand on this by introducing dynamic-length sanctions to offset the negative feedback by increasing and decreasing the sanction length based on the performance of an individual compared to the collective. These help solve the problem of under-restriction, yet still falter in solving the issue of over-exclusion, as the initial duration is too high.

Finally, we unified these two sanction types to implement Ostrom’s formulation of graduated-length sanctions by incrementing the fixed-length sanctions by one turn for each successive contract break. This solves the issue of over-exclusion and allows for effectively infinite-length sanctions to be put in place without harming the survivability of the collective. Therefore, we conclude that in a situation where sanctioning is both necessary yet harmful to a collective, implementing graduated-length sanctions is the optimal strategy.

Acknowledgements. We are particularly grateful to the members of the SOMAS team at Imperial College London, specifically, Neel Dugar and Rasvan Rusu for developing a robust infrastructure, as well as Sacha Hakim and Michal Makowka for their contribution to the agent specification.





References

1. Artikis, A., Sergot, M., Pitt, J.: Specifying norm-governed computational societies. *ACM Trans. Comput. Logic* **10**(1) (2009). <https://doi.org/10.1145/1459010.1459011>
2. Balke, T., De Vos, M., Padget, J.: I-abm: combining institutional frameworks and agent-based modelling for the design of enforcement policies. *Artif. Intell. Law* **21**(4), 371–398 (2013). <https://doi.org/10.1007/s10506-013-9143-1>
3. Brennan, G., Pettit, P.: *The Economy of Esteem: An Essay on Civil and Political Society*. Oxford University Press, Oxford (2004). <https://doi.org/10.1093/0199246483.001.0001>
4. Cialdini, R.: *Influence: The Psychology of Persuasion*. William Morrow e Company, New York, NY (1984)
5. Condorcet, N.d.: *Essay sur l’Application de l’Analyse á la Probabilité des Décisions Rendue à la Pluralité des Voix*. Paris (1785)
6. Davoust, A., Rovatsos, M.: Social contracts for non-cooperative games. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 43–49. AIES ’20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3375627.3375829>
7. Fehr, E., Rockenbach, B.: Detrimental effects of sanctions on human altruism. *Nature* **422**(6928), 137–140 (2003)
8. Gigerenzer, G.: How to make cognitive illusions disappear: beyond “heuristics and biases”. *Eur. Rev. Soc. Psychol.* **2**(1), 83–115 (1991). <https://doi.org/10.1080/14792779143000033>
9. Hardin, G.: The tragedy of the commons. *Science* **162**(3859), 1243–1248 (1968). <https://doi.org/10.1126/science.162.3859.1243> www.science.org/doi/abs/10.1126/science.162.3859.1243
10. Hare, R.: *Moral Thinking: Its Levels, Method, and Point*. Oxford University Press, UK (1981). https://books.google.co.uk/books/about/Moral_Thinking.html?id=SverDwAAQBAJ&redir_esc=y
11. Hegel, G.W.F.: *Phenomenology of Spirit*. Oxford University Press, Oxford (1807)
12. Hemming, R., Kay, J.A.: The latter curve. *Fiscal Studies* **1**(2), 83–90 (1980). www.jstor.org/stable/24434417
13. Kurka, D.B., Pitt, J.: Disobedience as a mechanism of change. In: *12th International Conference on Self-Adaptive and Self-Organizing Systems*. IEEE (2018)

14. List, C.: Social Choice Theory. In: Zalta, E.N., Nodelman, U. (eds.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edn. (2022)
15. Mill, J.: *Utilitarianism*. The works of John Stuart Mill, Parker, Son and Bourn (1863). <https://books.google.co.uk/books?id=lyUCAAAAQAAJ>
16. Mongin, P.: Expected utility theory. *Handbook of Economic Methodology*, pp. 342–350 (1997)
17. Nardin, L.G., Balke-Visser, T., Ajmeri, N., Kalia, A.K., Sichman, J.S., Singh, M.P.: Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *Knowl. Eng. Rev.* **31**(2), 142–166 (2016). <https://doi.org/10.1017/S0269888916000023>
18. Ober, J.: *Democracy and Knowledge: Innovation and Learning In Classical Athens*. Princeton University Press, Princeton, NJ (2008)
19. Ostrom, E.: *Governing the Commons: The Evolution of Institutions for Collective Action*. Canto Classics. Cambridge University Press (2015). <https://books.google.co.uk/books?id=hHGgCgAAQBAJ>
20. Ostrom, E.: Common-pool resources and institutions: toward a revised theory. *Handb. Agric. Econ.* **2**, 1315–1339 (2002)
21. Ostrom, E.: The challenge of common-pool resources. *Environ.: Sci. Policy Sustain. Dev.* **50**(4), 8–21 (2008)
22. Perreau de Pinninck, A., Sierra, C., Schorlemmer, M.: Distributed norm enforcement: Ostracism in open multi-agent systems. In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) *Computable Models of the Law*, pp. 275–290. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85569-9_18
23. Pitt, J.: *Self-Organising Multi-Agent Systems*. World Scientific, London, UK (2021)
24. Scott, M., Dubied, M., Pitt, J.: Social motives and social contracts in cooperative survival games. In: *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XV: International Workshop, COINE 2022, Virtual Event, May 9, 2022, Revised Selected Papers*, pp. 148–166. Springer (2022)



Governing Agents on the Web (Blue Sky Ideas)

Victor Charpenay¹ , Matteo Baldoni² , Andrei Ciortea³ ,
Stephen Cranefield⁴ , Julian Padget⁵  , and Munindar P. Singh⁶ 

¹ LIMOS, UMR 6158, CNRS, INP Clermont Auvergne, Mines Saint-Etienne,
University of Clermont Auvergne, Saint-Etienne, France

`victor.charpenay@emse.fr`

² University of Torino, Turin, Italy

`matteo.baldoni@unito.it`

³ School of Computer Science, University of St. Gallen, St. Gallen, Switzerland

`andrei.ciortea@unisg.ch`

⁴ University of Otago, Dunedin, New Zealand

`stephen.cranefield@otago.ac.nz`

⁵ University of Bath, Bath, UK

`j.a.padget@bath.ac.uk`

⁶ North Carolina State University, Raleigh, NC, USA

`mpsingh@ncsu.edu`

Abstract. This paper explores the interplay of multiagent systems (MAS) with the Web to identify research challenges related to the governance of Web-based MAS. We first consider as a case study the process for allocating donated bodily organs to potential transplant recipients. We then discuss candidate architectures for governance in Web-based MAS that would support this scenario, from which we derive several research questions that emerge in pursuing the aim of agents on the Web.

1 Introduction

There are significant similarities in the motivations behind multiagent systems (MAS) and the Web. Both disciplines and practices seek to advance decentralization and openness in that ideally there is not a single locus of control and participants can behave and interact broadly autonomously under local control. Their approaches are relatively complementary however, in that within a MAS decentralization is an objective to be managed by the designer and the developer, likely through explicit actions in the agent platform, whereas for the web decentralization is an architectural feature, realised by careful design of artefacts and protocols, but whose actual emergence is a function of usage and not necessarily foreseen by its designers, due to the looser coupling of components.

A vision for a web that supports socio-technical systems was set out by Berners-Lee in “Weaving the Web” [3] over two decades ago, at which time the concept of agent was well established, but only relatively rarely has there

been much interaction between the two communities. In parallel, the conception of MAS as a basis for socio-technical systems—whether through (social) simulation, digital twinning or mixed human-agent environments—has been a sub-theme in MAS, although not always using the term socio-technical system. The initiation (ca. 2016) of discussions about ethically-aligned design [47] and the EU High-Level Expert Group report [25] on Trustworthy AI has turned the governance question from an occasional process external to a (socio-technical) system to an on-going process that is intrinsic to the system design, where physical world concepts are reified into software and software entities are reflected into the physical world [45] to facilitate human oversight, through, for example an approach to engineering stakeholder values into systems, as described by Noriega et al. [32].

The above is the context in which this paper addresses the interplay of multi-agent systems with the Web. Specifically, it considers how constructs and techniques identified in the study of the governance of MAS be realized over the Web architecture and how the governance of the Web be beneficially structured based on constructs and techniques developed for the governance of MAS. In other words, it seeks to identify synergies in both these directions:

- What does the Web offer to support the governance of MAS?
We anticipate ways to use the interoperability and scalability of the Web to build easy-to-use, widely deployed MAS. Scalability, evolvability, and visibility are important non-functional requirements that are “guaranteed” if a system’s architecture relies on the Web.
- What does the governance of MAS offer the Web?
We anticipate approaches for governance that provide flexibility and local control with formal models that support correctness and generality going beyond the typically procedural kinds of governance seen on the Web today. A challenge here is to map abstract models for governance in MAS to Web components, in a way that preserves Web architectural constraints and thereby guarantees the associated non-functional requirements of scalability and decoupling.

The main contribution of this paper is an identification of some of the crucial challenges pertaining to the interplay of MAS and the Web, and the formulation of some initial research questions that might guide future research on this topic. To be more specific, our goal is to understand and explain how MAS, and how the governance of MAS, might be realized through outlining a minimum viable product: one that satisfies the requirements for MAS/MAS governance but aligns with the design idioms of web architecture. We approach this issue firstly, in Sect. 2, by summarizing the aspects of MAS and the Web that in the view of the authors pertain to governance in MAS and the features of the Web that might facilitate the provision of governance-as-a-service, through the manifestation of MAS technologies as Web components. Section 3 reproduces elements of a previously published scenario conceived as a regulated MAS, parts of which are then modelled using existing normative MAS languages, and subsequently provides some concrete elements for the mapping of MAS elements

to Web components in Sect. 4. Our aim here is to be illustrative of how MAS governance technologies could be applied to the problem and then how those technologies could be packaged and made available as services through the web architecture. While this section considers specific solutions to the representation of governance mechanisms, Sect. 5 addresses how to hide those specific choices behind abstractions so that agents can remain agnostic of the implementation while enjoying the function, through the presentation of normative concepts as web resources. Lastly, in Sect. 6, we draw the exploration of MAS, web architecture and hypermedia together to put forward some high level research questions to begin the investigation of their unification.

2 Key Concepts and Considerations

2.1 MAS Concepts

There are three terms, as discussed here, in relation to MAS that are key to the exploration of the question of MAS and the Web, namely norms, governance and institutions. We put forward some general-purpose intuitions about each of those to ground the discussion that follows.

The notion of norm was well-established in other disciplines (social science, law, logic) before its importation into MAS, as signalled by [40], amongst others. Much work in MAS tends to draw on [49], which established the logic of forbidden, permitted and obliged actions or states of affairs, which Von Wright developed further [50] for the legitimate expectations of behaviour.

Both North [33] and Ostrom [36] approach the matter of constraining behaviour from orthogonal economic perspectives, through the recognition of the notion of institution as a (consistent) set of norms that work together to support stakeholders goals.

Ostrom [36] however starts from the question of governance, in that she observes the process by which the norms are developed and refined into an institution, which is captured in the ADICO framework and the eight rules that she puts forward to characterise the circumstances of such social institutions. This is of course not the only model of governance and of institution creation, but it does provide great insight into normative structures that unite social and legal institutions.

Our goal here is to consider how these three concepts may be realised outside the controlled environment of a particular MAS platform and transplanted into the global computation ecosystem of the World Wide Web.

2.2 Web Concepts

We understand the Web as a combination of uniform resource identifiers (URI) supporting hyperlinked representations, along with a computational architecture that supports locating and accessing the identified resources. The computational architecture is based on standard protocols for manipulating resources (e.g., HTTP, CoAP). We think here of architectural constraints (such as for

caching, layering, and uniform interaction) as captured by the original design rationale behind the Web Architecture [21]. The W3C Recommendation for the Web Architecture [26] provides additional information on identification, interaction, and representation of resources on the Web. Linked Data principles [24] characterize the relevant constraints for large-scale data sharing, which can be supplemented by ontology specification [51] in general. Recent W3C Recommendations for the Social Web, including Linked Data Notifications, ActivityStream, ActivityPub, and WebSub [23], offer ways to develop social applications under Web architectural constraints.

Some extensions to the above computational and information architectures, such as through the observer pattern (implemented in CoAP [39]) and local state transfer [42], aim at moving from a Web of Documents [22] towards a Web of Things [29] and a Web of Agents, in which interactions are highly asynchronous and potentially lossy.

2.3 Relating MAS and the Web

There exist numerous studies that relate the Web with MAS, following two general trends. Some studies focus on applying RDF and Linked Data to expose agents to hypermedia-driven environments [6, 12, 18, 35]. Other studies combine a formal declarative model of norms [9, 14, 44] to specify social protocols [10]. Such social protocols provide a more thoroughly decentralized conception of Berners-Lee's [3] notion of social machines.

Dimensions that are common to all MAS architecture, such as the environment, organization and interaction dimensions, are defined at a higher level of abstraction than Web resources and protocols. Constraints such as caching, layering and uniform interaction apply to components, exposing a certain functionality through ports. Web components may only have client or server ports, exchanging messages in a standard protocol. Components with client ports only are called origin clients, those with server ports only are origin servers and a third kind of components, proxies, have an equal number of client ports and server ports, forwarding requests from clients to servers or vice-versa [21].

To be able to analyse the interplay between MAS and the Web, a mapping from MAS abstractions to (more concrete) Web components is necessary. Given the complexity of both fields, there is no trivial mapping and most likely not a unique mapping across the two levels of abstraction. In the following, we perform a case study to help identify mappings that would preserve effective governance mechanisms developed in MAS research.

3 Case Study: Organ Allocation

As a case study, we consider a simplified version of the process for allocating donated bodily organs to potential transplant recipients [48]. Due to the scarcity of organ donors, the limited period of organ viability after removal from the donor, and the desire to maximise the chances of a successful outcome, national bodies such as the United Network for Organ Sharing (UNOS),¹ in the United

¹ <https://unos.org/>.

States, or the Organización Nacional de Trasplantes (ONT),² in Spain, have been formed to manage organ waiting lists, match donors with recipients, and develop and monitor policies governing this process.

3.1 Carrel Revisited

Vázquez-Salceda et al. [48] present the design of Carrel, using an agent-mediated electronic institution (e-institution) [31] for the organ allocation process. This design specifies in a formal manner [19] the structure of the interactions between hospitals, tissue banks, and institutional agents managing the process, as well as the norms that govern these interactions and the match between a donor and recipient. The distribution of organs and tissues in Carrel, given its Spanish context, would be overseen by the Spanish ONT, together with the Catalan Organització CATalana de Trasplantament (OCATT).³

The Carrel platform is conceived to model the servicing of hospitals by connecting them with sources of tissues and organs, where the goal is to provide the *best* matched organ or tissue across all the sources registered with the platform. The interpretation of *best* is complicated and depends on a variety of factors that change over time, particularly in the case of organs. A strong requirement for a capacity for evolution motivated the development using an e-institution, made concrete through explicit computable norms. Because capacity for evolution is one of the main properties of the Web, Carrel offers an interesting case study through which to start motivating our research questions.

Tissue distribution is essentially demand-driven because tissues can be preserved and stored over extended periods with no significant degradation. Organ distribution is essentially supply-driven because the need is known before a suitable part becomes available. For the purposes of this paper, we only consider organ distribution where the need is known before availability.

Each hospital interfaces with Carrel through its Transplant Coordination Unit (TCU). A surgeon can request an organ or tissue via this unit which leads to the creation of an agent whose task it is to join the Carrel institution to obtain an organ or tissue that satisfies the surgeon's requirements. The requirements include the urgency of the request, hospital authentication information, organ/tissue data, recipient data, and a set of constraints on the organ or tissue.

To achieve its goal, the requesting agent must negotiate with other agents representing hospitals with potential donors.⁴ All agents are subject to behavioral norms that, if violated, are sanctioned. In the original version of Carrel, which was based on the ISLANDER framework [20], participating agents are in effect regimented by so-called governor agents, to prevent non-compliance. Here,

² <https://www.ont.es>.

³ <https://trasplantaments.gencat.cat>.

⁴ In its original form, Carrel includes a waiting list of donors, which other TCU agents can consult once a donor is available. Here, we only consider agent-to-agent interactions and assume that the agent holding the request asks other agents for potential donors.

we make the more general assumption that agents are regulated by norms and may choose to take non-compliant actions. Thus, we assume that agent actions in this contemporary Carrel are all visible to the regulatory bodies, as is the case in the physical world: hospital TCUs communicate their requests to members of ONT/OCATT, who then contact other hospitals to find a donor and select the best match.

In the course of the negotiation, hospitals may be tempted to behave unfairly towards others. They may want to provide incorrect information to ONT/OCATT to maximize their chances of finding a matching donor. The sanctioning power of ONT/OCATT is therefore essential. If the Carrel institution is to be cast as an (institutional) environment deployed on the Web, this sanctioning power should be retained by the environment. For the purposes of this paper, we consider a simple norm guaranteeing fair allocation of donors: *a requesting agent must not accept an organ or tissue if it was previously offered a better match* (with respect to criteria defined by the institution). The associated sanction is that the institution may reject any future request made by the violating agent.

3.2 Norms and Roles

Following one approach from the MAS literature [8], we can capture the sociotechnical system requirements in terms of accountability [1,2,11] and only then proceed to identify the information exchanges between the agents and from there their individual actions.

For concreteness, in the example below, we adopt the Custard notation [9]. The language is based on a conventional relational model. We can think of each predicate, such as **Registered** and **Certify**, as mapping to a relation and containing event instances. Additional relations are computed from the norm semantics: these include **violated(AcceptBestMatch)**, which refers to the event of the **AcceptBestMatch** norm being violated. In Listing 1, the concerned roles are institution and hospital. A computational entity representing the stakeholder playing the institution role is empowered to revoke the certification of an agent representing the stakeholder playing a hospital role. The power is instantiated when a hospital is registered and applies when it violates the **AcceptBestMatch** norm.

Listing 1. OCATT’s power to revoke certification in Custard. Here, “power” is the norm type, “DecertifyPower” is its schema name, the IDs refer to the parties concerned with the “by” indicating who has power over whom; “create” is the event with which the schema is instantiated, “detach” is the event under which it goes into effect, and “discharge” is the event that describes what the power brings about. As in SQL, the attribute (column) names are elided but the matches (joins) take place based on their values.

```
power DecertifyPower hospitalID by institutionID
create Registered
detach violated(AcceptBestMatch)
discharge not Certify
```

3.3 Allocation Protocol

We provide an overview of the (simplified) negotiation protocol followed by TCU agents. The agent holding a surgeon's request for transplantation sends its request to another agent, which answers with a donation offer. The requesting agent has then the choice of confirming or rejecting the offer, presumably depending on the strength of this match vis à vis any other offers it may have received from other prospective donor hospitals.

Listing 2 shows a simple protocol in the Blindly Simple Protocol Language (BSPL) language [41, 43]. Some message parameters are adorned *out*, meaning the sender can set them freely (constrained only by key integrity); the sending of a message generates a binding for each such parameter. Other parameters are adorned *in*, meaning the sender must know the binding prior to sending the message, which means it must have obtained the binding from a previous message it sent or from a message it received from another agent. In particular, the requesting agent must obtain a certificate to confirm or reject an offer. This certificate must be obtained by some system component that embodies ONT/O-CATT, which may or may not deliver it, depending on prior violations made by the agent. The protocol of Listing 2 assumes the certificate is known prior to this protocol being enacted, which fact is indicated by the *in* adornment on the parameter `recipientCertificate`.

Listing 2. A possible *Organ Donation* protocol.

```

Donation {
  roles RecipientH, DonorH
  parameters out organRequestID key, out offerID key, out recipientInfo, out
             donorInfo, in recipientCertificate
  private decision, transferInfo

  RecipientH → DonorH: request[out organRequestID, out recipientInfo]
  DonorH → RecipientH: response[in organRequestID, out offerID, out
                               donorInfo]

  RecipientH → DonorH: confirmation[in organRequestID, in offerID, in
                                   recipientCertificate, out transferInfo, out decision]
  RecipientH → DonorH: rejection[in organRequestID, in offerID, in
                                 recipientCertificate, out decision]
}

```

The norm of `AcceptBestMatch` would apply to how this protocol is enacted and the norm `DecertifyPower` builds on top of `AcceptBestMatch`. To give `DecertifyPower` teeth, the information architecture must be such that the requisite information is available to the concerned party (i.e., the institution).

3.4 Norm Representation and Monitoring

Over the last 30 years, researchers in MAS have proposed many approaches to reasoning about norms as well as computational approaches to monitoring a MAS for norm violations [4, 7, 15]. A crucial aspect of this work is to provide a formal representation of norms (see, e.g., [17] for an overview of approaches). Here we choose one possible representation of the `AcceptBestMatch` norm for illustrative purposes (Listing 3). Using the Expectation Event Calculus (EEC)

[16], we can express the norm as a conditional rule of expectation, i.e. one that introduces a constraint on the future, that should be monitored for fulfilment or violation, once a condition is satisfied.

Listing 3. `AcceptBestMatch` norm expressed in the Expectation Event Calculus (EEC).

```
initially (
  exp_rule (
    and ([request (OrganRequestID, RecipInfo),
          response (OrganRequestID, OfferID1, DonorInfo1),
          quality_of_match (DonorInfo1, RecipInfo, Q1)]),
    never (
      and ([response (OrganRequestID, OfferID2, DonorInfo2),
            quality_of_match (DonorInfo2, RecipInfo, Q2),
            Q2 < Q1,
            eventually (confirmation (OrganRequestID, OfferID21, -, -, -))])))
```

The first argument of this rule checks the record of messages, assumed to be recorded as event calculus ‘fluents’, for the existence of an organ request and offer that match with a quality $Q1$. If that condition is satisfied, an instantiation of the second argument is created within an expectation fluent, stating that it should never be the case that another offer with a lower match quality has been made and then (eventually) confirmed by the recipient.

The EEC engine will track this expectation over time as new information arrives and will create a violation event if such a confirmation occurs. Further reasoning with related norms may conclude that an associated sanction should be applied or that a compensating action should be performed. Here, the EEC engine is an example of an active component of a MAS governance system that should be accommodated within any architecture for the governance of agents on the Web.

4 Candidate Architectures

On the Web, “effective” power is held on the server side: servers may hide information, redirect requests or reject them, thereby reducing the action space of agents. It is natural to think of institutions as components with a server port, receiving requests from the client port of agents, such as described in [37]. Several candidate architectures offer various levels of control, however. We have identified four classes of components for institutions (see Table 1).

Autonomous agents differ from classical user agents in the Web architecture (i.e., browsers). In particular, an autonomous agent might require both client and server roles simultaneously and can have one-sided elementary interactions, where they are not awaiting a response. The requirements of autonomous agents are closer to the ones of servients in the W3C Web of Things Architecture [29] lingo. Servients are components with both client and server roles that can interact in a peer-to-peer manner. An agent could be implemented as such a component, or it could be a process that runs in a runtime environment provided by such a component. If the agent is visible to other agents, i.e. if its representation is dereferenceable, the representation can point, for instance, to a Linked Data Notification inbox that receives messages from other agents [5].

Table 1. Mappings of an institution to Web components and associated characteristics with respect to governance; checkmark (✓): the characteristic of the institution is guaranteed by constraints on the Web component, question mark (?): the characteristic of the institution depends on the MAS architecture, not on the Web component

Institutional component	Norm update	Statefulness	Sanctioning
Read-only server	✓		
Read-write server	✓	✓	
Proxy	✓	✓	✓
Servient	✓	?	?

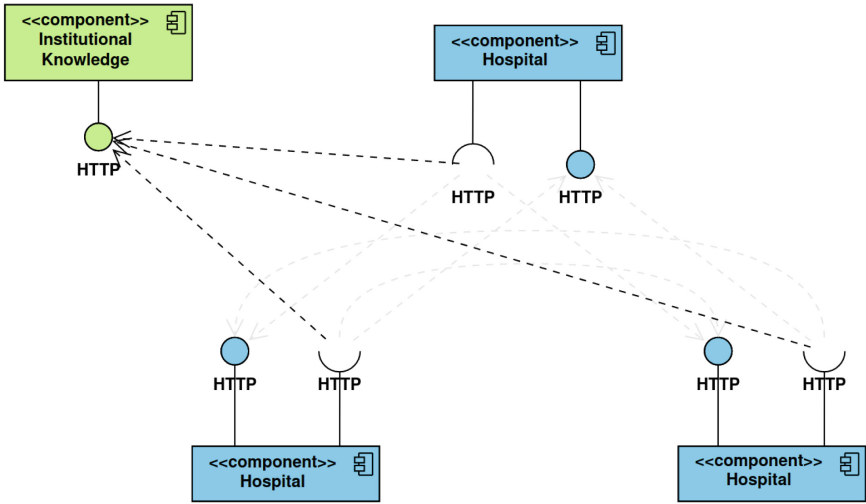


Fig. 1. Architecture for an institution governing Web agents: Institution as Read-Only Server

In a typical MAS, an institution may not materialize as a Web component at all. If norms are defined at design time, agents are guaranteed to behave (in general) as per these norms. Certification may also occur at design time, such that an agent either uses a single certificate throughout the system’s lifetime or periodically renews its certificate as long as its behavior specification does not change. In the above configuration, however, the institution has no sanctioning power at run time. It cannot easily redesign the normative framework in which agents interact either. Updating a norm would potentially require modifying the behavior of all agents at the same time.

4.1 Institution as Read-Only Server

Figure 1 illustrates one possible architecture. The behavior of an agent may easily be decoupled from the normative framework that regulates it, though:

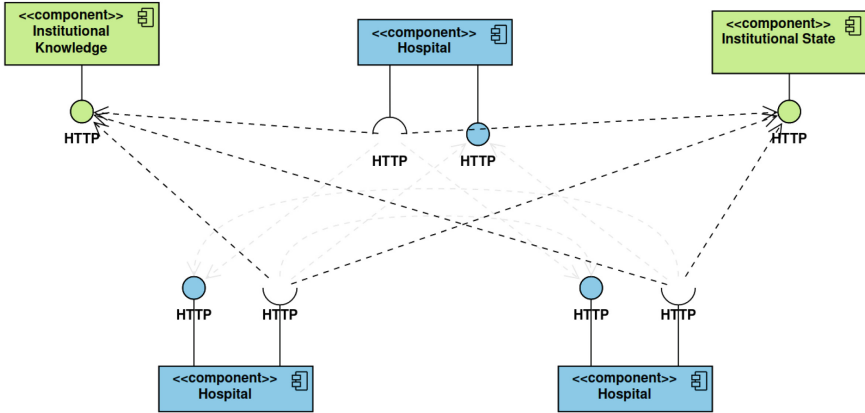


Fig. 2. Architecture for an institution governing Web agents: Institution as Read-Write Server

if norms are exposed by a read-only (origin) server, an agent may dereference the norms from time to time and internalize whatever formal specifications the server returns.

In this alternative configuration, the institutional component gains the power of dictating norms and changing them at run time. The ability of agents to get a certificate may depend on the fact they are aware of the latest version of the normative framework.

4.2 Institution as Read-Write Server

In order for the institution to gain sanctioning power, another component may manage its real-time state (Fig. 2).

The state of the institution includes the level of obedience of each participating agent, which is directly derived from the confirmations/rejections they generate. To be able to maintain its state, the new institutional component must be able to observe each agent-to-agent interaction. For instance, the certificate may be signed not for an agent but for a pair (agent, donation offer ID), forcing agents to request a new certificate every time they make a decision. If the certification server stores a history of confirmations/rejections, it effectively becomes an institutional component that is capable of deciding in real time whether agents violate norms and, if they do, sanction them by rejecting their certification request.

The institutional server would become a stateful read-write component, as agents, through their certification requests, change the state of the overall institution. Yet, it remains a purely reactive component, with a single server port.

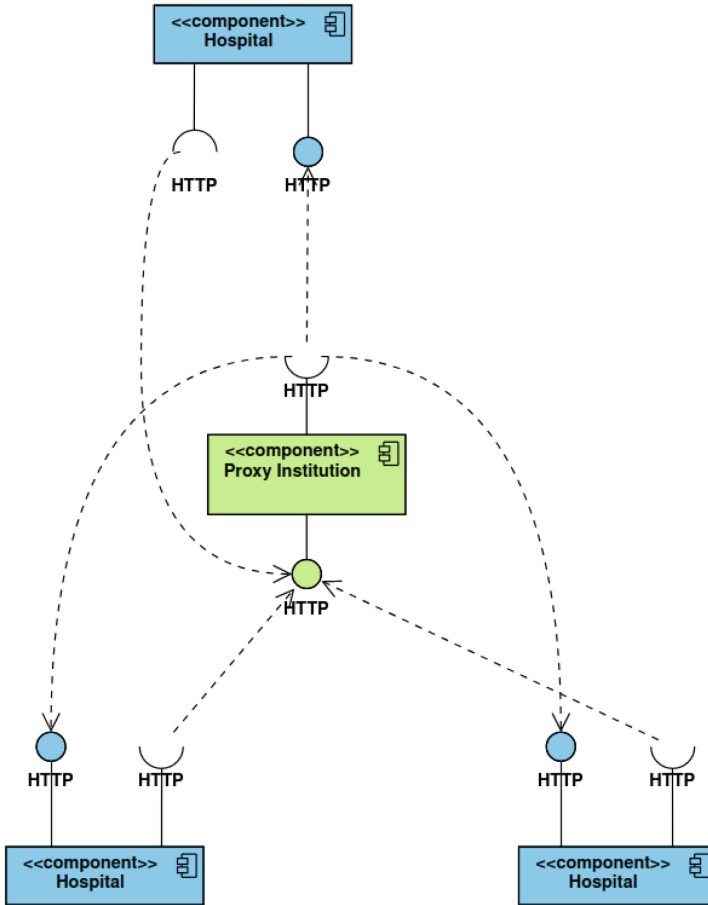


Fig. 3. Architecture for an institution governing Web agents: Institution as Proxy

4.3 Institution as Proxy

In the above configuration, the institutional component has no knowledge of how agents negotiate. If the institution is to be omniscient, another kind of component should be used. On the Web, it is common to use proxy servers to monitor activity (Fig. 3).

The main architectural constraint over proxies is that they have a client port and a server port, such that incoming requests (on the server port) are either immediately responded to or forwarded to another server, possibly after a rewriting step.

In our working example, the institutional server may be replaced by a proxy without modifying in any way the behavior of agents. Agents send requests to the proxy, which can keep track of negotiations and add a certificate on-the-fly if the requesting agent behaves properly. The proxy may also turn a confirma-

tion into a rejection, to sanction any misbehaving agent. The requesting agent receives feedback on the sanction through the other agent (which acknowledges the rejection, instead of the initial confirmation).

4.4 Institution as Servient

An alternative is to capture the institution as an explicit stakeholder supported by an agent on par with the other parties in the system. The institution becoming both reactive and proactive, must include independent client and server ports and becomes a *minima* a servient.

In ordinary operations, this agent may have little to say beyond conveying institutional norms and facts as in the previous approach. (In some real-life cases, the institutional agent is the same as a member of the system taking on that additional role.) However, by identifying this institutional entity, we make it subject to accountability. As a result, we can more perspicuously model the accountability relationships between the concerned parties than otherwise. A party can also question the institution, for example, if they fail to receive an organ in a timely fashion. This process may result in the institutional facts being disputed and adjudicated [46] and the norms potentially revised.

If the institution is embodied by an agent, its monitoring and sanctioning power doesn't depend on architectural constraints (at the level of Web components) but on the behavior of other agents (at the level of MAS abstractions).

5 Hypermedia-Driven Interaction

In the previous section, we discussed several alternatives for implementing institutions as Web components. To promote interoperability on the open Web, agents should be agnostic to such implementation details. This can be achieved by hiding the specifics of a configuration behind a (semantic) hypermedia layer.

To this end, hypermedia-driven interaction can support autonomous agents to interact with Web resources in a uniform way while being decoupled from the underlying components. Most prominently, this approach is used on the Web of Things to decouple clients from Web-enabled devices by hiding the interfaces used to access the devices behind abstract interaction possibilities and hypermedia controls. To illustrate how this works, an HTML page typically provides the user with a number of action possibilities, such as navigating to a different page by clicking a hyperlink or sending an order by filling out and submitting an HTML form. Performing any such action transitions the user to a new page and exposes a new set of possible actions. In each step, the user's browser retrieves not only an HTML representation of the current page from a server but also the hypermedia controls required to transition to new pages. Retrieving all this information through hypermedia allows websites to evolve without impacting the browser, and allows the browser to transition seamlessly across components. Hypermedia-driven interaction reduces coupling between components (e.g., browsers, proxies, origin servers) and allows them to be deployed

and to evolve independently from one another—a central feature that allowed the Web to scale up to the size of the Internet.

In the context of an institution on the open Web, the various action possibilities—such as retrieving formal specifications of norms, requesting a certificate, or sending messages to other agents—can be made available to the agents through hypermedia controls. Such hypermedia controls would encapsulate all the information required by an agent to interact with the component(s) that implement the institution, but to use the hypermedia controls in a reliable manner the agent would have to operate on an abstract model of the institution. For example, if an agent is required to obtain a certificate at run time to enact the *Organ Donation* protocol in Sect. 3.3, the agent could discover such an action possibility through hypermedia—but the agent would have to be aware of the notion of a certificate and should have the ability to request a certificate at run time. If an institution is realized as a read-only server, agents can only query the institution from time to time to check for norm updates or the need for a new certificate. The institution does not store the history of actions concerning certificates. If instead the institution is realized as a read-write server, such a history is kept and reactive behaviors can be specified to respond to norm violations. In these two cases, agents interact directly with one another. Alternatively, an institution can function as a proxy and interaction between agents is mediated by the proxy. Lastly, if the institution is realized as a servant, it can be proactive, and so resembles an agent. The degree of autonomy thus delegated can be controlled by the use of accountability frameworks.

In long-lived Web-based MAS, the institutional model could also evolve throughout the agents' lifetimes, for example, from using norms and certificates defined at design time to a model based on an evolving set of norms [30, 38] and certificates that have to be obtained at run time. Such an evolution would be reflected in the hypermedia environment through the set of action possibilities provided to agents at run time, and thus, to cope with this evolution, the agents would have to adapt by synthesizing a new course of action that meets their design objectives. While engineering such agents is still an open challenge, some related work investigates the design of agents able to plan and adapt to dynamic hypermedia environments (e.g., see [13, 28]).

6 Discussion: Research Questions and Challenges

The main contribution of this paper is in identifying some interesting research questions that can motivate research on the interface of MAS and Web architectures. Specifically, we propose the following research questions:

How should we model the presence of a governance layer in a MAS? In most common approaches, the governance layer is either implicit (agents are regimented) or embodied by norm-enforcement agents. In the Custard language for instance (Listing 1), institutions and agents are at the same level of abstraction. An alternative approach would be to model an *institutional* environment, as a

generalised form of agent environment (see [34,52]), such that reusable server or proxy components may be applied on several MAS architectures.

What aspects of a web-based deployment of a MAS may be subject to governance policies and included in an institutional environment? Examples would include constraints on the ownership and physical location of the component hosting an agent. A MAS that operates on a physical (or simulated) environment, e.g. on the Web of Things, must necessarily be regulated by stateful institutional components. Statefulness alone is however not enough, as institutional facts may be derived from brute facts generated in the physical environment. Only proxies and servients—components with a client port—could access the physical environment—exposed by environmental servers. If servers implement the observer pattern, as in CoAP, another class of institutional components may be added for “observer” servients. Observer servients would not be proactive (i.e. not agents) but still have the possibility to subscribe to environmental events.

How do we map the required properties of an institutional environment (monitoring, reasoning, sanctioning power) to constraints and mechanisms of a web-based deployment and more specifically to hypermedia control? Governance requires a certain point where governance decisions are made. Even in a decentralized architecture, that point reflects a form of centrality, albeit a weak one. Institutional components, such as the ones we have identified in the paper, have decreasing levels of centrality:

- proxy (to enforce policies), which supports real-time sanctions and requires no effort to continually re-engineer agents.
- institutional server (to maintain a shared institutional state), which supports violation detection and sanctioning but lets agents autonomously interact for the most part.
- federation of institutional servers, which relaxes the single institutional server case thereby reducing centrality but supporting potentially delayed sanctioning.

As suggested in Sect. 5, hypermedia controls, such as links and forms, should help agents follow high-level social protocols while still deviating locally, depending on the controls that servers expose. Hypermedia controls make it easy to decentralize interactions (a proxy may, e.g., redirect to a read-write server at run time) but hard to recentralize interactions (if agents interact in a peer-to-peer fashion, moving to proxy-mediated interactions would require re-engineering all agents).

We close this section with some more concrete questions, which could be considered implicit in the above:

How should one governance component be coupled with another? In the same way as the physical world is governed by many interacting governance frameworks, where something in one governance space takes on significance in another (for example) it make sense to modularize and reuse governance frameworks in the software world. An approach in principle is discussed in [14], but it is a long way

from there to the on-demand connectivity for a governance system that spans the web, but federations of federations (see the preceding question) might begin to address the question.

How should governance be governed? There are answers to this in the physical world through various approaches to collective decision-making and to regulation, but there is much to do to translate what works in the physical world into the software world: the functions may remain the same, but the mechanics must change to account for the properties of the software world. Regulation is easily interpreted as prescription, but such approaches are quite brittle, so in the same way as a robot is not programmed procedurally, so too must the reflection of regulation into software offer scope for agents to find solutions that fit the circumstances rather than being told what to do. This in turn could be applied to the creation of institutions on demand by specifying norms about norms [27] *ex ante*, supported by oversight (and revision [30]) in operation and by logging (for audit functions) *ex post*. This is all easily said, but far from easily achieved effectively and elegantly at Web scale.

7 Conclusion

Thinking about MAS and the Web together opens up new opportunities for building large-scale sociotechnical systems. Such systems would take advantage of the flexibility derived from MAS and the scalability and familiarity (to most developers) derived from the Web. We have intentionally focused here on approaches to governance that are within the authors' experience, since we understand the requirements of those approaches. Conversely, the authors with knowledge of the web, understand the capabilities and idiomatic usage of web components to guide the design stage so as to ensure the preservation of the Web architecture properties from which MAS could benefit. In crafting the connections between particular approaches to governance and particular configurations of Web components, as set out in Sects. 4 and 5, we believe we have established a "minimum viable product" on paper for MAS on the Web, but it will take many more iterations from the wider community to complete the transition for MAS from silo to the World Wide Web. The possibilities are however promising and we invite the research community to join us in investigating them.⁵

Acknowledgments. Singh thanks the US National Science Foundation (grant IIS-1908374) and the Department of Defense (Science of Security Lablet) for support for this research.

The authors acknowledge the Leibniz Center for Informatics for hosting *Dagstuhl Seminar 23081: Agents on the Web* held in February 2023. Dagstuhl provided the working environment for the "Governance" group whose discussions provide the basis for this paper. In addition, the authors acknowledge Mehdi Dastani and Wendy Hall for participation in working group discussions.

The authors acknowledge the helpful guidance from the anonymous reviewers highlighting issues for attention from the initial submission.

⁵ <https://www.dagstuhl.de/23081>.

References

1. Baldoni, M., Baroglio, C., Micalizio, R., Tedeschi, S.: Reimagining robust distributed systems through accountable MAS. *IEEE Internet Comput.* **25**(6), 7–14 (2021). <https://doi.org/10.1109/MIC.2021.3115450>
2. Baldoni, M., Baroglio, C., Micalizio, R., Tedeschi, S.: Accountability in multi-agent organizations: from conceptual design to agent programming. *J. Auton. Agents Multi-Agent Syst.* **37**(7) (2023). <https://doi.org/10.1007/s10458-022-09590-6>
3. Berners-Lee, T.: *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. Harper Business, New York (1999)
4. Boella, G., van der Torre, L.W.N., Verhagen, H.: Introduction to normative multi-agent systems. *Comput. & Math. Organ. Theory* **12**(2–3), 71–79 (2006). <https://doi.org/10.1007/s10588-006-9537-7>
5. Capadislì, S., Guy, A.: Linked data notifications (2017). www.w3.org/TR/ldn/
6. Charpenay, V., Käfer, T., Harth, A.: A unifying framework for agency in hypermedia environments. In: Alechina, N., Baldoni, M., Logan, B. (eds.) *Engineering Multi-Agent Systems*, pp. 42–61. Springer International Publishing (2022)
7. Chopra, A., van der Torre, L., Verhagen, H., Villata, S. (eds.): *Handbook of Normative Multiagent Systems*. College Publications (2018)
8. Chopra, A.K., Dalpiaz, F., Aydemir, F.B., Giorgini, P., Mylopoulos, J., Singh, M.P.: Protos: Foundations for engineering innovative sociotechnical systems. In: *Proceedings of the 22nd IEEE International Requirements Engineering Conference (RE)*, pp. 53–62. IEEE Computer Society, Karlskrona, Sweden (2014). <https://doi.org/10.1109/RE.2014.6912247>
9. Chopra, A.K., Singh, M.P.: Custard: computing norm states over information stores. In: *Proceedings of the 15th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pp. 1096–1105. IFAAMAS, Singapore (2016). <https://doi.org/10.5555/2936924.2937085>
10. Chopra, A.K., Singh, M.P.: From social machines to social protocols: software engineering foundations for sociotechnical systems. In: *Proceedings of the 25th International World Wide Web Conference*, pp. 903–914. ACM, Montréal (2016). <https://doi.org/10.1145/2872427.2883018>
11. Chopra, A.K., Singh, M.P.: Accountability as a foundation for requirements in sociotechnical systems. *IEEE Internet Comput. (IC)* **25**(6), 33–41 (2021). <https://doi.org/10.1109/MIC.2021.3106835>
12. Ciorrea, A., Boissier, O., Ricci, A.: Engineering world-wide multi-agent systems with hypermedia. In: Weyns, D., Mascardi, V., Ricci, A. (eds.) *Engineering Multi-Agent Systems*, pp. 285–301. Springer International Publishing, Cham (2019)
13. Ciorrea, A., Mayer, S., Michahelles, F.: Repurposing manufacturing lines on the fly with multi-agent systems for the web of things. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 813–822. AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)
14. Cliffe, O., De Vos, M., Padget, J.: Specifying and reasoning about multiple institutions. In: Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., Matson, E. (eds.) *Coordination, Organizations, Institutions, and Norms in Agent Systems II*, pp. 67–85. Springer, Berlin, Heidelberg (2007)
15. Conte, R., Falcone, R., Sartor, G.: Introduction: agents and norms: how to fill the gap? *Artif. Intell. Law* **7**(1), 1–15 (1999). <https://doi.org/10.1023/A:1008397328506>

16. Cranefield, S.: Agents and expectations. In: International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems. Springer (2013). https://doi.org/10.1007/978-3-319-07314-9_13
17. Cranefield, S., Winikoff, M., Vasconcelos, W.: Modelling and monitoring interdependent expectations. In: Cranefield, S., van Riemsdijk, M.B., Vázquez-Salceda, J., Noriega, P. (eds.) Coordination, Organizations, Institutions, and Norms in Agent System VII, pp. 149–166. Springer, Berlin, Heidelberg (2012)
18. Dikenelli, O., Alath, O., Erdur, R.C.: Where are all the semantic web agents: establishing links between agent and linked data web through environment abstraction. In: Weyns, D., Michel, F. (eds.) Agent Environments for Multi-Agent Systems IV, pp. 41–51. Springer International Publishing, Cham (2015)
19. Esteva, M., Padget, J., Sierra, C.: Formalizing a language for institutions and norms. In: Meyer, J.J., Tambe, M. (eds.) Intelligent Agents VIII. Lecture Notes in Artificial Intelligence, vol. 2333, pp. 348–366. Springer (2001). ISBN 3-540-43858-0
20. Esteva, M., de la Cruz, D., Sierra, C.: ISLANDER: an electronic institutions editor. In: The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, July 15–19, 2002, Bologna, Italy, Proceedings, pp. 1045–1052. ACM (2002). <https://doi.org/10.1145/545056.545069>
21. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. ACM Trans. Internet Technol. **2**(2), 115–150 (2002). <https://doi.org/10.1145/514183.514185>
22. Fielding, R.T., Taylor, R.N., Erenkrantz, J.R., Gorlick, M.M., Whitehead, J., Khare, R., Oreizy, P.: Reflections on the REST architectural style and “principled design of the modern web architecture” (impact paper award). In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, pp. 4–14. ESEC/FSE 2017, Association for Computing Machinery, New York, NY, USA (2017). <https://doi.org/10.1145/3106237.3121282>
23. Guy, A.: Social web protocols (2017). www.w3.org/TR/social-web-protocols/
24. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology, vol. 1, No. 1, pp. 1–136 (2011). <https://doi.org/10.2200/S00334ED1V01Y201102WBE001>
25. High-Level Expert Group on Artificial Intelligence (AI HLEG): Ethics Guidelines for Trustworthy AI (2019). www.digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai
26. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, volume one. Technical report, World Wide Web Consortium (2004). www.w3.org/TR/webarch/
27. King, T.C., De Vos, M., Dignum, V., Jonker, C.M., Li, T., Padget, J., van Riemsdijk, M.B.: Automated multi-level governance compliance checking. Autonomous Agents and Multi-Agent Systems, pp. 1–61 (2017). <https://doi.org/10.1007/s10458-017-9363-y>
28. Kovatsch, M., Hassan, Y.N., Mayer, S.: Practical semantics for the internet of things: physical states, device mashups, and open questions. In: 2015 5th International Conference on the Internet of Things (IOT), pp. 54–61 (2015). <https://doi.org/10.1109/IOT.2015.7356548>
29. Lagally, M., Matsukura, R., McCool, M., Toumura, K.: Web of Things (WoT) architecture 1.1. Technical report, World Wide Web Consortium (2023). www.w3.org/TR/wot-architecture/
30. Morris-Martin, A., Vos, M.D., Padget, J.A., Ray, O.: Agent-directed runtime norm synthesis. In: Agmon, N., An, B., Ricci, A., Yeoh, W. (eds.) Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems,





- AAMAS 2023, London, United Kingdom, 29 May–2 June 2023, pp. 2271–2279. ACM (2023). <https://doi.org/10.5555/3545946.3598905>
31. Noriega, P.: Agent-Mediated Auctions: The Fishmarket Metaphor. No. 8 in IIIA Monograph Series, Institut d'Investigació en Intel·ligència Artificial (IIIA), Bellaterra, Spain (1997). Ph.D. Thesis
 32. Noriega, P., Verhagen, H., Padget, J., d'Inverno, M.: Ethical online AI systems through conscientious design. *IEEE Internet Comput.* **25**(6), 58–64 (2021). <https://doi.org/10.1109/MIC.2021.3098324>
 33. North, D.: *Institutions, Institutional Change and Economic Performance*, 1st edn. Cambridge University Press, Cambridge (1990)
 34. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. *Auton. Agents Multi-Agent Syst.* **17**(3), 432–456 (2008). <https://doi.org/10.1007/s10458-008-9053-x>
 35. O'Neill, E., Lillis, D., O'Hare, G.M.P., Collier, R.W.: Delivering multi-agent microservices using CArtAgO. In: Baroglio, C., Hübner, J.F., Winikoff, M. (eds.) *Engineering Multi-Agent Systems*, pp. 1–20. Springer International Publishing, Cham (2020)
 36. Ostrom, E.: *Governing the Commons. The Evolutions of Institutions for Collective Action*. Cambridge University Press, Cambridge (1990)
 37. Padget, J., De Vos, M., Page, C.A.: Deontic sensors. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 475–481. International Joint Conferences on Artificial Intelligence Organization (2018). <https://doi.org/10.24963/ijcai.2018/66>
 38. Savarimuthu, B.T.R., Cranefield, S.: Norm creation, spreading and emergence: a survey of simulation models of norms in multi-agent systems. *Multiagent Grid Syst.* **7**(1), 21–54 (2011). <https://doi.org/10.3233/MGS-2011-0167>
 39. Shelby, Z., Hartke, K., Bormann, C.: *The Constrained Application Protocol (CoAP)*. Technical Report RFC 7252, Internet Engineering Task Force (IETF), Fremont, California (2014). Proposed standard. www.tools.ietf.org/html/rfc7252
 40. Shoham, Y., Tenenholz, M.: On the synthesis of useful social laws for artificial agent societies (preliminary report). In: Swartout, W.R. (ed.) *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 276–281. AAAI Press/The MIT Press (1992)
 41. Singh, M.P.: Information-driven interaction-oriented programming: BSPL, the Blindingly Simple Protocol Language. In: *AAMAS-11*, pp. 491–498. IFAAMAS, Taipei (2011). <https://doi.org/10.5555/2031678.2031687>
 42. Singh, M.P.: LoST: local state transfer—an architectural style for the distributed enactment of business protocols. In: *Proceedings of the 9th IEEE International Conference on Web Services (ICWS)*, pp. 57–64. IEEE Computer Society, Washington, DC (2011). <https://doi.org/10.1109/ICWS.2011.48>
 43. Singh, M.P.: Semantics and verification of information-based protocols. In: *AAMAS-12*, pp. 1149–1156. IFAAMAS, Valencia, Spain (2012). <https://doi.org/10.5555/2343776.2343861>
 44. Singh, M.P.: Norms as a basis for governing sociotechnical systems. *ACM Trans. Intell. Syst. Technol. (TIST)* **5**(1), 21:1–21:23 (2013). <https://doi.org/10.1145/2542182.2542203>
 45. Smith, B.C.: *Procedural reflection in programming languages*. Ph.D. thesis, Massachusetts Institute of Technology (1982). www.hdl.handle.net/1721.1/15961
 46. Telang, P.R., Kalia, A.K., Madden, J.F., Singh, M.P.: Combining practical and dialectical commitments for service engagements. In: *Proceedings of the 13th International Conference on Service-Oriented Computing (ICSOC)*, pp. 3–18. No. 9435

- in Lecture Notes in Computer Science, Springer (2015). https://doi.org/10.1007/978-3-662-48616-0_1
47. The IEEE Global Initiative on Ethics of Autonomous and Intelligent System: Ethically aligned design: a vision for prioritizing human well-being with autonomous and intelligent systems, first edition (2019). www.standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead1e.pdf
 48. Vázquez-Salceda, J., Cortés, U., Padget, J., López-Navidad, A., Caballero, F.: The organ allocation process: a natural extension of the Carrel agent-mediated electronic institution. *AI Commun.* **16**(3), 153–165 (2003)
 49. Von Wright, G.H.: *Norm and Action: A Logical Enquiry*. Humanities Press, New York, International Library of Philosophy and Scientific Method (1963)
 50. Von Wright, G.H.: Deontic logic: a personal view. *Ratio Juris: Int. J. Jurisprud. Philos. Law* **12**(1), 26–38 (1999). <https://doi.org/10.1111/1467-9337.00106>
 51. W3C: OWL 2 Web Ontology Language: Document overview (2012). www.w3.org/TR/owl2-overview/. w3C Recommendation. Accessed 03 March 2023
 52. Weyns, D., Omicini, A., Odell, J.J.: Environment as a first class abstraction in multiagent systems. *Auton. Agent. Multi-Agent Syst.* **14**, 5–30 (2007)

Studies on the Notion of Value



Addressing the Value Alignment Problem Through Online Institutions

Pablo Noriega¹(✉) , Harko Verhagen² , Julian Padget³ ,
and Mark d’Inverno⁴ 

¹ Artificial Intelligence Research Institute (IIIA-CSIC), 08193 Barcelona, Spain
pablo@iiia.csic.es

² Stockholm University, 114 19 Stockholm, Sweden

³ University of Bath, Bath BA2 7AY, UK

⁴ Goldsmiths, University of London, London SE14 6NW, UK

Abstract. As artificial intelligence systems permeate society, it becomes clear that aligning the behaviour of these systems with the values of those involved and affected by them is needed. The value alignment problem is widely recognised yet needs addressing in a principled way. This paper investigates how such a principled approach regarding online institutions—a class of multiagent systems—can provide key insights on how the value alignment problem can be addressed in general.

Keywords: Engineering values · Value alignment · Online institutions · WIT design pattern · Conscientious design

1 Motivation and Background

The objective of AI has been characterised as the design and construction of artificial autonomous entities. Arguably, such autonomy is the source of the most significant contributions of AI to society but also of its most significant concerns. One way to modulate artificial autonomy is to incorporate ethical considerations into the design and construction of artificial systems. In particular, to conceive a form of ethics as a means of controlling that autonomy. Stuart Russell articulated this intuition as the challenge *to build systems that are provably aligned with human values* which is now referred to as “the Value Alignment Problem” (VAP) [15].

The Value Alignment Problem can be understood as an engineering challenge that needs a rigorous approximation to the notion of “value” if one intends to evidence the degree to which an AIS provably aligns with a set of values. We propose to address the VAP challenge with a principled approach that starts by circumscribing our treatment of the VAP to a particular class of AIS: *Online Institutions* (OI), then establish relevant conceptual distinctions for this scoped version of the problem and define constructs that capture those distinctions.

With these elements—and the background of “conscientious design” [9, 11]—we can then propose heuristics and methodological guidelines for the design, operation, and monitoring of OIs that are provably aligned with some values. Although this is a restricted version of the VAP, we claim that value alignment for OIs involves the same requirements as the full VAPs. However, we argue that the very definition of OIs includes some particular features that allow a precise characterisation of value engineering.

This paper is an argument for this claim, organised into three parts. First, in order to set the terms of the argument, in Sect. 2 we present a broad motivation for online institutions and in Sect. 3 discuss their most relevant features in intuitive terms. Next, in Sect. 4 we make explicit some assumptions about values that can be predicated on online institutions. Finally, in Sect. 5, we enumerate specific heuristics that illustrate how these (now explicit) assumptions support the dual empirical problem of embedding values in a system and assessing that the resulting system is provably aligned. The final section elaborates on our assumptions and gives some context for future work.

This paper is another step towards our goal of designing a principled approach to the VAP. The key technical details and their contextualization that complement the argument we present here can be found in four previous publications. (i) In “A Manifesto for Conscientious Design” [9] we outlined a research programme for value-driven design of artificial intelligent system; (ii) “Anchoring Online Institutions” [8] contains a more systematic presentation of the contents of Sects. 2 and 3; (iii) In “Ethical online AI systems through conscientious design.” [11] we outlined our proposal for a principled approach to VAP and discuss in some detail the motivation, background and core elements of the proposal; (iv) Finally, in “Design Heuristics for Online Ethical Online Institutions” [10] we discussed the value operationalisation process and some heuristics for how to attack the process.

2 An Intuitive View of OIs

Online Institutions are inspired by a set of overtly practical artefacts: conventional institutions, where a collective activity—say a classical auction—is run according to some institutional rules. One can simply look into the principles of how such conventional institutions work and translate them online. As we discuss next, online institutions interpret that intuition in a way that is convenient for all sorts of applications. Several commercial systems fall into the class of OIs, for instance, *Uber* and *Amazon*, and in [10] we use an ideal online ticketing service as a typical OI (to illustrate the value engineering process).

The following is an informal characterisation of online institutions as a multiagent system, and its distinguishing features are discussed below in Sect. 3. A more rigorous characterisation is in provided [8].¹

¹ In OIs, like in any multiagent system, one can identify two primitive components: the active agents in the institution and the environment that capacitates and governs the

Contract 1. Online institutions is the class of *multiagent systems* that are:

- (i) *open*: there is an “inside” and an “outside” of the OI, and while participants may enter and leave the OI, a priori one knows not which agents are active inside the OI;
- (ii) *hybrid*: human and software agents²;
- (iii) *situated*: it is part of the actual world and functions within a particular socio-technical context;
- (iv) *online*: the OI is a technological entity, and agents interact with it and among themselves via the environment(s) in which they are situated;
- (v) *regulated*: all agent interactions are subject to some constraints that are declared and enforced by the OI³;
- (vi) *state-based*: the institutional state is unique and the same for every participating agent, and only enabled institutional actions and feasible institutional events can change it;
- (vii) satisfy the *dialogical* and the *observability* stances (see Constructs 3 and 4 respectively). ‡

Features (vi) and (vii) are included in this definition because the OI governs a collective interaction that evolves over time. Thus, we need to refer to an institutional state that changes but changes when and only when *institutionally recognised* events and actions take place (and this last part is supported by Features (iv), (v) and Constructs 3 and 4).

Contract 2. The institutional state at time t (s_t) is the set of facts that hold in the institution at that time.⁴ ‡.

interactions of those agents [4, 7]. In OIs, the environment itself includes a limited ontology—which includes a set of entities that are involved in the description of the facts that may at some point hold in the institution, as well as enabling actions and feasible events—that is common to all the active agents. Because we mean to capture the governance functions of conventional institutions, the environment also provides the devices that determine whether agents can enter the environment, as well as the devices that govern the activity of agents (communication, display of information, enforcement of institutional constraints).

² Humans don’t need to be involved in every OI; what is, in fact, assumed is that the decision-making of participating (non-institutional) agents is “opaque” or not accessible to the institution. The point of this property is to acknowledge the need to govern the behaviour of participating agents that may be heterogeneous, incompetent, malevolent, or belong to different principals.

³ This feature may be realised in different ways; one is to think of OIs as *normative multiagent systems* (see [3]); however, in a given OI, the particular representation of institutional constraints and their enforcement is reflected in the institutional model (Ψ of \mathcal{I}) see Sect. 3.

⁴ We can be more precise defining it as a point in the institutional space at time t . That is, $s_t \in \mathcal{S}_t = \times_{i=1}^n D_i$, where each D_i is a “domain”, there is an initial state \mathcal{S}_0 that changes only when an event or an action performed by a participating agent complies with the active institutional constraints (actions and events are partial functions on \mathcal{S}).

The *Dialogical Stance* supports the enforcement of Feature (v) above (by filtering all potential changes through the interface implicit in Feature (iv)). The *Observability Stance* allows us to detect that a change has taken place.

Contract 3. Dialogical Stance. All institutional interactions are *illocutory acts* that are mediated by the OI *interface*.[‡]

Contract 4. Observability stance. At any point in time, the institutional state of the world is a finite set of observable facts.[‡]

The OI concept has been evolving over the years within the MAS community where various frameworks for social coordination have been proposed (see for example [1]).⁵

3 An Abstract View of OIs: The WIT Model

In general terms, an OI establishes, enforces, and processes *capabilities and constraints* to govern the collective activity of a community of hybrid autonomous agents. To make this view concrete, we can use the *WIT model* represented in Fig. 1 to characterise an OI as the combination of three components:

- \mathcal{W} corresponds to the fragment of the real world that is *relevant* for the activity that is performed within the OI,
- \mathcal{I} is an abstract representation of \mathcal{W} that establishes the “rules of the game” and thus provides the specification of how the OI is meant to operate, and
- \mathcal{T} consists of the information technology that implements and supports the OI.

In coarse terms, \mathcal{W} is the working system that humans or their software counterparts interact with. Those interactions involve tangible objects and have effects on the perceivable physical reality. By construction, the OI determines in \mathcal{W} the entities that are involved in those facts that are recognised in the state of the world (Construct 2). Also, by construction, the OI will recognise that only certain actions and events can change it (Construct 1 (vi)) and since all recognisable actions are illocutionary actions mediated by the OI (Constructs 1 (iv) and 3), the OI has to enable those actions for participating agents through the interface in \mathcal{W} .

While \mathcal{W} capacitates interactions in the real world, \mathcal{I} establishes how those interactions acquire an institutional status. Intuitively, we say that an action is enabled if that action can be executed by an agent inside the OI and all the real-world conditions for its execution can be met and their effects can be acknowledged by the OI (we say that all the “physical constraints” can be met). However, to be deemed *institutional* (Construct 1 (v, vi)), an attempted action not only has to be enabled, it also needs to satisfy the artificial constraints

⁵ We have previously referred to OIs as socio-cognitive technical systems and as hybrid online social systems in previous publications (see [5, 8, 11, 21]).

that are the “rules of the game” that the OI imposes on participating agents. Correspondingly, \mathcal{I} contains two models: an abstract model (Φ) of how that part of the world is relevant for the OI functions (including the natural or physical constraints of the relevant part of the real world); and another model (Ψ) that contains the artificial (institutional) constraints that govern those interactions (see [7] for a discussion of a metamodel to represent these constraints and their enforcement). In contrast with the entities of the real world that are part of \mathcal{W} , in \mathcal{I} there are agent identifiers constants and variables that stand for real-world entities and facts, and functions that stand for events and enabled actions that happen in the real world.

Finally, \mathcal{T} includes data structures that model the state of the world, processes that correspond to the activity of real agents, the code that implements the constraints established in \mathcal{I} and the rest of the technological platform that together supports the operation of the institution (see a discussion in [7]). Figure 1a suggests how the three components are interrelated and how these interrelations reflect some conventional notions about institutions.

How the three components of the *WIT* model complement each other is established in Property 1 (cohesiveness) (see [8]). On one hand, the OI defines an “ontology”. It determines in \mathcal{W} what the relevant part of the world is and in particular establishes as part of that ontology what events and what actions of the real world are relevant in the OI. It also includes as part of that ontology all other entities of the real world that are needed for those events and actions to accomplish their intended institutional effects. In other words, we say that the OI provides *capabilities* to participating agents by recognising the real-world objects, events and actions that enable participating agents to act within the institution. But by excluding some real-world objects, events and actions from the *relevant* part of the world, the OI also establishes constraints on what actions can be attempted and what events can take place.

On the other hand, by definition, OIs are regulated multiagent systems (Construct 1, Feature (v)) that establish and—thanks to Feature (iv) in Construct 1 and the Dialogical Stance (Construct 3)—enforce the “artificial constraints” (beyond the physical constraints) that govern the empowered actions and are modelled on (Ψ).

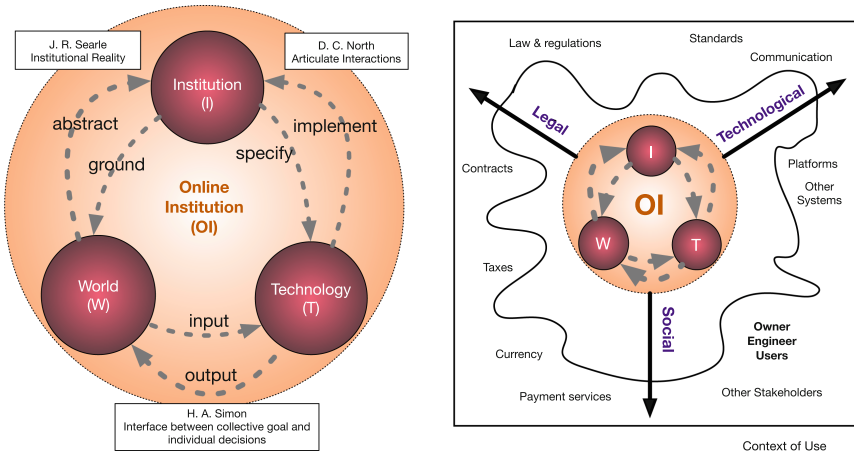
Our use of the term “institution” and our characterisation of OI purposely reflect four conventional interpretations of the term. (i) Searle’s distinction of an *institutional reality* that is different than the *crude* reality, (see [17]) is captured both in Features (i), (iii), and (vi) of Construct 1 and in the relationship between the \mathcal{W} and \mathcal{I} components of the *WIT* model in Fig. 1. (ii) North’s understanding of institutions as artificial constraints that determine the *rules of the game* [12] is the reason for Feature (v), and become a specification in \mathcal{I} that is implemented in \mathcal{T} , and (iii) Construct 1 captures Simon’s view of institutions as *interfaces* between individual decision-making and a collective objective [19] through Feature (iv) and the *Dialogical Stance* (Construct 3). This view is reflected in the relationships between the \mathcal{W} and \mathcal{T} views in Fig. 1. Finally, (iv) Ostrom’s ADICO framework and her outlook on the social insertion of

institutions [13] are addressed, the first one, in the expressiveness of the \mathcal{I} view of OIs (the way in which the artificial constraints are specified and enforced in Ψ) and, the second one, in the compatibility of a situated institution (Feature (iii) and the compatibility property discussed in the next section).

3.1 Properties of an OI

It is convenient to distinguish between the perspective of an OI as an entity on its own—a “stand-alone OI” (Fig. 1a)—as opposed to the perspective when considering an OI that is situated in its operating environment (Fig. 1b). In this section, we discuss two properties of the first perspective (Cohesiveness and Integrity) and one property of the second (Compatibility).

As noted above, the (stand-alone) OI is the combination of the three *WIT* components. It is convenient to look at them separately because they make explicit different features that need to be articulated in order to have a well-defined working OI. In fact, this decomposition becomes essential for the purpose of engineering values in an OI. In particular, the six arrows that connect the three components (Fig. 1) are key for separating design concerns and the contextualisation of values (see below and [10]). However, the three parts need to work in a cohesive way to ensure that an agent action can be properly accepted and executed following the “rules of the game” and thus correctly affect the relevant part of the world (see [8]).



(a) Relations between the *WIT* components and their relationship with three conventional views on institutions (Searle [17], North [12] and Simon [18])

(b) The Situated OI with its three compatibility requirements

Fig. 1. The “stand-alone” and “situated” views of an Online Institution (from [10]). Source for (a) Searle [17], North [12] and Simon [18]

Property 1. Cohesiveness An OI is *cohesive* if the three components are isomorphic with respect to actions and events.‡

Cohesiveness is based on the postulate that OIs are state-based and that only some actions and events can change the state of the institution (Features (vi) and (vii) in Construct 1). Technically speaking, the property assumes that (i) the (crude) agents, actions and events in \mathcal{W} correspond to agent identifiers, abstract actions, and events in \mathcal{I} , and to agent processes and inputs in \mathcal{T} ; and (ii) that there is a “state of the institution” that is defined by states that are specific to each view (\mathcal{W} , \mathcal{I} and \mathcal{T}). Thus, cohesiveness means that if at a given time the state of \mathcal{W} changes (because a crude event or action is deemed “institutional”), the state of \mathcal{I} and the state of \mathcal{T} change accordingly.

In spite of being situated in a particular context (Feature (iii)), a stand-alone OI is itself, an entity whose functioning and contents should not be contaminated, exploited, or altered by the external world.

Property 2. Integrity. An OI is *integral* if (i) only those agents that are admitted by the OI are provided an interface; (ii) the interfaces work correctly (i.e., only admissible institutional inputs enter the OI and only institutional outputs leave); (iii) institutional data is incorruptible (communication works, inputs are processed correctly, results of processes are persistent and outputs are properly sent); and (iv) the OI is impervious (only information that is requested, admitted or emitted by the OI enters or leaves the OI).‡

Finally, by definition (Feature (iii) in Construct 1), OIs are meant to support interactions that will have an effect in the real world, and actual individuals and organisations are involved in its operation. Therefore, in particular, to be effective they have to be *compatible* with the real world along three dimensions: those aspects of the actual world that (i) enable its online operation (technological standards, communication infrastructure, data, IP devices, ...); (ii) validate and make the transactions legally effective (contracts, applicable regulations and law), and (iii) are relevant for its successful social operation (economic conditions, social norms, commercial and working practices,...).

Property 3. Compatibility An OI needs to comply with *technological, legal, and socio-economic* standards, practices, and norms that enable its effective operation in the environment wherein it is situated.‡

3.2 Three Remarks on Conscientious Design

In the introduction, we proposed to understand VAP as a design problem. In the next two sections, we make reference to ideas that contribute to “conscientious design” (CD), as formulated in [9, 11]; here, we only touch upon three issues that support the design of OIs in which values are embedded.

Issue 1 At the core of CD is the understanding of design as a participatory process where the design stakeholders are involved in a cycle from the conception of the OI to its final decommissioning. This understanding assumes that values are taken into account in all the stages of the cycle, and that design stakeholders reach *consensus* at the different stages of the cycle (hence Assumption *CD.1* in Sect. 4.6). This understanding also leads to the realisation that no matter what the actual purpose or functionalities of the OI, and in addition to any other direct or indirect stakeholders of the OI, there are at least *three stakeholders* that are always involved in the *design, construction, and deployment* of the situated OI: the eventual users of the OI, the team of engineers, designers, and support people that are in charge of the construction, maintenance and operation, and monitoring of the OI and the owner, (the entity) who commissions, releases and operates and monitors the OI. Hence,

Property 4. Design Stakeholders Any OI always has at least three types of *design stakeholders*: owner, builder, and users.‡

Issue 2 The *WIT* Model serves as the blueprint for the design of OIs, in the sense of Alexander’s “design patterns” [2]. Its four salient elements have already been mentioned: the separation of concerns into the six arrows that link the *WIT* views: abstraction/grounding, specification/implementation, and input/output; the existence of three essential design stakeholder types (user, builder and owner); the two stand-alone OI properties: cohesiveness and integrity; and the three types of compatibility requirements of the situated OI (legal, technological and socio-economic).

Issue 3 there are three CD value categories: *thoroughness, mindfulness, and responsibility* that encompass other value categories proposed for embedding values in AIS (a comparison with the values proposed in EU [6] and IEEE [20] is detailed in [11]). In particular, for this paper, these three categories serve to validate the contextualisation of values (*Heuristic 2*) and legitimise the assessment procedures proposed for value alignment (*Heuristic 5*).

4 OI-Based Assumptions for Conscientious Design

As stated in Sect. 1, we are interested in a version of the *Value Alignment Problem* that applies to the design and building of OIs, not of AIS in general. The reason for choosing OIs to characterise a version of the VAP is that OIs justify some assumptions that in turn facilitate value engineering. The following is an attempt to make those assumptions explicit and to illustrate how these assumptions are put to work.

4.1 The Conventional Understanding of Values

We assume a rather standard motivational/cognitive view of values (compatible with e.g., Schwartz [16]) with the following properties:

- V. 1 Values motivate goals.
- V. 2 Values justify actions.
- V. 3 Values legitimise goals.
- V. 4 Values serve as criteria to determine preferences between states of the world.
- V. 5 Values are contextual.
- V. 6 In the assessment of a state of the world or in justifying an action, several values may simultaneously apply and these may be in conflict.‡

4.2 Assumptions for the Value Alignment Problem

There are three implicit assumptions in the wording of the Value Alignment Problem that clarify three issues: (i) that one can choose some values that the system should support, (ii) that those values can be embedded into the system, and (iii) that one can assess the alignment of the system with those values.

- Vap.1 The VAP can be decomposed into two problems: value embedding and the assessment of value alignment.
- Vap.2 One needs to be explicit about the values that will be embedded in a given OI and determine the alignment of the system with respect to all those values (see [14]).
- Vap.3 We understand that “provably aligned” is meant as an informal but objective (not necessarily proof-theoretic) way of determining that an AIS is aligned with a value or a set of values.‡

4.3 Assumptions for the Value Alignment Problem in OI

Because we are concerned with the VAP only with respect to OIs, we make explicit the way that the VAP is interpreted for the design of OIs with the following additional assumptions:

- VapOI. 1 In OIs, the VAP concerns the engineering of values in two different entities: in the governance of the multiagent system, and in the decision-making model of individual (artificial) *institutional agents*.
- VapOI. 2 We claim that the process of engineering values in an OI can be organised in a cycle with three main stages whose outcome is the specification (in \mathcal{I}) of how values will be implemented in the OI (in \mathcal{T}).
 - i *Contextualisation* in OIs: The choice of values depends on the domain of application of the OI, the needs and preferences of design stakeholders, and the separate design concerns and compatibility requirements of the OI (as induced by the WIT-design pattern). We assume that such contextualisation applies also to the embedding and assessment decisions.

- ii *embedding* can be split into two tasks that are closely linked with assessment: (i) *interpretation* (the features that make the value observable and its alignment objective) and (ii) *instrumentation* (the means that modulate the outcomes of actions accordingly). In OIs this is part of \mathcal{I} .
- iii *Assessment*. How to determine whether an OI is “provably” aligned with a value and with a set of values.

4.4 The Objective Stance

This fundamental assumption makes explicit how to interpret “provability” of alignment (*VAP.3*) in the case of OIs and motivates the working assumptions needed to eventually engineer specific values in OIs.

OS. 1 [Objective Stance:] The alignment of an OI with a value can be measured as a function of the state of the world.‡

In other words, values can be represented as a function of a finite set of observable facts.⁶

In order to make this *Objective Stance* fully operational, however, we still need to make explicit three additional *assessment assumptions* that materialise the measurement of the alignment of single values—identifying the degree of alignment of a value with a combination of the degree to which goals for that value are achieved—and also to deal with the alignment of multiple values simultaneously.

OSa.1 Goal satisfaction function: Given a goal for a value v , one can define a function that, for each state of the world, measures the degree of satisfaction of that goal (with respect to the value) in that state.

OSa.2 Value satisfaction function. Given a value and the set of all its goals, one can define a goal aggregation function that, for each state of the world, measures the degree of satisfaction of the value as a combination of the satisfaction of its goals, in that state.

OSa.3 Value alignment assessment: Based on the above one can define functions that capture different interpretations of alignment with respect to particular value interpretations. ‡

⁶ The rationale is as follows: *First*, by definition, OI are *state-based* and by the (*Observability Stance* (Construct 4)), the institutional state is a *finite set of observable facts*. *Second*, from *Val.4*), we assume that values can determine preferences over the state of the world, and therefore, one can define a preference relation on the set of institutional states P_v for any given value v . *Third*, Since the state of the world is finite, one can choose preferable states for a given value v and define them as *goals* G_v that are motivated for that value (*Val.1*) and also legitimised by it (*Val.3*)). *Fourth*, note that any goal (g) of value v_i will be included also in the preference relation (P_{v_j}) for every other value v_j (because g is one state of the world and because of *V.6*, several values may be involved in the assessment of a state of the world), however, it might not be a goal for v_j (g may or may not be in G_{v_j} .)

We label these assumptions “operational” because they need to be complemented with specific heuristics such as the ones we propose in Sect. 5. Such specific heuristics reflect different meta-ethical positions about values to some extent.⁷

Likewise, Assumption *OSa.3* makes operational alternative notions of “objectively aligned” because it allows different ways of understanding the combination of several values (from Assumption *V.6*).⁸

4.5 Assumptions About Instrumentation

Actions and events can be seen as functions that map the current institutional state into a new institutional state. Hence, since only institutionally acknowledged events and actions can change the state of the world, the way to embed values in the governance of an OI (in \mathcal{I}) is to enable, curtail, promote, or discourage individual actions or to modulate events in order to better achieve the intended goals. Analogously, institutional agents will be said to have value-aligned behaviour if and when their actions lead to the achievement of the intended goals. This alignment will depend either on predetermined behaviour that guarantees alignment with respect to specific goals by default, or because as institutional agents they are bound to comply with the institutional constraints and therefore the previous remark applies to their goal-driven reasoning.

Since institutional actions change the state of the institution, one can measure the effects (positive or negative) of an action α with respect to a goal g using the goal satisfaction function introduced in *OSa.1*. Note, though, that any given action can have measurable effects (positive or negative) towards the achievement of other goals and one can ascertain trade-offs in the effects of any particular action with respect to each one of the different goals and, ultimately, all values, using the satisfaction functions introduced in *OSa.1* and *OSa.2*. In other words:

- Ins.1* Let G be a goal whose observable facts is set F ; then, for each action α that affects a fact $f \in F$, one can measure the effect of α towards G by the change of the degree of satisfaction of goal G ; and likewise for any other goal G' whose observable facts include f .
- Ins.2* For each goal G one can choose instruments that either promote actions that have positive effects on G , or discourage actions that may have a detrimental effect.
- Ins.3* There are three types of value-embedding *instruments* for OIs: (i) *actions* that are recognised (in \mathcal{W}) by an institution for a given agent to have an

⁷ For example, the conjunction of Heuristics 2, 3 and 7 amounts to a weak form of consequentialism in which values are identified with goals but only for one specific OI and by the consensus of the design stakeholders who agree on the *consequences* of values.

⁸ The heuristics we propose in Sect. 5 (notably *Heuristic 5*) are meant to allow value alignments that reflect the individual perspectives of the different design stakeholders, the consensual perspective and a combination of the two.

institutional effect; (ii) *norms* (in $\Psi \in \mathbf{I}$) that regulate the conditions and effects of institutional actions; and (iii) *information* that may influence the decision-making process of participating agents.⁹ ‡

4.6 Assumptions from Conscientious Design

We make explicit three design assumptions that make the previous assumptions on values applicable in OIs. They are based on our remarks in Sect. 3.2. The *WIT* pattern provides assumptions for heuristics on value contextualisation and assessment features, on Conscientious Design Value categories, for heuristics to identify and tailor goals, and to define value alignment criteria. Other design assumptions about design—not CD-specific—are considered in [10].

- CD.1* Design stakeholders can reach *consensus* about OI values and goals, their satisfaction and aggregation, and about the impact of instruments and criteria for measuring alignment.
- CD.2* Values and their engineering should be *contextualised* for (i) the OI domain (i.e, the purpose of the OI, taking into account the \mathcal{W} ontology, enabled actions, and roles of participating agents); (ii) the three design stakeholders (user, owner, builder); (iii) the integrity and compatibility properties of the OI; and (iv) the six *WIT* separate design contexts (the six “arrows” of the *WIT* diagram: abstraction, grounding, specification, implementation, input, and output).
- CD.3* Conscientious design value categories (thoroughness, mindfulness, and responsibility) can be used to ascertain *completeness and correctness* of goals in the *WIT* contextualisation process and in the functions to *ascertain* the global alignment of the OI. ‡

5 Example Heuristics for *Value Engineering* OIs

The following remarks illustrate how the assumptions we made explicit above may be used to design value-aligned OIs.¹⁰ An OI is built with some general purpose in mind that needs to be properly contextualised and interpreted (*VapOI*. 2 (Sect. 4.3), *CD.1* and *CD.2* (Sect. 4.6)). Values inform the way this purpose is achieved: they clarify goals, assess and compare the outcomes of actions, and determine what governance instruments provide the best alignment (*OS*). In summary, values underlie the identification of what is relevant in the world and what “courses of action” lead to desired states of the world. More specifically:

Heuristic 1. An OI defines a *context for interaction* that capacitates actions and the constraints that modulate them. Values *enable courses of action* within that context. ‡

⁹ In fact, one may implement institutional agents whose behaviour operationalise those three types of instruments values. For example, institutional agents that perform discretionary norm-enforcement functions.

¹⁰ These heuristics complement the ones in [10].

Intuitively, the point of bringing values into the design process is to identify what actions should be available to the users of a system, how to evaluate the worthiness of different states of the world that are reachable through those actions and how to constrain or foster actions towards desirable states. In practice, this means that

- (i) Values serve to identify and adopt explicit goals. These goals need to be made precise enough (*OS*) so that they reflect the needs and motivations of each and all stakeholders and of the different design concerns (*CD.1* (Sect. 4.6)). Values consequently clarify and validate the ontology that needs to be incorporated into the OI (as part of the relevant fragment of reality (\mathcal{W}) and its abstract representation, Φ in \mathcal{I}).
- (ii) Goals are validated by values: each goal is a desirable state of the world for some value and the governance instruments lead actions toward that state (*Ins.3*). This happens for every goal of every value.
- (iii) The way that values are embedded in the OI—as capabilities and governance instruments that condition the evolution of the institutional state—validates the ontology and modulates the activity of participating agents towards desired end-states (*Ins.2*); that is, values refine the space of interaction and enable courses of action.
- (iv) The assessment of value alignment clarifies the preferable courses of action; because it measures the consensual satisfaction (of the consensual OI goals and values, for all contextualised values), the satisfaction for each stakeholder and the relative cost/benefit of alternative governance instruments.

Heuristic 2. Value contextualisation and embedding. OI values can be contextualised and embedded in four successive stages: (i) values for the application domain and CD categories for the consensual preferences of the three design stakeholders towards the OI, (ii) for the individual preferences of each of the design stakeholders of the OI; (ii) then for the compatibility requirements of the situated OI; and, finally, (iii) for the six WIT-articulation design concerns (abstraction, grounding, specification, implementation, input and output). ‡

Value interpretation (*VapOI* 2.i) is achieved by defining value-specific goals, and for each goal the features that are involved in the assessment of the contribution of that goal to that value; whereas *value instrumentation* is achieved by identifying the means to achieve those goals (*Ins.1,3*). In turn, *value assessment* (*VapOI*: 2.iii) is achieved by adopting goal measurement and aggregation functions (*OSa.* 1-3); as well as a way of assessing the impact (positive and negative effects) of the instruments with respect to all the goals (*Ins.* 2).

While establishing “courses of action” requires consensus among all stakeholders, different design stakeholders’ preferences should still be considered in the final assessment of value alignment. We articulate these remarks with *OSa.* 1-3 in mind:

Heuristic 3. OI’s values, goals, goal satisfaction functions, goal aggregation functions, value alignment functions and value instrumentation are *consensual*. ‡

Heuristic 4. Each stakeholder holds its own values, goal satisfaction, goal aggregation functions, and value alignment assessment functions. These stakeholder-specific functions apply to the assessment of the consensual OI's goals, and therefore do not necessarily coincide with the consensual assessments. Likewise, these stakeholder-specific functions are used for identifying and measuring the effects of the embedded values and will therefore provide each stakeholder with the elements for its own assessments of value alignment. ‡

Recall that the aim of our proposal is to embed values in an OI in such a way that the OI is *provably aligned* with them (*Vap.3*). Based on the previous two heuristics, we propose to address value alignment through a combination of three types of alignment that keep the consensual and individual differences in mind:

Heuristic 5. Value alignment can be assessed as a combination of three assessment procedures:

- 1 An assessment of the *effectiveness* of the governance instruments to satisfy the OI goals and the resulting aggregated value satisfaction based on consensual features (values, goal satisfaction, and goal aggregation functions (*Heuristic 3*)).
- 2 Assessing how *adequate* are the governance instruments for producing the alignment in terms of their direct and indirect effects (equally effective sets of instruments may have different cost-benefit trade-offs)
- 3 Assessing how *acceptable* the governance instruments are for the stakeholders. Acceptability combines the individual assessments of all the stakeholders. This individual assessment is the stakeholder's assessment of the effectiveness and adequacy of the instruments with respect to their own values (*Heuristic 4*), not the (consensual) OIs values. ‡

With the previous heuristic in mind, we now list heuristics that apply to the consensual aspects of the OI design: OI goals, governance instruments, goal satisfaction functions, and goal aggregation functions.

Heuristic 6. Choice of values and their goals can be addressed as a goal decomposition process (which is accompanied by a means-ends analysis). The resulting tree (for each value) is rooted in an abstract “tellic” value and its leaves are consensual goals. ‡

Goals determine the facts that need to be observable and there should be a consensus on how to assess, for any state of the world the extent to which that goal is satisfied (*OSa.1*). There should also be a consensus on how the combined satisfaction of those goals amounts to a satisfaction of the value that motivates them (*OSa.2*). From the Objective Stance, (Sect. 4.4) we propose a pragmatic compromise for goal satisfaction and goal aggregation: (i) an *objective function* that defines an ordering of the states of the world with respect to how good that state is for the satisfaction of the goal, and (ii) a threshold —*aspiration level*—for each objective function that determines the minimal level of satisfaction for that

goal. This way we can limit contradictions and tensions between the goals of the stakeholders and thus obtain a goal aggregation function.

Heuristic 7. Goals determine an *objective* function that gives the degree of satisfaction of the goal for each state of the world (with respect to a value). For each goal, there is an aspiration level that determines the minimal value of a state that achieves the satisfaction of the value. ‡

One can think of these objective functions for goals as a way of imposing a total order on the states of the world with respect to each goal, as a primitive sort of utility function of that goal, with positive and negative utilities separated by the aspiration level. Value satisfaction is determined by a composition of the goal satisfaction functions and amounts to an aggregated utility function of the combined satisfaction of its goals, with the value aspiration level as its threshold. Notice that, as a side-effect, *Heuristic 7* suggest how goal aggregation functions induce an ordering of goals.

Heuristic 8. Values are embedded in the OI as instruments that modulate what is actionable, in order to affect the parameters of an OI goal. ‡

Heuristic 8 implements the instrumentation assumptions but makes reference to goal parameters in order to identify the direct and indirect impact of an instrument. This allows the identification of trade-offs of the different instruments in order to address the *Adequacy* and *Acceptability* assessments in *Heuristic 5*.

In practice, for each (consensual) goal, the process—based on *Ins.3* (Sect. 4.5)—is first to identify those actions that affect the observable parameters involved in the assessment of the goal and explore for each action its (direct) effects on that goal and (the indirect effects) in other goals, based on *Ins.1*. Second, to instrument the action (*Inst.2*) to achieve the best effects; that is, (i) to enable the action (add it as a new action in \mathcal{W}), (ii) to inhibit the action (or eliminate it form \mathcal{W}), (iii) to regulate the action (foster, discourage, curtail or prohibit), or (iv) design information to incline participating agents decisions towards those effects.

However, *Heuristic 8* alone would produce too many instruments. One way to navigate this problem is to execute the instrumentation incrementally, by looking into the cost-benefit trade-offs of the instruments that may be more relevant for an effective alignment of the OI goals. To achieve this, one can use the goal aggregation functions to prioritise goals to identify the actions that impact the most important goals, and instrument only the most adequate (i.e. the ones with the best cost-benefit trade-offs).

Heuristic 9. Prioritise values and their goals, and instrument first those actions that affect most the more significant goals. Measure and compare the effects of instruments on the prioritised goals. ‡

6 Closing Remarks

Earlier publications on online institutions (OIs) provide substance and scope to the assumptions and heuristics we present here. In particular, in [10] we describe an online ticketing system to motivate and illustrate heuristics for value engineering, and in [8] we use *Uber* to motivate and exemplify the definitions and properties of the WIT pattern. Furthermore, a fuller description of Conscientious Design value categories (which are mentioned only tangentially here), and their relationship with other value taxonomies can be found elsewhere [11].

Here, our focus has been centred on the governance provided by any OI and has only mentioned ethical decision-making in passing. As we mentioned in earlier work [10], one can engineer values in an artificial agent in three ways: reactive behaviour, learned behaviour, and/or symbolic and explicit value-driven reasoning. This is particularly relevant when designing autonomous *institutional agents* who become active in an OI on behalf of the institution itself (such as performing some norm-enforcement functions, for example). A full discussion of this aspect is beyond the scope of this paper but the heuristics we propose in Sect. 5 also apply in principle to the engineering of values in autonomous agents.

Our understanding of the VAP—as expressed in the assumptions in Sects. 4.2 and 4.3—makes it a “design problem”. We actually propose a methodological approach to the design and construction of particular systems that would be provably aligned with a set of values, not a general solution of the VAP. Because we see the VAP as a design problem—and because one may wish to account for issues related to bounded rationality—our Objective Stance (Sect. 4.4) does not commit to any specific form of assessing value alignment or value aggregation. That choice would result from a consensus of the stakeholders who are involved in the design of a particular system.

The heuristics we propose for value engineering may also apply to other artificial autonomous intelligent systems but we have yet to explore this. Nevertheless, the class of OIs is interesting in and of itself for its intrinsic complexities but also because it encompasses an increasingly large class of existing AI-enabled systems.

In closing, we note that the Value Alignment Problem is only one instance of the relevance of values for AI in general. Our proposal, albeit centred on OIs, contributes to a broader project on an AI-oriented theory of relating human values to artificial system behaviour. We look forward to further investigating these possibilities.

Acknowledgements. Research for his paper is supported by the EU Project VALAWAI 101070930 (funded by HORIZON-EIC-2021-PATHFINDER-CHALLENGES-01), project VAE (grant TED2021-131295B-C31 funded by MCIN/AEI /10.13039/501100011033 and by the European Union’s NextGenerationEU/PRTR), and CSIC’s project DESAFIA2030 (BILT2005 funded by the Bilateral Collaboration Initiative i-LINK-TEC).




References

1. Aldewereld, H., Boissier, O., Dignum, V., Noriega, P., Padget, J.: Introduction, pp. 3–9. Springer (2016). https://doi.org/10.1007/978-3-319-33570-4_1
2. Alexander, C.: *A Pattern Language: Towns, Buildings, Construction*. OUP, Oxford (1977)
3. Andrighetto, G., Governatori, G., Noriega, P., van der Torre, L.W.N. (eds.): *Normative Multi-Agent Systems*, vol. 4. Dagstuhl Publishing, Saarbrücken (2013)
4. Argente, E., Boissier, O., Carrascosa, C., Fornara, N., Mcburney, P., Noriega, P., Ricci, A., Sabater-Mir, J., Schumacher, M.I., Tampitsikas, C., Taveter, K., Vizzari, G., Vouros, G.A.: The role of the environment in agreement technologies. *Artif. Intell. Rev.* **39**, 21–38 (2013)
5. Christiaanse, R., Ghose, A.K., Noriega, P., Singh, M.P.: Characterizing artificial socio-cognitive technical systems. In: Herzig, A., Lorini, E. (eds.) *Proceedings of the European Conference on Social Intelligence (ECSI-2014)*, pp. 336–446. CeUR (2014). www.ceur-ws.org/Vol-1283/
6. High-Level Expert Group on AI (AI HLEG): *Ethics guidelines for trustworthy AI* (2019). www.ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai
7. Noriega, P., Padget, J., Verhagen, H., d’Inverno, M.: Towards a framework for socio-cognitive technical systems. In: Ghose, A., Oren, N., Telang, P., Thangarajah, J. (eds.) *Coordination, Organizations, Institutions, and Norms in Agent Systems X*, *Lecture Notes in Computer Science*, vol. 9372, pp. 164–181. Springer International Publishing (2015). https://doi.org/10.1007/978-3-319-25420-3_11
8. Noriega, P., Padget, J., Verhagen, H., d’Inverno, M.: Anchoring online institutions. In: Casanovas, P., Moreso, J.J. (eds.) *Anchoring Institutions. Democracy and Regulations in a Global and Semi-automated World*. Springer ((in press))
9. Noriega, P., Verhagen, H., d’Inverno, M., Padget, J.A.: A Manifesto for Conscientious Design of Hybrid Online Social Systems. In: Cranefield, S., Mahmoud, S., Padget, J.A., Rocha, A.P. (eds.) *COIN@AAMAS*, Singapore, May 2016, *COIN@ECAI*, The Hague, The Netherlands, August 2016, *Revised Selected Papers. LNCS*, vol. 10315, pp. 60–78. Springer (2016)
10. Noriega, P., Verhagen, H., Padget, J., d’Inverno, M.: Design Heuristics for Ethical Online Institutions. In: Ajmeri, N., Morris Martin, A., Savarimuthu, B.T.R. (eds.) *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XV*, pp. 213–230. Springer International Publishing, Cham (2022)
11. Noriega, P., Verhagen, H., Padget, J., d’Inverno, M.: Ethical online AI systems through conscientious design. *IEEE Internet Comput.* **25**(6), 58–64 (2021)
12. North, D.: *Institutions. Institutional Change and Economic Performance*. CUP, Cambridge (1991)
13. Ostrom, E.: *Governing the Commons. The Evolutions of Institutions for Collective Action*. Cambridge University Press, Cambridge (1990)
14. van de Poel, I.: Embedding values in artificial intelligence (AI) systems. *Mind. Mach.* **30**(3), 385–409 (2020)
15. Russell, S.: Of Myths and Moonshine. A conversation with Jaron Lanier, 14–11-14. *The Edge* (2014). www.edge.org/conversation/the-myth-of-ai#26015. Accessed 12 Dec 2022
16. Schwartz, S.H.: An overview of the Schwartz theory of basic values. *Psychol. Cult.* (Online readings) **2**(1), 11 (2012)

17. Searle, J.R.: *The Construction of Social Reality*. The Penguin Press, Allen Lane (1995)
18. Simon, H.A.: *The Sciences of the Artificial*, 3rd edn. MIT Press, Cambridge (1996)
19. Simon, H.A.: Fact and value in decision-making. In: *Administrative Behavior: A Study of Decision-making Processes in Administrative Organization*, 4th edn. The Free Press (1997)
20. The IEEE Global Initiative on Ethics of Autonomous and Intelligent System: Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems, first edition (2019), www.standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead1e.pdf
21. Verhagen, H., Noriega, P., d’Inverno, M.: Towards a design framework for controlled hybrid social games. In: *Social Coordination: Principles, Artefacts and Theories*, SOCIAL.PATH 2013 - AISB Convention 2013, pp. 83–87 (2013)



Adding Preferences and Moral Values in an Agent-Based Simulation Framework for High-Performance Computing

David Marin Gutierrez³, Javier Vázquez-Salceda¹(✉) ,
Sergio Alvarez-Napagao^{1,2} , and Dmitry Gnatyshak² 

¹ Universitat Politècnica de Catalunya-BarcelonaTECH, Edifici Omega, C/ Jordi Girona 1–3, E-08034 Barcelona, Spain
{jvazquez,salvarez}@cs.upc.edu

² Barcelona Supercomputer Center (BSC), C/ Jordi Girona 1–3, E-08034 Barcelona, Spain
dmitry.gnatyshak@bsc.es

³ PwC Spain, Av. Diagonal, 640, E-08017 Barcelona, Spain
maringutierrezdavid@gmail.com

Abstract. Agent-Based Simulation is a suitable approach used nowadays to simulate and analyze complex societal environments and scenarios. Current Agent-Based Simulation frameworks either scale quite well in computation but implement very simple reasoning mechanisms, or employ complex reasoning systems at the expense of scalability. In this paper we present our work to extend an agent-based HPC platform, enabling goal-driven agents with HTN planning capabilities to scale and run parallelly. Our extension includes preferences over their objectives, preferences over their plans, actions, and moral values. We show the expressiveness of the extended platform with a sample scenario.

Keywords: Agent-based simulation · Goals · Preferences · Values

1 Introduction

Agent-Based Simulation (ABS) is a computational approach for simulating the activities and interactions of autonomous agents (individual or collective entities such as organizations or groups) in order to better understand how a system behaves. Furthermore, they allow for the simulation of complex environments where perception, decision-making processes and actions carried out are dispersed among several stakeholders or agents. The purpose of ABS is therefore to obtain explanatory insight into the behavior of a group of agents which share a common environment. ABS can be applied to many fields such as biology, social sciences, ecology, economics, policy-making, sociology.

Many ABS frameworks have been built focusing on large simulations to be run in High-Performance Computing (HPC) platforms. In current HPC-Based

ABS approaches (such as Repast [27], NETLOGO [24], and MASON [16]) models may be elevated to and examined at genuinely large scales at the expense of having agents with limited reasoning capabilities and/or limited interaction among them, sometimes even reducing agents to mere rule-based or functional input-to-output transformers. There are some simulation scenarios in which these simplifications of the agents' reasoning suffices (e.g., if we want to simulate the general traffic flow in a big city, it may be enough to have agents with simple behaviors who simply react to changes in the environment around them), but there are other scenarios (analysing complex human-human social relations or sociotechnical systems with intricate human-AI interactions) which require agents capable of more complex deliberative, goal-driven reasoning to simulate more complex behaviors including the effects of interactions with others.

There are many Multi-Agent frameworks in literature (such as Jadex [5], 2APL [9], BDI4Jade [17] or GOAL [12], to name a few) offering implementations of cognitive agents with more powerful practical reasoning capabilities, making them capable of exhibiting more complex social behaviours. Many of these implementations are inspired by the Beliefs-Desires-Intention (BDI) theory [7] and the BDI abstract architecture [20], modelling rational agents that use their beliefs about the current state of the world to choose which goal or goals to pursue, to then select actions or plans to fulfill the intended goal or goals. But this comes at the expense of having very limited scalability: the need to explore multiple potential instantiations of abstract goals ("which of all my goals are feasible/reachable now?") and plans ("which plans are applicable now?") in a given state of the system is computationally expensive. Many other approaches in literature offer different levels of reasoning and scalability ([1] and [21] provide an interesting comparative analysis on many of them, showing the reasoning level vs. scalability trade-off).

There is a need for new ABS platforms that could support big populations of goal-driven agents. Those ABS have the potential of being very useful in the creation and animation of richer social simulations to analyze the social relationships between agents by means of computational models of policies, norms, moral values and social conventions. By having agents whose reasoning and behaviour is influenced by these computational social models we can analyse how and when the agents adhere to the norms and moral values, how they affect and limit their actions, and how they may change over time as the agents interact with each other and their environment. A first step in this direction was presented by Gnatyshak et al. in [11]: a custom Python-based BDI-agent simulation framework capable of both hosting agents imbued with more powerful practical reasoning capacity *and* running simulations with large numbers of these agents. Scalability is tackled in this framework by parallelizing via PyCOMPSs [23] the reasoning cycle of goal-oriented agents, allowing them to run concurrently whenever possible.

In this paper we address the issue of further enhancing Gnatyshak et al.'s framework by giving agents the capability to deal with preferences over their objectives, preferences over the actions they take in order to accomplish those

objectives and (moral) values, as the next step towards a powerful agent-based micro-simulation framework to analyse the impact of social values, norms and conventions in large populations. In this work we also aim to explore how *far* we can go without using numbers in our preference mechanisms. Generally, humans do not reason using hard numbers (e.g., “today I prefer to go to the beach with a weight of 86, but to go to the cinema with a weight of 91; therefore, I will go to the cinema”) but in qualitative terms (e.g., “today it is raining; I would rather go to the cinema than to the beach; therefore, I will go to the cinema”) However, all state-of-the-art approaches we have analysed [6, 8, 19, 25, 26] end up adding hard numbers and/or ad-hoc numerical formulae to their selection strategy. So we aim to explore how *not* using numbers limits the expressiveness of our system, how severe this limitation is, and draw some conclusions as to whether it is acceptable to use numbers to attain a desirable level of complex reasoning.

This paper is structured as follows: in Sect. 2 we briefly describe the previous works we used as reference; in Sect. 3 we describe the conceptual model and how we added goals, preferences over goals, preferences over plans and actions, and support for the expression of moral values; in Sect. 4 we show how our additions to the model work in a sample scenario; and in Sect. 5 we conclude by discussing some limitations of the current model and extensions to be explored as future work.

2 Related Work

Our model of goals for Agent-Based Simulations in HPC has been inspired by two agent frameworks with working implementations: GOAP and BDI4JADE.

GOAP [18] is the AI created for the enemies of the video game F.E.A.R., mainly formalized by Jeff Orkin. It is relevant for our work as it provides goal-oriented agents in a multiagent gaming platform with strong scalability requirements. In GOAP, goals are represented by specifying a **desired state of the world** that agents strive to achieve. This desired state is described using the same structure used for the current state of the world, an agent’s beliefs, actions’ effects, etc. Agents can have many independent goals, but they can only pursue one at the same time. In order to plan, an agent must have a set of available actions, a set of beliefs about the world and sensors to periodically update those beliefs, and a set of goals. Each goal has a current priority, and the agent will choose to plan for the goal with the highest current priority. GOAP uses numeric priorities (i.e., a quantitative relation rather than qualitative). A* is used to plan with a heuristic minimizing the weighted number of actions used to reach the desired state., i.e., minimize the sum of costs of the actions in the plan. We borrow such goals defined as desired world states (see Sect. 3.1).

However, we aim to provide our agents with a more expressive goal model where agents may have a great number of declared goals, but only a few of them are intended to be achieved in a given point in time. Oliveira de Nunes’s BDI4JADE [17] platform provides a BDI layer on top of JADE [2]. It uses the

same structure as Orkin’s GOAP to represent goals (desired state of the world). It supports the declaration of different types of goals: *belief goals* (goals that deal with states of the world described by boolean variables), *beliefset value goals* (same as before, but variables are continuous or have more than two possible values), *composite goals* (used to represent goals composed of subgoals which have to be achieved sequentially or in parallel), etc. It also differentiates between desires (non-committed goals) and intentions (committed goals). Plans are an ordered set of actions and are executed to achieve a specific goal. In BDI4JADE agents do not have a set of actions that they can use to build plans, but rather, they have a library of plans that the agents can choose from. Each plan in the library has some applicability conditions (equivalent to actions’ preconditions) that are used in the plan selection process. We get inspiration from BDI4JADE on its plan selection strategy.

Our main inspiration for the modelling of preferences over goals comes from CP-nets [4]. Although our actual implementation is definitely not an implementation of a CP-net, the main inspirations we have drawn from them is to establish one default and many conditional preorder relationships over goals, and building a graph to both visualize them and interpret them. We also analysed Cranefield et al.’s approach in [8] to model values (to adapt it to model preferences over goals), but upon closer inspection, we decided not to follow this approach since it uses numerical values and in this work we aim for a more qualitative approach.

In the case of preferences over plans, we drew a great deal of inspiration from Visser et al.’s work in [25]. It introduces the concepts of goals’ properties, which we use extensively in our modeling of priorities over plans. We also make use of their mechanism for property propagation in our implementation. We should note that our implementation is simpler than theirs. For instance, the paper defines both properties of goals (discrete values that a property can take) and resources of goals (numerical values and intervals that represent how much of a resource -e.g., money, food- is being consumed by a goal or a sub-goal), but we chose to simplify the approach and add only discrete properties, as we want to explore a qualitative, scalar-free preference approach.

3 Conceptual Model

A **multi-agent system** for Agent-Based Simulation in HPC \mathcal{M} is defined as the tuple $\mathcal{M} = \{E, \mathcal{A}^+, \mathcal{C}\}$ where:

- E is the model of a simulated **environment**, in which the agents reside, that they can perceive, gather information from, and act on;
- \mathcal{A}^+ is a non-empty **set of agents**;
- \mathcal{C} is a **controller** (a structure that maintains the multiagents’ environment model, regulates how agents access and act upon it, and handles agent-to-agent communication within the HPC execution environment), which is defined as the tuple $\mathcal{C} = \{\mathcal{I}, inAcs\}$ where \mathcal{I} is the inbox for all the agents’ outgoing messages (supporting agent communication), and $inAcs$ is the set

of all the actions to be exercised on the environment (regulating how agents access and act upon it).

An **agent** is defined as $\mathcal{A}_i = \{ID, msgQs, outAcs, Bh, \mathbb{B}, \mathbb{G}, g_c, \mathcal{P}_c, \mathcal{MP}, \mathbb{P}_g, \mathbb{P}_p\}$ where:

- $ID = \{AgID, AgDesc\}$ is \mathcal{A}_i 's identity data:
 - $AgID$ is the unique identifier of \mathcal{A}_i
 - $AgDesc$ is an arbitrary description of \mathcal{A}_i
- $msgQs = \{\mathcal{I}, \mathcal{O}\}$ is the set of \mathcal{A}_i 's message queues
 - $\mathcal{I} = \{\dots, msg_i, \dots\}$ is the Inbox, the set of messages sent *to* \mathcal{A}_i
 - $\mathcal{O} = \{\dots, msg_i, \dots\}$ is the Outbox, the set of messages sent *by* \mathcal{A}_i
 - $msg_i = \{AgID_s, AgID_r, performative, content, priority\}$ is a **message** sent from agent with $ID = AgID_s$ to the agent with $ID = AgID_r$, with the corresponding (FIPA-like) performative type, content, and priority.
- $outAcs$ is the set of **external actions** to be executed on the environment. It is composed of tuples of the form: $\{senderID, a^e\}$, where ID is the sender's ID , and a^e is the action that is being sent.
- $Bh = \{\mathbb{RG}, \mathbb{P}\}$ is \mathcal{A}_i 's **role behavior**, which is composed by:
 - \mathbb{RG} is the set of **role goals** associated with the Bh which \mathcal{A}_i is enacting
 - \mathbb{P} is the set of plans \mathcal{P} associated with the Bh
- \mathbb{B} is the set of \mathcal{A}_i 's **beliefs**. It uses the same world state structure as E
- \mathbb{G} is the set of \mathcal{A}_i 's **own goals** (see Sect. 3.1).
- $g_c \in (\mathbb{G} \cup \mathbb{RG})$ is the current **committed goal** (see Sect. 3.1).
- $\mathcal{P}_c = \{\dots, ab_i, \dots\}$ is \mathcal{A}_i 's current **plan**, which is an ordered set of action blocks. Each **action block** $ab_i = \{\dots, a_{ij}, \dots\}$ is an ordered set of actions (each a_{ij} is an action). There are three types of actions: **internal actions** (actions that are executed by the agent in order to change their beliefs), **external actions** (actions that are sent by the agent to the controller in order to be executed on the environment to alter it), **message actions** (actions that are used to generate messages intended to other agents)
- \mathcal{MP} is the **metaplanner**, a library of plans for each goal (see Sect. 3.2).
- \mathbb{P}_g is the set of **preferences over goals** (see Sect. 3.3).
- \mathbb{P}_p is the set of **preferences over plans** (see Sect. 3.4).

Our conceptual model extends the one presented in [11]. Our extensions are described in the following sections.

3.1 Adding Goal Structure

We extend the conceptual model in [11] by providing a formal model for goals: *what* they are, *how* they are defined, and how they are *related with plans*. We have chosen to model goals as desired states of the world that agents strive to achieve. It is equivalent to the concept of **desires** in BDI Theory [7]. A goal is therefore defined by a collection of subsets of the variables that describe a state of the world (its **conditions**), and an assertion of desired value for each variable. These conditions are expressions such as 'cash==10' or 'speed>=50'

to mean that having exactly 10 units of cash and that maintaining a speed of 50 or above are part of the desired state of the world, respectively. Each subset describes a conjunction of variables that describe a desired state of the world and, in order for a goal to be considered achieved, it is required that the goal condition evaluates as *true* in the eyes of the agent (that is, according to its **beliefs**).

We formally define the structure of a **set of goals** \mathbb{G} as an unordered set of the form $\mathbb{G} = \{g_1, g_2, \dots, g_n\}$ where each g_i is an individual goal among the many goals an agent has. A **goal** is defined as $g_i = \{name, descr, \mathbb{C}, status\}$ where *name* is a unique identifier of the goal, *descr* is an optional text describing the goal, \mathbb{C} is the set of conditions over the state of the world for the goal to be considered achieved, and *status* is a boolean that is *True* if and only if the conditions \mathbb{C} are satisfied according to the agent's current beliefs \mathbb{B} .

A **set of conditions over the state of the world** is defined as unordered collections of assertions over the state of the world (the *environment*) of the form $\mathbb{C} = \{a_1, a_2, \dots, a_n\}$ where $a_i = \{n_1 \star v_1, n_2 \star v_2, \dots, n_m \star v_m\}$ is a conjunction of statements over the values of variables of the agent's beliefs, defined by n_i , which is the *unique* name of a variable of the agent's beliefs; \star , which is a binary operator ($\{=, \neq, >, \geq, <, \leq\}$); and v_i , which is the value of interest that is being asserted to n_i .

The agent possesses the capabilities to check whether or not an individual goal has been achieved according to its beliefs: $check_goal(g_i, \mathbb{B})$ outputs *True* if, according to the agent's beliefs, the conditions of the goal have been met, and false otherwise. Our agents are allowed to have multiple goals (own goals \mathbb{G} and role goals \mathbb{RG}), but are restricted to pursuing only one at a time. This *commitment* to a goal that is intended to be pursued (g_c in the agent tuple) is equivalent to the concept of **intention** in BDI. Agents have the capability to re-consider which goal they want to pursue, and may change the goal they are committed to even if they have not achieved it, depending on their current beliefs and the state of the world they perceive.

3.2 Adding a Library of Plans

We also extend [11] to enable specifying different plans for each goal, and to pick different plans for a committed goal with an element that will act as a library of plans. The implementation of the means-ends reasoner for the platform is a Hierarchical Task Network (HTN) planner [15]. A HTN is a tree composed of three types of nodes: (i) Primitive Tasks, (ii) Methods, and (iii) Compound Tasks. The root of the HTN is an abstract compound task (e.g., *order food*).

Figure 2 provides an example. Our agents have a library of predefined HTN plans that the agent can pick from, and these plans will be related to goals by means of the structure of the **metaplanner**, which is the \mathcal{MP} element of the agent tuple. Formally, it can be viewed as $\mathcal{MP} : \mathbb{G} \longrightarrow \mathbb{P}^*$, a matching relationship from goals towards plans, where \mathbb{P} is the set of plans \mathcal{P} associated with goal g_i and \mathbb{P}^* is used to indicate that it can output tuples of plans of arbitrary cardinality (meaning one specific goal may have, for instance, three

plans associated to it, while a different goal might have five, or two). We need also to add applicability conditions to plans: $\mathcal{P} = \{\mathbb{C}, ab_1, \dots, ab_n\}$, where \mathbb{C} is the set of conditions over the state of the world (see Sect. 3.1) that determine a plan to be applicable, and each ab_i is an action block.

Other noteworthy aspects of the metaplanner are that it incorporates appropriate functions for plan selection. Therefore, it will not simply act as a library/collection of plans, but it will also perform part of the reasoning. This reasoning includes both checking which of the associated plans are available for application, as well as ordering them based on the preferences.¹ For the first functionality, the metaplanner features a *get_available_plans*(g_i, \mathbb{B}) function which, taking into account the current beliefs of the agent, it outputs a subset of the set of plans associated with the goal, containing only all plans that are applicable. For the second functionality, the metaplanner has a *pick_plans*($g_i, \mathbb{B}, \text{prefs}_{\mathcal{P}}$) function, where $\text{prefs}_{\mathcal{P}}$ are the agent's preferences over plans, that will pick the plan that is more adequate to the current situation according to the agent's preferences and beliefs, from among all the applicable plans.

3.3 Adding Preferences Over Goals

The next extension we introduce in the model are preferences over goals. As we explained in Sect. 2 we drew inspiration from CP-nets and conditional preference formulas but we simplified the approach in order to be able to work without scalars, that is, having a fully qualitative approach for the specification of preferences over goals.

To define preferences over a set of goals, the approach we have taken is to establish a strict partial order relation between them to indicate which goals must be pursued before trying to achieve other goals. These binary relations between goals are reflexive, transitive and asymmetric. To model the context-dependent nature of preferences, we allow the declaration of conditional preferences, which are also a strict preorder relation over goals, but they only apply when their trigger conditions are met. A nice property of strict preorders is that they have always a unique direct acyclic graph (DAG) associated to them.

In order to encode **preferences over goals** in our agents, we have added the following element, \mathbb{P}_g (which stands for “Preferences over goals”) to the agent tuple. We define it as $\mathbb{P}_g = \{dGP, cGP_1, cGP_2, \dots, cGP_n\}$, where *dGP* are the *default* preferences over goals (they apply under ‘normal’ circumstances), and *cGP_i* are *conditional* preferences over goals (they have some trigger set of conditions \mathbb{C}_i over the state of the world as defined in Sect. 3.1).

The *dGP* and each *cGP_i* are defined as a DAG that corresponds directly to a **strict partial order** relationship between goals, and the only difference between them is that the *dGP* is the one active by default (it does not need any conditions to be met), while the various *cGP_i* become active and replace *dGP* if some associated conditions are true.

¹ We describe how we model preferences over plans in Sect. 3.4.

Once all the strict preorder relations have been established, we deduce their associated DAGs. From those DAGs, we compute a valid topological ordering of each, and these orders are the ones in which goals will be pursued by the agents (by choosing the first non-achieved goal in the topological ordering), e.g.:

- We have one agent \mathcal{A} , which has the goals $\mathbb{G}_0 = \{g_0, g_1, g_2\}$. g_0 is a goal to tidy the agent’s bedroom, g_1 is a goal to tidy the agent’s kitchen, and g_2 is a goal to store clothes that are hanging out to dry in the open.
- If we denote “goal i must be achieved before goal j ” as $g_i \rightarrow g_j$, the **default preferences** over goals of agent \mathcal{A} , are $\{g_0 \rightarrow g_2, g_1 \rightarrow g_2\}$, that is, before storing the clothes that are outside, \mathcal{A} , must have cleaned both his bedroom and his kitchen. Notice how both g_0 and g_1 must be accomplished before focusing on g_2 , but there is no established order between g_0 and g_1 , as it is a strict *partial* order. A valid topological ordering might be: g_0, g_1, g_2 , but also g_1, g_0, g_2 . By default, \mathcal{A} , will pursue his goals in either of those orders.
- The set of **conditional preferences** over goals of agent \mathcal{A} , is $\{g_2 \rightarrow g_0, g_2 \rightarrow g_1\}$ with the associated trigger conditions that the variable ‘raining’ must be *True*. If it is raining, the agent’s top priority goal will be to collect the clothes (g_2), then cleaning their kitchen or bedroom, in no specific order. Therefore, the moment it starts to rain, \mathcal{A} , will switch to any of the topological orderings that can be given to this set (for instance, g_2, g_1, g_0).²

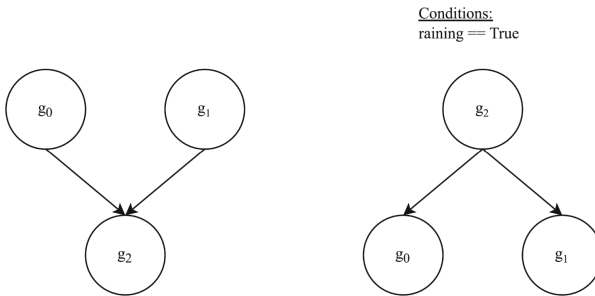


Fig. 1. Example of default and conditional preferences over goals

This example is depicted in Fig. 1. The left graph is the one deduced from the relations that defined the default preferences over goals, while the graph on the right-hand side is the one defined by the trigger condition (*raining = True*). A valid topological ordering of the left graph might be: g_0, g_1, g_2 , but also g_1, g_0, g_2 . By default, the agent will pursue his goals in either of those orders, but the moment it starts to rain, he will switch to any of the topological orderings that can be given to the right graph, for instance, g_2, g_1, g_0 , but also g_2, g_0, g_1 .

² In case of conflicts between preferences, the default behaviour is to choose by order of declaration in the HTN. This can be overridden by the designer. Refer to Sect. 5.

3.4 Adding Preferences Over Plans and Actions

By adding preferences over goals we provide agents with the capacity to choose *what* to pursue. But we also need to provide them with means to have preferences over *how* to achieve what they are pursuing. For example, when your goal is to eat, it is not the same to achieve that goal by eating a delicious pizza or to achieve it by eating a boring (but healthier), plain white rice, even if both actions achieve the goal all the same. We humans have preferences not only over *what* goals we want to achieve, but also over *how* we want to achieve them, and these preferences may be context-dependent. Some people might prefer to drive to their workplace, while some others would rather walk there. But the preference on walking may change in the case the weather is very cold or rainy, then preferring to commute to work by a combination of transportation modes. These examples provide us with further, key information: the preferences we have over how we achieve things are also context-dependent; we may wish achieve a specific goal by means of some actions under some circumstances, but under different circumstances we might prefer to achieve the same goal through different actions. Since the purpose of this work is to imbue agents with human-like social aspects for simulation purposes, we will need to take all these considerations into account when modeling preferences over plans and actions. In order to encode **preferences over plans and actions** in our agents, we have added element \mathbb{P}_p (which stands for “Preferences over plans”) to the agent tuple. We define it as $\mathbb{P}_p = \{gP_1, gP_2, \dots, gP_n\}$. We denote the preferences over plans for each goal g_i by $gP_i = \{dPP, cPP_1, cPP_2, \dots, cPP_n\}$, where dPP are the *default* preferences over plans for goal g_i (under ‘normal’ circumstances), and cPP_i are *conditional* preferences over plans for goal g_i (they have some trigger set of conditions \mathbb{C}_i over the state of the world).

A **property of a goal** is the name of a variable of interest that a goal has the capacity to alter. Said variable does not necessarily have to be the name of a variable in the set of beliefs of an agent. It is simply something noteworthy that achieving a goal has the capacity to give a specific set of values. For example, if a goal is to ‘order dinner’, some of the properties might be ‘vegetarian’ and ‘cuisine’, and their possibles values might be $\{True, False\}$ and $\{‘French’, ‘Italian’, ‘Spanish’, ‘Turkish’\}$, respectively. In our model each goal, plan, subplan, and action may have a set of properties PS , of the form $PS = \{prop_1, prop_2, \dots, prop_n\}$, and each property $prop_i$ is of the form $prop_i = \{v_1, v_2, \dots, v_n\}$ where: $prop_i$ is the *unique* name/identifier of the property, and v_i is one of the possible values that the property can take. These values can be boolean, numeric, etc., depending on the nature of the property itself. The set of values that make up each property are used to indicate possible values the property can take. All properties can have the special *None* value inside the set of their possible values. The presence of this value in a property of a plan or subplan indicates that said plan or subplan can be achieved through one or more actions that do not use or alter the property in question at all.

Propagation of properties consists in sending the properties ‘upwards’ from the most concrete actions, up to the root goal, passing through every sub-

plan and subgoal in the way. The full description of the method is provided in [25]. Given two *sequential* actions that have the same parent, the parent’s set of properties will be the result of computing the union between the two children’s properties. Each child will not have different possible values for the same properties, since they are sequential actions, and it would not make sense to design a plan in which child action no. 1 sets ‘cuisine’=‘Spanish’ only for the child action no. 2 to set the cuisine to be ‘French’. Therefore, the properties of the two (sequential) children will always be different, and the resulting properties of the parent node will simply be the joining of the children’s sets of properties, and it is trivial to see that this process applies to n sequential children actions.

Given two *alternative* actions that have the same parent, the parent’s set of properties will be the result of merging the properties of the children in the following manner: if both children set different values for the same property then, for the father, the values of the property will be the union of the values that the children had (e.g., if child no. 1 had ‘cuisine’=‘Spanish’ and child no. 2 had ‘cuisine’=‘French’, the parent task will have ‘cuisine’={‘Spanish’, ‘French’} to indicate that if that node is chosen, we will limit the possible values of ‘cuisine’ to those two values). If either child has a property that the other does not, the parent will simply take the same properties of the child that has it, and will add the special value *None*, to indicate that if that node is chosen, there is a path of the plan that accomplishes the goal without ever giving a value to that property.

Figure 2 provides an example of property propagation. It shows the set of plans associated to a goal of ordering dinner. There are three possible options: a plan to order burgers, a plan to order falafel, and a plan to order pizza. Let us assume for this example that there is only a local burger, a local falafel, and both a local pizza restaurant and a big company that makes pizza. Other assumptions that we take are that all burgers and pizzas are non-vegan, and that all falafels are vegetarian. The designer only needs to declare properties on the actions. Then, as a result of the property propagation process, all vertices have their own set of properties that have propagated upwards, from the leaves (actions). Notice how, in general, all properties have propagated towards the upward nodes. However, most of these propagations have been very simple ones: from single child to parent, although there are two cases worth mentioning. The first one is the propagation from the subplans to order local pizza and order from big pizza company. Notice how their properties are the same in all fields except for the ‘local’ field, with one holding it as True, and the other as False. However, these two *alternative* subplans share a common parent, and when their properties are propagated to it, they are merged in the way we described earlier: the parent has its property ‘local’ with *all* of its children values, to represent that, if that subgoal (or its parent subplan) is picked, then we can still order from either a local restaurant or a big chain. The other note-worthy example is the propagation of properties to the root node, where all options have been compiled in its properties.

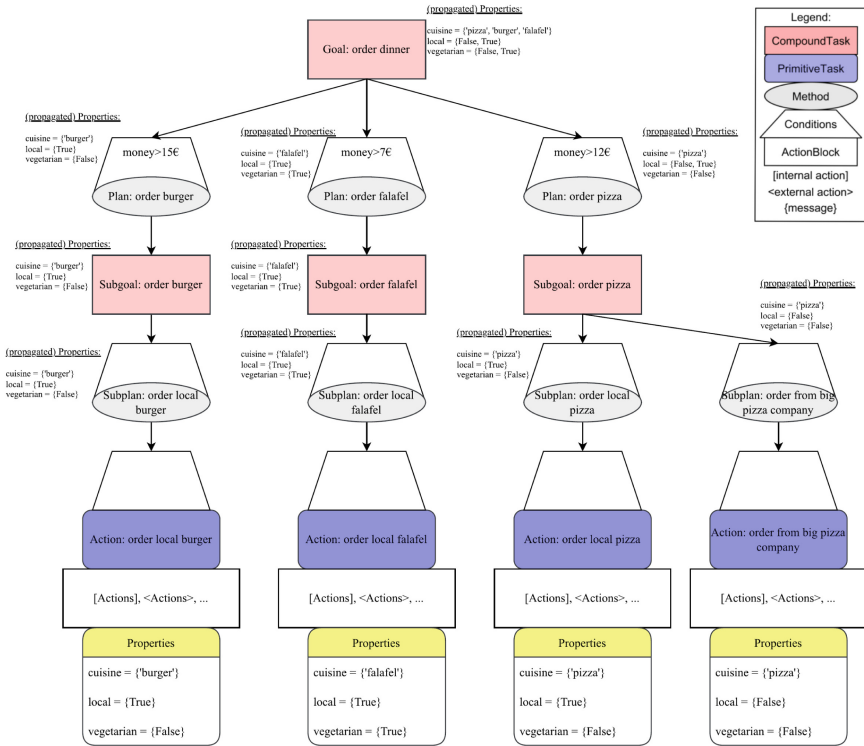


Fig. 2. Property propagation on an HTN plan associated to the *order dinner* goal

3.5 Selection of Plans and Actions Using Properties

We will now briefly describe the process of choosing a plan taking preferences into account. An assumption we make throughout this whole example is that all plans are available, that is, our choices are not restricted by the environment in any way, shape, or form. Given a concrete goal g_i (order dinner) an agent has a set of preferences over the plans to achieve g_i . We can define this set as $gP_i = \{dPP, cPP_1, cPP_2\}$, where dPP is the default set of preferences, and cPP_1, cPP_2 are conditional sets of preferences. The dPP and each cPP_i are all an *ordered* instantiation of the values of different properties of the goal's plans. We assume that we have the following preferences over how to achieve the goal to order dinner (see Fig. 2):

1. $dPP = \{cuisine = \{falafel\}\}$: by default the metaplanner would only follow the branch with this property, and order from the falafel restaurant.
2. $cPP_1 = \{cuisine = \{burger, pizza\}, local = \{True\}\}[weather = snowy]$: in case of snow the metaplanner would follow branches that are either burger or pizza cuisine, but only those that are local (in the case of pizza this restricts it to only the local pizza place option).

3. $cPP_2 = \{local = \{False\}, vegetarian = \{False\}, cuisine = \{burger\}\}[weather = rainy]$: in case of rain the metaplanner attempts to follow branches meeting all the conditions, but even if the agent prefers to order non-vegetarian burgers, the first property prevails and leads to the only non-local option (pizza from big company).

As we can see, the agent picks from all the plans that satisfy the leftmost property, then, from those plans, it picks from those that satisfy the next leftmost property, etc. This process is for both default and conditionally triggered preferences, as they have the same structure, the only difference being that the latter need to be activated in order to take over and replace the default properties.

3.6 Adding Moral Values

Moral values can be seen as an ordering of preferences [10] that may be used by (human and artificial) agents to evaluate both individual actions and world states [14]. The main idea is that actions and world states promoting the agent's moral values are preferred over others [10].

In our framework we model the influence of moral values in the selection of actions by using the system of preferences over plans and actions described in Sects. 3.4 and 3.5. Consider the previous example of ordering food. We can ingrain moral values into each plan as extra properties in their actions. For instance, in our food ordering example (see Fig. 2) primitive tasks are associated to a *local* value (meaning the social value to favour local businesses and products over globalization-oriented trade of products coming from far away) that can be connected to *Universalism* and *Self-Transcendence* in Schwartz's theory of human values [22]. Another example is provided in Fig. 3, where bike and walk options for transportations are positively associated to the *environmentalist* value (that also can be connected to *Universalism* and *Self-Transcendence*) and the *health* value (that can be connected to *Hedonism* and *Self-Enhancement*).

As we are associating moral values to the primitive tasks, this may look as if our model presupposes moral absolutism,³ but actually, that is not true. As properties are defined for each plan of each agent, we can create an agent who thinks that lying is morally wrong, and an agent that thinks that it is morally right. Also, since the same action can be part of different subplans, we can also encode the fact that the morality of actions depends on their context. For example, if an agent kills an animal as part of a subplan to have fun, we can label that action as morally evil, but if the same agent kills an animal in his job as a veterinarian, then that action can be labelled as not morally evil.

³ Moral absolutism is the position that there are universal ethical standards that apply to actions, and according to these principles, these actions are intrinsically right or wrong, regardless of what any person thinks, or context.

4 Example Scenario

We present a complex scenario to show how our agents fare with the new extensions: agents having many goals, goals decoupled from plans, preferences over goals, plans, and moral values.

In this Agent-Based Simulation, the agents' environment is a small town with some citizens living in it. These citizens are people which have their own set of daily goals (e.g., go to their workplace, have fun, eat dinner, etc.). Like real people, they have preferences over *in which order* to pursue their goals, as well as preferences over *how* to achieve them. Finally, they might have some moral inquiries into the actions we perform (e.g., being environmentalists and thinking the usage of cars is immoral, etc.).

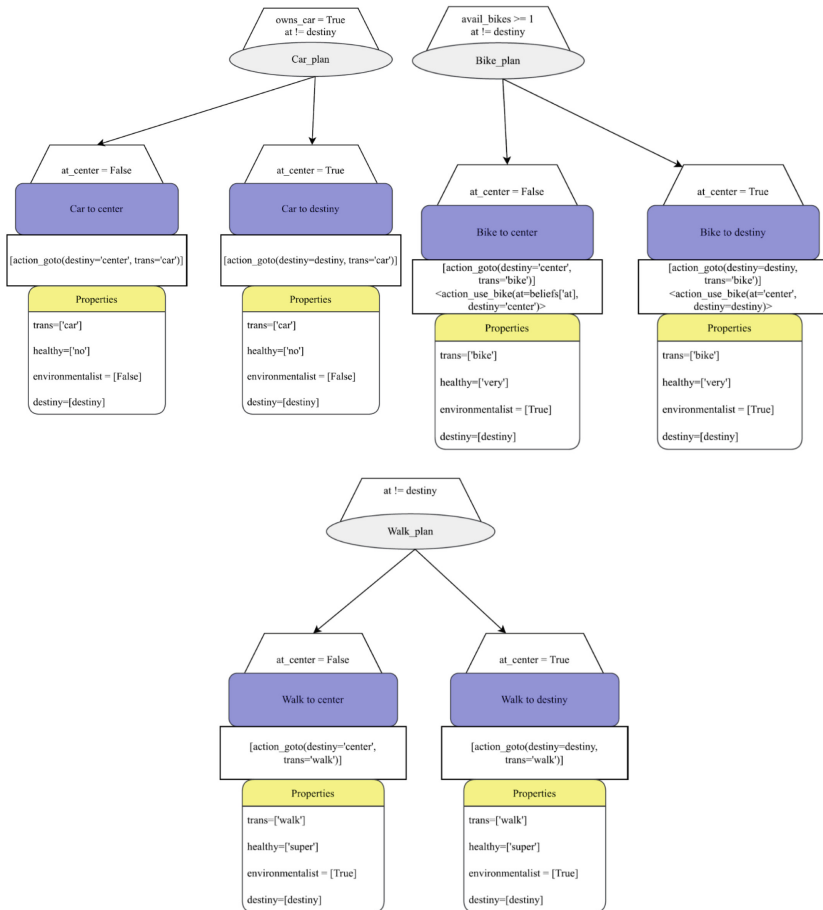


Fig. 3. Library of plans for transport goals

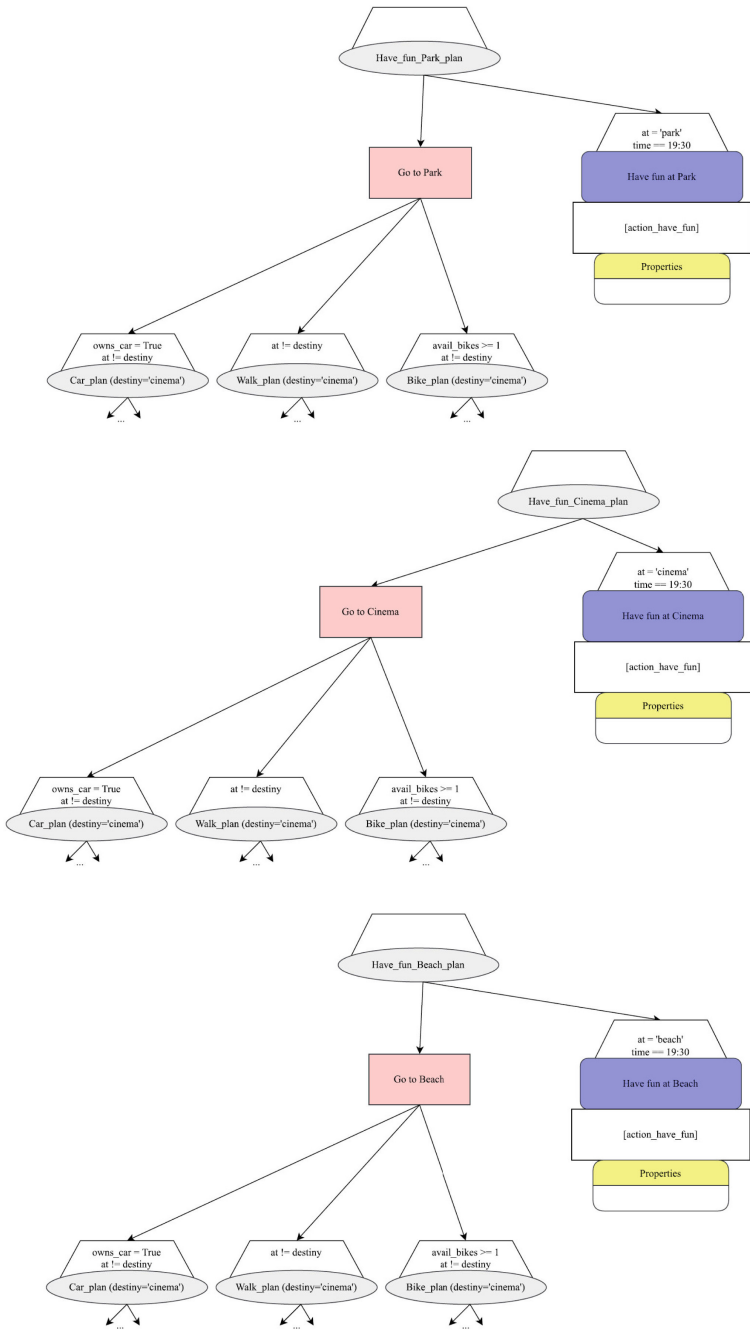


Fig. 4. Library of plans for fun-related goals

Table 1. Alice’s and Bob’s goals, preferences and values

ALICE’s self goals	ALICE’s role goals	BOB’s self goals	BOB’s role goals
g_6 - Go home g_7 - Eat dinner g_8 - Attend any medical emergency	g_1 - Take children to school g_4 - Go collect her kids to school g_5 - Have fun with her kids g_2 - Go to work g_3 - Work	g_3 - Have fun g_4 - Go home g_5 - Eat dinner g_6 - Attend any medical emergency	g_1 - Go to work g_2 - Work
ALICE’s preferences over goals Default: $[g_1 \rightarrow g_4 \rightarrow g_5 \rightarrow g_6], [g_1 \rightarrow g_2 \rightarrow g_3], [g_6 \rightarrow g_7]$ Conditional preferences: – if (medical emergency) $[g_8 \rightarrow g_1], [g_1 \rightarrow g_4 \rightarrow g_5 \rightarrow g_6], [g_1 \rightarrow g_2 \rightarrow g_3], [g_6 \rightarrow g_7]$ – if (snowing) $[g_2 \rightarrow g_3 \rightarrow g_6], [g_1 \rightarrow g_4 \rightarrow g_6], [g_6 \rightarrow g_7]$		BOB’s preferences over goals Default: $[g_1 \rightarrow g_2 \rightarrow g_4], [g_2 \rightarrow g_3], [g_4 \rightarrow g_5]$ Conditional preferences: if (medical emergency) $[g_6 \rightarrow g_1], [g_1 \rightarrow g_2 \rightarrow g_4], [g_2 \rightarrow g_3], [g_4 \rightarrow g_5]$	
ALICE’s values – For transport and fun-related goals : <ul style="list-style-type: none"> environmentalist = <i>False</i> – For food-related goals : <ul style="list-style-type: none"> local = <i>False</i> #big chains 		BOB’s values – For transport and fun-related goals : <ul style="list-style-type: none"> environmentalist = <i>True</i> healthy = $\{Super, Very\}$ – For food-related goals : <ul style="list-style-type: none"> local = <i>True</i> #local businesses 	
ALICE’s preferences over plans (transport goals) Default: $\{trans = \{car\}\}$		BOB’s preferences over plans (transport goals) Default: $\{trans = \{bike\}\}$ Conditional preferences: $\{trans = \{walk, bike\}\}[weather = cloudy]$ $\{trans = \{car\}\}[weather = \{rainy, snowy\}]$	
ALICE’s preferences over plans (fun-related goals) Default: $\{destiny = \{beach\}\}$ Conditional preferences: $\{destiny = \{park\}\}[weather = cloudy]$ $\{destiny = \{cinema\}\}[weather = \{rainy, snowy\}]$		BOB’s preferences over plans (fun-related goals) Default: $\{destiny = \{beach\}\}$ Conditional preferences: $\{destiny = \{cinema\}\}[weather = \{cloudy, rainy, snowy\}]$	
ALICE’s preferences over plans (food-related goals) Default: $\{cuisine = \{pizza\}\}$ Conditional preferences: $\{cuisine = \{chinese\}\}[weather = \{rainy\}]$		BOB’s preferences over plans (food-related goals) Default: $\{cuisine = \{pizza\}\}$	

Each day of the simulated city is discretized in 64 steps. The simulated day starts at 08:00, and ends at 00:00 of the next day. Each simulation step corresponds to 15 minutes in the town. By default, the town starts with clear weather. Every iteration, there is a 10% chance of the weather changing. If that chance happens, there is a 60% chance of the weather becoming clear, 30% chance of becoming cloudy, 9% chance of raining, and 1% chance of snowing. At every iteration, there is also a 0.2% chance, for every agent, to experience a medical emergency. All these parameters are configurable by the user. The environment is randomly generated using a *seed*, and the agents will react and plan accordingly to the changes on the environment.

The town is composed of locations. Agents can move from one location to another by means of transport plans. The *town center* is the central location that connects with the others. People can live in the city center or in other city locations (residential neighborhoods). There are places where people go to have fun (a beach, a park and a cinema). There are also workplaces (a factory and corporate offices). There exist some places to go shopping: a local market (which includes Italian, Chinese and falafel restaurants), a supermarket and a big shopping center (with a big chain pizza company, a fast food burger company and a big chain of wholefood/vegetarian meals company). The city has locations with some public services (a school for kids, a hospital to treat citizens).

Citizens might, by chance, experience a medical emergency, in which case, if they go to the hospital, they will be tended to and cured for free, so they can carry on with their day

The town can experience the following weather conditions: clear (sunny), cloudy, rainy and snowy. Those weather conditions may affect the citizens' choices (e.g., some may not use a bike if it rains). Some city services may be affected, too (e.g., schools are closed under snowy weather).

There are three main ways to go around the city: by car, by bike, or on foot. In order to drive a car, an agent needs to own one. In order to drive a bike, an agent needs to be at a location where a bike from the public rental system

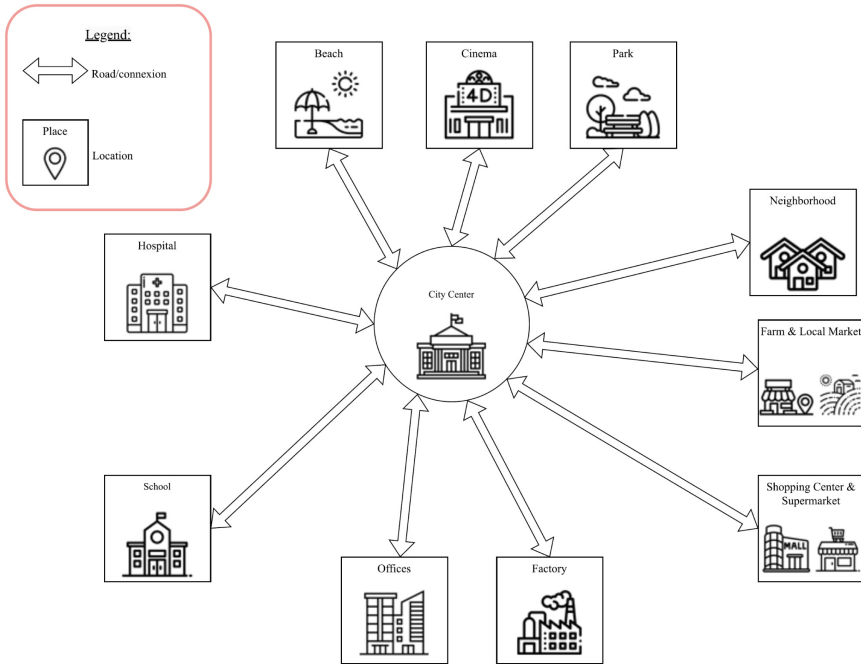


Fig. 5. Map of the town with its locations

is available, pick it, and leave it in another location. In our simulated city it is possible that some locations might not have any bikes at any given moment.

The `environment` class implements the map of city locations as well as other variables such as the current weather, the time, and extra internal variables for purposes of running the simulation. When an agent perceives the environment, they will only perceive the current time, the current weather, and the information of the location that they are currently in. For instance, if an agent is at the city center, it will not update its information about the state of the school, only about the state of the city center, the weather, and the time.

There are two main actors in our environment, **Alice** and **Bob**. They both are complex agents with numerous goals, conditional preferences over these goals, a rich library of plans, and preferences over those plans, along with moral values.

Alice is the CEO of a big company. She works at the office every day until 16:45. She has to take the children to school every morning, collect them from school at 17:00, and go have fun with them in the afternoons (until 19:45). Then, they order food at 20:00. Her initial beliefs are her current location, the current weather and time, the current location of her children, whether she owns a car, whether she has worked, if her children have gone to school, if she is at the center of the city, and whether there is a medical emergency. Table 1 shows her goals, preferences over goals and plans and her values. Alice's library of plans consists of three sets of complex plans: one set of plans for fun-related goals (see Fig. 4), one set of plans for transport goals (see Fig. 3) and one set for food plans (an extension of the one shown in Fig. 2 with an extra plan branch for Chinese food). Goals g_1 , g_2 , g_4 , and g_6 include commuting, and therefore are mapped to transport plans by the metaplanner. g_5 and g_7 are mapped to fun and order meal plans, respectively. The other plans for other goals are trivial: they have a single plan, with a single action (e.g., in the case of the plan to work, there is only one method, with a single action).

Bob is the second agent we have created for this test scenario. Like Alice, he has his own set of beliefs, a place where he lives, a place where he goes to work, preferences over how to have fun, etc. Bob lives in the city center and is a worker in the local factory, every day until 16:45. He has no children so he goes to work directly every morning. Once he is done, he goes to have fun however he prefers. Then he goes back home and orders food at 21:00. His initial beliefs are similar to Alice's, excluding those children-related. Table 1 shows Bob's goals, preferences over goals, plans and values. Goals g_1 , g_3 and g_4 include commuting and therefore are mapped to transport plans by the metaplanner. g_3 and g_5 are mapped to fun and order meal plans, respectively. Bob's goals are a subset of Alice's goals and are mapped to the same plans, but Bob will not act like Alice, as their personal preferences and moral values differ.

4.1 Tests and Results

In this section we show some execution runs to see that agents plan according to their goals, preferences and values, and that they respond to changes in the

environment that might cause them to reconsider their contextual preferences and, therefore, need to replan, or even reconsider their goals.

Figure 6 shows the result of a simulation with all default parameters except for `emergencyodds = 0.2` (20%). At step 35 we can see that Alice is working in her workplace when she receives a medical emergency of one of the kids. Then, *her conditional preferences over goals activate*, she changes her current goal, and she rushes to the hospital, as we can see in the next step. Although not shown in the picture, when she goes to the hospital and is cured, her preferences over goals revert to default, and she goes back to the offices to continue working.

In Fig. 7 there is the result of a simulation with all default parameters except for `changeodds = 1`, `rainodds = clearodds = 0.5`, and `clouddods = snowodds = 0`. At step 43, both agents were having fun at the beach. However, it suddenly started to rain, and then *their preferences over plans changed*. The goal (to have fun) does not change. What changes, however, is *how* they decide to have fun. According to their conditional preferences for fun, in case of rain they prefer to go to the cinema, and they replan giving priority in the HTN to the branches with the `destiny={cinema}` property.

Figure 8 shows an example of the interwork of conditional preferences over plans and values. The Observer Agent tells us that it is raining. In the case of Bob, his conditional preferences over food-related goals determine that its single, permanent, default preference is always pizza (see Table 1). Therefore, Bob’s HTN related to the “order food” goal (Fig. 2) will select the order pizza branches (except if Bob has less than \$12, then the order falafel branch will be explored). But to choose among the two order pizza sub-branches, Bob’s values (*local = True*) are used to make the choice. From the two possible options to order pizza, only “order local Pizza” has its local value True and is chosen (see Bob’s mental state in Fig. 8). The rainy weather has also triggered a change

```

==== STEP 35 ====
Agent Observer Agent's (id: 0) Inbox: Messagebox:
MSGs( from: 0
  To: 0
  Performative: state
  Content: {'observing': True, 'school': {'bikes': 2}, 'mall_super': {'bikes': 0}, 'center': {'bikes': 0}, 'farm_local': {'bikes': 0}, 'cinema': {'bikes': 0}, 'mehgh': {'bikes': 3}, 'hospital': {'bikes': 0}, 'park': {'bikes': 0}, 'factory': {'bikes': 3}, 'weather': 'clear', 'offices': {'bikes': 0}, 'PLAN': 'CompTask: Plan to m_observe', 'ttin': '16:30', 'needs': {'bikes': 3}, 'total_bikes': 5, 'GOAL': 'observing'})
Agent Alice's (id: 1) Inbox: Messagebox:
MSGs( from: 1
  To: 1
  Performative: state
  Content: {'has_worked': False, 'GOAL': 'g1', 'hour': 16, 'owns_car': True, 'children_went_school': True, 'last_transport': 'car', 'weather': 'clear', 'at': 'offices', 'medical_emergency': True, 'at_center': False, 'children_at': 'school', 'avail_bikes': 0, 'minute': 30, 'PLAN': 'CompTask: Plan to m_work_plan'})
Agent Bob's (id: 2) Inbox: Messagebox:
MSGs( from: 2
  To: 2
  Performative: state
  Content: {'has_worked': False, 'hour': 16, 'had_fun': False, 'GOAL': 'g2', 'owns_car': True, 'last_transport': 'walk', 'weather': 'clear', 'at': 'factory', 'medical_emergency': False, 'at_center': False, 'avail_bikes': 1, 'minute': 30, 'PLAN': 'CompTask: Plan to m_work_plan'})
==== STEP 36 ====
Agent Observer Agent's (id: 0) Inbox: Messagebox:
MSGs( from: 0
  To: 0
  Performative: state
  Content: {'observing': True, 'school': {'bikes': 2}, 'mall_super': {'bikes': 0}, 'center': {'bikes': 0}, 'farm_local': {'bikes': 0}, 'cinema': {'bikes': 0}, 'mehgh': {'bikes': 3}, 'hospital': {'bikes': 0}, 'park': {'bikes': 0}, 'factory': {'bikes': 3}, 'weather': 'clear', 'offices': {'bikes': 0}, 'PLAN': 'CompTask: Plan to m_observe', 'ttin': '16:45', 'needs': {'bikes': 3}, 'total_bikes': 5, 'GOAL': 'observing'})
Agent Alice's (id: 1) Inbox: Messagebox:
MSGs( from: 1
  To: 1
  Performative: state
  Content: {'has_worked': False, 'GOAL': 'g0', 'hour': 16, 'owns_car': True, 'children_went_school': True, 'last_transport': 'ambulance', 'weather': 'clear', 'at': 'center', 'medical_emergency': True, 'at_center': True, 'children_at': 'school', 'avail_bikes': 0, 'minute': 45, 'PLAN': 'CompTask: Plan to m_ambulance_plan to hospital'})
Agent Bob's (id: 2) Inbox: Messagebox:
MSGs( from: 2
  To: 2
  Performative: state
  Content: {'has_worked': True, 'hour': 16, 'had_fun': False, 'GOAL': 'g2', 'owns_car': True, 'last_transport': 'walk', 'went_to_work': True, 'weather': 'clear', 'at': 'factory', 'medical_emergency': False, 'at_center': False, 'avail_bikes': 1, 'minute': 45, 'PLAN': 'CompTask: Plan to m_work_plan'})
  
```

Fig. 6. Agent Alice changing preferences over goals

in his transportation means (car), which is fully mandated by his conditional preference over transportation plans. Here it is interesting to see that a conflict arises between the properties attached to the Car plan (`healthy={no}` and `environmentalist={false}`) and Bob's values (`(healthy={Super, Very}` and `environmentalist={true}`). As we have no numbers to rate the relative importance of conflicting preferences, we have to solve the conflict by explicitly placing in the scenario definition file the `trans` preference before the `healthy` one.

In general, we see that our agents react to changes in their current context by changing their priorities, and always plan according to them. Additionally, by looking at the whole verbose dump of a simulation, we see that they function as expected: they pursue their default goals in the correct order, change priorities over goals whenever they should, replan according to changes in both priorities over goals and plans, and make choices based on them.

5 Conclusions

In this paper, we describe an extension to an agent-based simulation environment for High Performance Computing enabling goal-driven agents with hierarchical

```

=== STEP 43 ===
Agent Observer Agent's (id: 0) inbox: Messagebox:
MSGs( From: 0
      To: 0
      Performative: state
      Content: {'observing': True, 'school': {'bikes': 2}, 'mall_super': {'bikes': 0}, 'center': {'bikes': 1}, 'farm_local':
{'bikes': 0}, 'cinema': {'bikes': 0}, 'neigh': {'bikes': 1}, 'hospital': {'bikes': 0}, 'park': {'bikes': 0}, 'factory': {'bikes': 0}, 'weather': 'rain', 'offices': {'bikes': 0}, 'PLAN': 'CompTask: Plan to: m_observe', 'time': '18:30', 'beach': {'bikes': 1}, 'total_bikes': 5, 'GOAL': 'observing'})
      Priority: False )
Agent Alice's (id: 1) inbox: Messagebox:
MSGs( From: 1
      To: 1
      Performative: state
      Content: {'has_worked': True, 'GOAL': 'g5', 'hour': 18, 'owns_car': True, 'children_went_school': True, 'last_transport': 'car', 'went_to_work': True, 'weather': 'rain', 'at': 'beach', 'medical_emergency': False, 'at_center': False, 'PLAN': 'CompTask: Plan to: m_have_fun_in_beach', 'children_at': 'beach', 'avail_bikes': 1, 'minute': 30, 'picked_children_school': True})
      Priority: False )
Agent Bob's (id: 2) inbox: Messagebox:
MSGs( From: 2
      To: 2
      Performative: state
      Content: {'has_worked': True, 'hour': 18, 'had_fun': False, 'GOAL': 'g3', 'owns_car': True, 'last_transport': 'car', 'went_to_work': True, 'weather': 'rain', 'at': 'beach', 'medical_emergency': False, 'at_center': False, 'avail_bikes': 1, 'minute': 30, 'PLAN': 'CompTask: Plan to: m_have_fun_in_beach'})
      Priority: False )
=== STEP 44 ===
Agent Observer Agent's (id: 0) inbox: Messagebox:
MSGs( From: 0
      To: 0
      Performative: state
      Content: {'observing': True, 'school': {'bikes': 2}, 'mall_super': {'bikes': 0}, 'center': {'bikes': 1}, 'farm_local':
{'bikes': 0}, 'cinema': {'bikes': 0}, 'neigh': {'bikes': 1}, 'hospital': {'bikes': 0}, 'park': {'bikes': 0}, 'factory': {'bikes': 0}, 'weather': 'rain', 'offices': {'bikes': 0}, 'PLAN': 'CompTask: Plan to: m_observe', 'time': '18:45', 'beach': {'bikes': 1}, 'total_bikes': 5, 'GOAL': 'observing'})
      Priority: False )
Agent Alice's (id: 1) inbox: Messagebox:
MSGs( From: 1
      To: 1
      Performative: state
      Content: {'has_worked': True, 'GOAL': 'g5', 'hour': 18, 'owns_car': True, 'children_went_school': True, 'last_transport': 'car', 'went_to_work': True, 'weather': 'rain', 'at': 'center', 'medical_emergency': False, 'at_center': True, 'PLAN': 'CompTask: Plan to: m_have_fun_in_cinema', 'children_at': 'center', 'avail_bikes': 1, 'minute': 45, 'picked_children_school': True})
      Priority: False )
Agent Bob's (id: 2) inbox: Messagebox:
MSGs( From: 2
      To: 2
      Performative: state
      Content: {'has_worked': True, 'hour': 18, 'had_fun': False, 'GOAL': 'g3', 'owns_car': True, 'last_transport': 'car', 'went_to_work': True, 'weather': 'rain', 'at': 'center', 'medical_emergency': False, 'at_center': True, 'avail_bikes': 1, 'minute': 45, 'PLAN': 'CompTask: Plan to: m_have_fun_in_cinema'})
      Priority: False )

```

Fig. 7. Agents Alice and Bob changing preferences over plans

task network (HTN) plans to choose among goals and among plans based on preferences and a simple moral values model. We have summarized extensions done on the agent model and how they work in a sample scenario. We have also been able to see how ‘far’ we could go without using any numbers to express preferences over goals, plans, and moral values. As we have seen, we have been able to express conditional preferences over both, have these preferences change based on context, and agents replan based on environmental changes. This work is one more step towards our goal to have a powerful agent-based micro-simulation framework to analyse the potential impact of social values, policies, norms and conventions in large populations of social-aware agents.

```

=== STEP 64 ===
Agent Observer Agent's (id: 0) Inbox: Messagebox:
MSG[s]( From: 0
  To: 0
  Performative: state
  Content: { 'observing': True, 'school': { 'bikes': 2 }, 'mall_super': { 'bikes': 0 }, 'center': { 'bikes': 1 }, 'farm_local': { 'bikes': 0 }, 'cinema': { 'bikes': 0 }, 'neighbor': { 'bikes': 1 }, 'hospital': { 'bikes': 0 }, 'park': { 'bikes': 0 }, 'factory': { 'bikes': 0 }, 'weather': 'rain', 'offices': { 'bikes': 0 }, 'PLAN': 'comptask: Plan to observe', 'time': '23:45', 'beach': { 'bikes': 1 }, 'total_bikes': 5, 'GOAL': 'observing' }
  Priority: False }
Agent Bob's (id: 2) Inbox: Messagebox:
MSG[s]( From: 2
  To: 2
  Performative: state
  Content: { 'has_worked': True, 'hour': 23, 'had_fun': True, 'GOAL': 'Idle', 'owns_car': True, 'last_transport': 'car', 'went_to_work': True, 'weather': 'rain', 'PLAN': 'Idle task', 'medical_emergency': False, 'at_center': True, 'has_ordered': 'local pizza', 'eaten_dinner': True, 'avail_bikes': 4, 'minute': 45, 'at': 'center' }
  Priority: False }

```

Fig. 8. Agent Bob has used his preferred means of transport for when it rains and his “local business” values to choose the local pizza option

One of the biggest limitations in how we declare goals is that, at any given moment, our agent can only pursue one goal at a time. This limitation is also common in many BDI-inspired implementations. Only few agent platforms (such as Jason [3] or 2APL [9]) allow to pursue several goals at the same time. We are already working on an extension of the model and its implementation to allow several goals at the same time, specially to allow handling combinations of achievement goals and maintenance goals. Another limitation is that our agents do not support adding (or removing) goals in runtime. Goals can be either achieved or not achieved at any given moment, but they cannot be eliminated (nor new goals can be added). This limitation was introduced for performance reasons. We plan to tackle this in future extensions.

Perhaps the biggest limitation in our declaration of preferences over goals and plans is that they are absolute, and this stems from the fact that we aimed to not use numbers in our model. Therefore, we cannot express things like ‘I prefer this *a little* more than that’, or ‘I prefer that *a lot* more than this’ that could be used to solve conflicts (such as Bob’s conflicts between the plan preference and his values). Visser’s et al.’s approach [25] provides a more complex structure that allows their agents to have more complex preferences (e.g., agents can reason about quantities, quantity optimization, limitation by quantity, etc.). Also, their agents are able to automatically extract properties of goals by looking at the actions, and then derive the relevant properties of the goals. Our model relies on the designer carefully listing (within the scenario description file) the properties

and the preferences in the *right* order. In future work we will explore more flexible and expressive ways to solve this (with no numerical values, if possible).

One related issue we plan to investigate further is related to what to do when the trigger conditions of non-default preferences over goals overlap (e.g., it is snowing and a medical emergency occurs), especially in the case they define different preorders. Our current approach is to pick the first goal preorder (by declaration order in the scenario file), and to allow the designer to implement an ad-hoc, more complex solution, if their scenario requires so. It would be better to modify our model to allow for a native way to handle this issue.

Finally, our encoding of moral values also totally relies on the designer carefully listing which actions have what moral implications and, while this is good from an expressiveness point of view (it allows us to declare moral relativism as different agents having different moral convictions) and context-dependent morality (the same action carried out under different circumstances having different moral implications), it is a very exhaustive and daunting task. It would be good to have the system partly automated, perhaps employing some matching between the purpose of an action and a value-tree structure rooted in a well-founded model of values (such as Schwartz's [22], which is used in [8,13,14]).

Acknowledgements. This work has been partially supported by EU Horizon 2020 Project StairwAI (grant agreement No. 101017142).

References

1. Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.: Agent based modelling and simulation tools: a review of the state-of-art software. *Comput. Sci. Rev.* **24**, 13–33 (2017)
2. Bellifemine, F., Poggi, A., Rimassa, G.: JADE—A FIPA-compliant agent framework, pp. 97–108. The Practical Application Company Ltd. <http://jade.csel.it/papers/PAAM.pdf> (1999)
3. Bordini, R.H., Hübner, J.F., Wooldridge, M.: *Programming Multi-Agent Systems in AgentSpeak Using Jason* (Wiley Series in Agent Technology). John Wiley & Sons Inc, Hoboken, NJ, USA (2007)
4. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: CP-nets: a tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.* **21**, 135–191 (2004)
5. Braubach, L., Pokahr, A., Lamersdorf, W.: Jadex: a BDI reasoning engine. In: *Multi-agent programming: Languages, platforms and applications*, pp. 149–174 (2005)
6. Casali, A., Godo, L., Sierra, C.: A graded BDI agent model to represent and reason about preferences. *Artif. Intell.* **175**(7), 1468–1478 (2011)
7. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *Artif. Intell.* **42**(2–3), 213–261 (1990)
8. Cranefield, S., Winikoff, M., Dignum, V., Dignum, F.: No pizza for you: value-based plan selection in BDI agents. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 178–184. IJCAI'17, AAAI Press (2017)
9. Dastani, M.: 2APL: a practical agent programming language. *Auton. Agents Multi Agent Syst.* **16**, 214–248 (2008)

10. Di Tosto, G., Dignum, F.: Simulating social behaviour implementing agents endowed with values and drives. In: Multi-Agent-Based Simulation XIII: International Workshop, MABS 2012, Valencia, Spain, June 4–8, pp. 1–12 (2012)
11. Gnatyshak, D., Oliva-Felipe, L., Álvarez Napagao, S., Padget, J., Vázquez-Salceda, J., Garcia-Gasulla, D., Cortés, U.: Towards a goal-oriented agent-based simulation framework for high-performance computing. In: Artificial Intelligence Research and Development: Proceedings of the 22nd International Conference of the Catalan Association for Artificial Intelligence, pp. 329–338. IOS Press (2019)
12. Hindriks, K.V., Roberti, T.: GOAL as a planning formalism. In: Lecture Notes in Artificial Intelligence, vol. 5774, pp. 29–40 (2009)
13. Kammler, C., Dignum, F., Wijermans, N.: Utilizing the full potential of norms for the agent’s decision process. In: Proceedings of the Social Conference 2022, Milan, 12–16 September (2022)
14. Kruelen, K., de Bruin, B., Ghorbani, A., Mellema, R., Kammler, C., Vanhée, L., Dignum, V., Dignum, F.: How culture influences the management of a pandemic: a simulation of the COVID-19 crisis. *J. Artif. Soc. Soc. Simul.* **25**(3), 1013–1020 (2022)
15. Kutluhan, E., Hendler, J., Nau, D.S.: HTN planning: complexity and expressivity. In: Proceedings of the AAAI Conference on Artificial Intelligence, 12, pp. 1123–1128 (1994)
16. Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K., Balan, G.C.: MASON: a multi-agent simulation environment. *SIMULATION Trans. Soc. Model. Simul. Int.* **81**(7), 517–527 (2005)
17. Oliveira de Nunes, I., Lucena, C., Luck, M.: BDI4JADE: a BDI layer on top of JADE. In: Proceedings of the 9th Workshop on Programming Multiagent Systems, pp. 88–103 (2012)
18. Orkin, J.: Three states and a plan: the A.I. of F.E.A.R. In: Game Developers Conference 2006. https://alumni.media.mit.edu/jorkin/gdc2006_orkin_jeff_fear.pdf (2006)
19. Padgham, L., Singh, D.: Situational preferences for BDI Plans. In: 12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013 2, pp. 1013–1020 (2013)
20. Rao, A.S., Georgeff, M.P.: An abstract architecture for rational agents. In: Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR-92), pp. 439–449 (1992)
21. Rousset, A., Hermann, B., Lang, C., Philippe, L.: A survey on parallel and distributed multi-agent systems for high performance computing. *Comput. Sci. Rev.* **22**, 27–46 (2016)
22. Schwartz, S.H.: An overview of the Schwartz theory of basic values. *Online Read. Psychol. Cult.* **2**(12), (2012)
23. Tejedor, E., Becerra, Y., Alomar, G., Queralt, A., Badia, R.M., Torres, J., Cortes, T., Labarta, J.: PyCOMPSS: parallel computational workflows in Python. *Int. J. High Perform. Comput. Appl.* **31**(1), 66–82 (2017)
24. Tisue, S., Wilensky, U.: NetLogo: a simple environment for modeling complexity. In: International Conference on Complex Systems, vol. 21, pp. 16–21 (2004)
25. Visser, S., Thangarajah, J., Harland, J., Dignum, F.: Preference-based reasoning in BDI agent systems. *Auton. Agents Multi Agent Syst.* **30**(3), 291–330 (2016)
26. Winikoff, M., Sidorenko, G., Dignum, V., Dignum, F.: Why bad coffee? explaining BDI agent behaviour with valuings. *Artif. Intell.* **300**, 103554 (2021)

27. Zia, K., Riener, A., Farrahi, K., Ferscha, A.: A new opportunity to urban evacuation analysis: very large scale simulations of social agent systems in repast HPC. In: 2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation, pp. 233–242 (2012)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





Social Value Orientation and Integral Emotions in Multi-Agent Systems

Daniel E. Collins^(✉) , Conor Houghton , and Nirav Ajmeri 

Department of Computer Science, University of Bristol, Bristol, UK
{daniel.collins, conor.houghton, nirav.ajmeri}@bristol.ac.uk

Abstract. Human social behaviour is influenced by individual differences in social preferences. Social value orientation (SVO) is a measurable personality trait which indicates the relative importance an individual places on their own and on others' welfare when making decisions. SVO and other individual difference variables are strong predictors of human behaviour and social outcomes. However, there are transient changes in human behaviour associated with emotions that are not captured by individual differences alone. Integral emotions, the emotions which arise in direct response to a decision making scenario, have been linked to temporary shifts in decision making preferences. In this work, we investigated the effects of modifying social preferences according to transient integral emotions in multi-agent societies. We developed *Svoie*, a method for designing agents that make decisions based on established SVO policies, as well as alternative integral emotion policies in response to task outcomes. We conducted simulation experiments in a resource-sharing task environment, and compared societies of *Svoie* agents with societies of agents with fixed SVO policies. We find that societies of agents that adapt their behaviour through integral emotions achieved similar collective welfare to societies of agents with fixed SVO policies, but with significantly reduced inequality in welfare among agents with different SVO traits. We observed that by allowing agents to adapt their policy in response to task outcomes, our agent societies achieved reduced social inequality.

Keywords: Individual differences · Social decision making · Simulation

1 Introduction

Social value orientation (SVO) is a spectrum of personality traits that describes individual differences in social preferences, in terms of the relative value an agent places on its own welfare and the welfare of others when making decisions [33, 34]. The SVO spectrum includes agents who are: *altruistic* or caring only for others, *cooperative* or caring both for self and others, and *selfish* or caring only for self. SVO is measurable in humans and considered to be relatively stable over time. Further, SVO has been found to be strongly correlated with patterns of social

behaviour through empirical study, such as the tendency to act cooperatively or individualistically [3, 4].

Seminal works from social psychology provide a clear conceptual model of the influence of SVO on individual preferences in social interactions. A robust framework for agent simulation has been developed, the ring model [22, 31], which defines utility functions for SVO traits that are now standard in multi-agent research. In a social dilemma, a rational agent would be expected to make decisions that maximise the utility associated with their individual preferences. However, humans are not rational agents, and they will not always seek optimal outcomes that would be expected for their stable characteristics. Through empirical studies, patterns of irrational decision making in humans have been linked to transient changes in affective state, emotions and mood states, resulting from changes in immediate circumstance or environment, or the consequence of longer-term contingencies or goals that interact with the current task. Emotions may serve an important role in adaptive decision making by motivating and guiding behaviour based on observations and judgements about the current context of the decision making environment and other within it.

Lerner et al. [30] outline two main categories of emotion, *incidental* and *integral*. *Incidental emotions* are task-unrelated emotions that arise in response to factors that are irrelevant to the current decision scenario, but which are nevertheless present during the decision making process. For example, a person who receives a frustrating message from a friend before an important meeting at work may be influenced by the unpleasant emotions during the meeting, even though they are task-unrelated, that could lead to impulsive decision making or unnecessary conflicts with colleagues. *Integral emotions* are task-related emotions that arise in direct response to the current decision, and are known to have a strong influence on behaviour. Integral emotions can be either *anticipated*, feelings about a potential future event or the possible outcome of an action, or *immediate*, feelings about a recent event or the observed outcome of an action. Our interest is in the latter, for example, the immediate integral emotion of feeling satisfied after performing well on an exam, and choosing to spend time helping others with their studies.

The “wounded pride” model of integral emotion [52] suggests that agents may react to unfair outcomes by feeling negative emotions, and acting spitefully, even when they know that it will result in a worse outcome for themselves on that specific task [40]. This is an example of how integral emotions can give rise to behaviour that is not explained by individual differences alone. Agents that adapt their policies based on integral emotion as in the wounded pride model may fare better than agents that only act based on SVOs, since some SVO policies may perform poorly on a given task compared to others. In this work, we investigated whether socially beneficial effects of altering social preferences according to integral emotions could be observed by modelling integral emotions in multi-agent societies with individual differences in SVOs.

Contributions. We developed *Svoie*, a method for designing agents that make decisions based on SVO and integral emotions. Our *Svoie* agents combine well-

established SVO decision making policies with a simple protocol for temporarily adopting alternative policies based on integral emotion. We define two alternative social-preference-based policies representing *positive* and *negative* emotions, that minimise or maximise payoff inequity respectively. These policies incorporate the wounded pride model of spiteful human decision making, and an idealised counter model for positive integral emotion. We model integral emotion as an internal state, that changes depending on the outcomes of recent decisions, and that defines the probability that an agent will adopt an integral-emotion-based policy in their next decision.

Findings. To evaluate *Svoie*, we conducted simulation experiments using a variant of the Colored Trails game [16, 18], a resource-sharing task environment designed for studying social decision making. We generated societies of agents with heterogeneous SVOs, and simulated sequences of games between random pairs of agents in the society. We compare the distribution of payoffs accumulated by agents between *Svoie* and *Stable-SVO* societies, and evaluate societal outcomes in terms of collective welfare, a measure of the total payoff to all agents in a society, and welfare inequality, a measure of the variation of payoff between agents.

We investigated whether *Svoie* societies would have lower welfare inequality relative to *Stable-SVO* societies, by allowing agents to adapt their social preferences based on the frequency with which they are succeeding or failing to achieve their individual goals. We find that societies of *Svoie* agents exhibit significantly lower welfare inequality than *Stable-SVO* agents in societies with more than one SVO, with a small reduction in collective welfare.

Organisation. Section 2 describes preliminaries necessary to understand our contribution. Section 3 describes our method for modelling SVO and integral emotions in agents. Section 4 presents our experimental setup, results, and evaluation. Section 5 concludes with a discussion of future directions.

2 Preliminaries and Related Works

We now introduce the preliminaries necessary to understand our contributions.

2.1 Social Value Orientation

The SVO model describes a continuum of orientation types, reflecting the nature of social preferences in decision making [33, 34]. SVOs are used in agent-based simulation to define agent decision making policies. SVO policies are typically implemented using the ring model of SVO [31]. In this model, an SVO utility function can be defined by any point on a unit circle, where the extent of preference for reward to self and to others is mapped to the x and y axes respectively. For example, this spectrum includes:

Altruistic Preference to take actions that increase the welfare of others, regardless of their own welfare.

- Cooperative** Preference to work with others to increase the welfare of themselves as well as others.
- Selfish** Preference to take actions that increase the welfare of themselves, regardless of the welfare of others.

These three SVO types cover the positive quadrant of the ring model, in which SVO utility functions only consider positive preferences for reward to self, other or both. The complete spectrum of SVO traits also includes negative preferences, for example, competitive agents have a preference for increasing their own reward while also reducing the reward of others. Different SVO decision making policies are well defined, and give predictable differences in performance in simulated social task environments [22]. The relative performance of SVO policies depends on the nature of the task.

Social preferences have been explored in the context of developing autonomous agents for applications in various real-world domains such as cyber-security [26], and SVO has been utilised to simulate social behaviour in autonomous vehicle decision-making [7, 12, 42]. Multi-agent simulation incorporating SVO has been used alongside experimental data to better understand how individual differences can influence cognition and behaviour to benefit societies, e.g., through social cooperation [2] and adapting to changes in environment [47], and SVO has been used in the simulation of normative multi-agent systems to understand the emergence of prosocial and cooperative behaviour [46]. Related works have looked at agent-based modelling of other individual difference variables, such as Myers-Briggs personality types [6]. In this work, we aim to better understand the relationship between emotion and social preferences through agent-based simulation.

2.2 Integral Emotions

Integral emotions describe task-related emotions that are directly influenced by the current decision making process, for example, an individual may experience positive or negative integral emotions depending on whether they achieve their goal on a particular task [52].

Seminal works in psychology shed light on the influence of integral emotions on human behaviour through empirical studies using ultimatum games [19]. An ultimatum game between two agents, Alice and Bob, can be described as follows: Alice and Bob are in separate rooms. Alice is told that Bob has been given an amount of money, and has been asked to share some of this money with Alice. Bob can offer any portion of the money to Alice that they choose. Alice can either accept this offer, or reject it. If Alice rejects the offer, neither Alice nor Bob receive any of the money, hence Bob's offer is an ultimatum.

A key finding of early work on ultimatum games is that people often reject small amounts of money despite the fact that this results in a worse outcome for themselves—they are rejecting “free money”. This finding has been replicated in numerous studies [51]. This may be thought of as a calculated spiteful behaviour, e.g., paying a cost in order to harm another. Emotional reactions like spite may

be considered in the context of social norms, pervasive expectations of certain behaviours within societies. Spiteful actions, in which a cost is paid to punish a perceived wrongdoer, may be adaptive behaviours which encourage cooperation norms, by enforcing sanctions in the form of punishments when cooperation norms are violated [39]. This could be extended to any norm related to how an individual expects that others should behave in a society, regardless of how they do. If an individual has a strong expectation for a particular norm, they may experience negative emotions when that norm is violated, and respond with spiteful actions. behaviour of this nature is common in online communities, for example, in commenting behaviours on the website Stack Overflow, [9].

The perspective of emotions as norm enforcing mechanisms is complicated by observations from ultimatum game experiments which show that spiteful behaviour may arise in the absence of any perceived social injustice, in the absence of any punishable perpetrator, and that once triggered, spiteful behaviour may be sustained and subsequently directed towards others arbitrarily. By altering the set-up of the ultimatum game, Straub and Murnigham [44] observed that participants sometimes rejected small offers even if they did not know the total amount of money from which the offer had been made, suggesting the rejection is not motivated by a sense of social inequity between participants. Further, they found that participants were just as likely to reject small offers when they did not know that the money had been split by another participant. They hypothesised that offers of small amounts of money were rejected because they evoked feelings of wounded pride, a direct emotional response to an unsatisfactory outcome. Pillutla et al. [40] conducted experiments using a sequence of ultimatum games between different pairs of participants, and found that participants who spitefully rejected a small offer would be more likely to take spiteful actions in subsequent games against new participants. In ultimatum games, individuals who receive an unsatisfactory offer may still try to act in retaliation, even if they cannot cause a disadvantage to the proposer of the unfair offer, suggesting that spiteful actions are a form of emotional release, or an expressions of internalised emotions [50]. The emotion may arise due to norm violation, but the resulting action may not be a calculated effort to enforce that same norm. More recently, Criado et al. [11] have explored role of emotions as motivators for norm compliant decision making towards the development of autonomous agents act in accordance with human norms.

These works describe a model of wounded pride, in which undesirable task outcomes can provoke a strong negative emotional response, which is expressed through subsequent non-cooperative behaviour. If an agent perceives that an outcome is unfair and unduly negative to them or contrary to an expectation of self-worth, feelings of wounded pride and anger are aroused which will influence their subsequent actions even if those actions cannot lead to a redress of the perceived wrong. In other words, when an individual experiences negative emotions in response to an unsatisfactory outcome, but cannot directly express these emotions to some perceived wrongdoer, they are nevertheless willing to retaliate by making sub-optimal decisions, which disadvantage others at some cost

to themselves. This mechanism may be beneficial in protecting altruistic agents from being repeatedly taken advantage of by self motivated agents. Conversely, we can conceive of a counter mechanism to wounded pride, wherein disproportionate success may elicit positive emotions, which in turn influence an agent to temporarily relax their preferences for high payoff and promote generosity. This aligns with ideas from social psychology on behaviour changes associated with positive emotion [21, 48].

A common method of monitoring integral emotions in human studies is through self reporting of emotion valence, the degree of positive or negative feelings at a particular moment in time. This derives from the appraisal theory of emotion [36], which posits that human emotions are internal phenomena, constructed through the appraisal of external events and stimuli, for example by evaluating whether an event outcome aligns with personal goals or norm expectations. Valence has been used in autonomous agent research to define internal states related to emotions, for example, to define intrinsic rewards for guiding the behaviour of reinforcement learning agents [20], and as a component of comprehensive decision making architectures based on psychological theories [13]. These related works often make use of other components of appraisal theory, such as arousal and motivation. For simplicity, we will focus on the valence of integral emotion associated with task outcomes. A similar approach has been taken previously to investigate the relationship between emotions and behavioural norms [35].

2.3 Social Task Settings

Simulations of agent behaviour in game environments can be directly compared to human decision-making data on the same or similar tasks or used as an abstraction of complex real-world social decision-making scenarios. In stochastic games, random variations in the parameters of the game’s setup and the agents involved in the game can give rise to a variety of different emergent scenarios. Sequences of stochastic games of varying complexity have been used to approximate complex real-world task environments for studying the influence of emotion and social factors on behaviour, both in empirical human studies and agent simulation [8, 12]. There is a breadth of work in which stochastic games have been used to study the relationship between SVO and social behaviour [3]. Stochastic games have also been used to study how emotions influence behaviour. Bono et al. [5] use a stochastic resource-allocation game to study how emotions mediate SVO preferences in human decision making.

Colored Trails (CT) [16, 18] is a research test bed designed for studying social factors in decision making. In CT, agents enter into a negotiation [27] and exchange resources to achieve their own individual goals. CT can be described in terms of generic elements of the task setting:

- Agents have individual goals they try to bring about.
- Agents have individual resources they can use to bring about their individual goals.

- Agents receive a reward upon achieving their goals.
- Individual circumstances of agents may vary, and therefore they may require different resources to bring about their goals compared to their peers.
- Agents may have insufficient resources to bring about their goals, or they may have surplus resources.
- Agents may negotiate an exchange of resources to help each other reach their goals.

CT is a highly flexible and expressive stochastic game, with various parameters that can be modified to customise the task environment. We chose to adopt CT as an environment for evaluating our agent societies, as it benefits from a clear task setting, and the random elements in the game’s set-up allow agents to encounter different unique social tasks over a sequence of games [23].

3 Method

We now detail our implementation of the CT game environment, agent decision-making policies, and agent models.

3.1 Simulation Environment

We implemented a simplified version of CT as a simulation environment for studying *Svoie* and *Stable-SVO* agent societies, based on an existing Python implementation from Sloan and Ajmeri [43].

A game of CT is played between two agents, and it consists of two separate rounds. At the start of each game round, a new game-board is generated: a 4×4 grid of coloured tiles, where each tile is randomly assigned one of four possible colours (red, blue, green, yellow). Each agent is then placed on the game-board at separate random starting positions. A random goal position is then assigned on the game-board, which is not vertically or horizontally adjacent to either agent’s starting positions. At the start of each round, each agent is allocated resources—a set of four randomly coloured chips—that agents can place to move to an adjacent position on the board where the chip colour matches the tile colour. The objective of the game is to get as close as possible to the goal position using the allocated resources. We assume agents have access to full information about the state of the game, e.g., the game-board, agent positions, goal position, and the resources of both agents.

Once per round, the agents may negotiate and exchange some or all of their resources to help each other reach the goal. During negotiation, one agent takes the role of *Proposer* and the other takes the role of *Responder*. The *Proposer* sends a proposal to the *Responder* comprising an offer, chips they will send from their own inventory, and a request, chips they want to receive from their opponents inventory. The *Responder* can then either accept the proposal, initiating the proposed exchange, or decline the proposal, meaning there is no-exchange and both players are left with their original allocated resources. Agents can then

use their resources to move as close as possible to the goal position, and receive a score, S , at the end of the round:

$$S = n + 1.5u(1 + g) \tag{1}$$

where n is the number of unused chips remaining in the agents inventory, u is the number of tile-chips used to create a path and g is equal to 0 or 1 depending on whether or not the agent reached the goal position respectively. This scoring function is taken from [24], and is designed to prioritise goal achievement strategies over strategies which seek to maximise score by gathering tiles, or creating long paths to arbitrary positions. Agents switch their negotiation roles between the two rounds of the game, so that each agent has one round as *Proposer* and one round as *Responder*.

By only allowing one offer and response per game round, CT becomes a more expressive form of the traditional ultimatum game discussed in Sect. 2. Here, the *Responder* can only choose between two possible outcomes: the ultimatum offer sent by the *Proposer*, or the no-exchange outcome determined by the randomised parameters of the game set-up. Random variations in individual circumstances and individual goals are encoded in CT through random variations in game-board set up, resource allocation, starting positions and goal positions.

Figure 1 shows a schematic example of one possible CT set-up, demonstrating how agents can cooperate to achieve a greater reward.

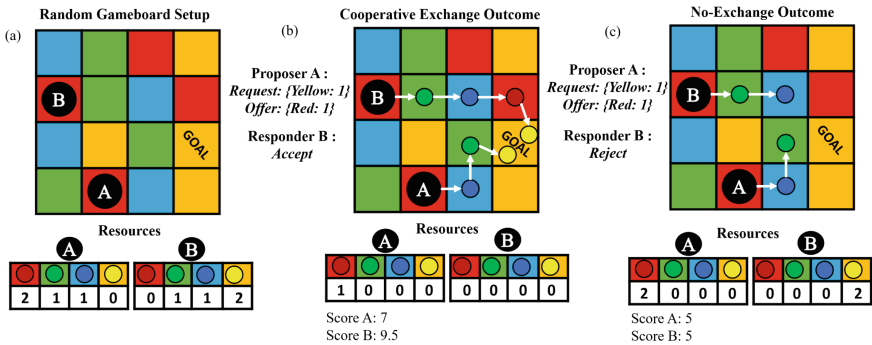


Fig. 1. Schematic example of one round of CT between agents A and B. **a** Random game-board setup parameters are generated at the start of the game: coloured tiles, agent positions, goal position and allocated resources. In CT, the resources are coloured chips that agents can use to move to an adjacent tile with the same colour. In this illustrated setup, neither agent can reach the goal using their initial resources. **b** A possible game outcome is shown, where B has agreed to A’s mutually beneficial exchange proposal; A sends one red chip to B, and B sends one yellow chip to A. Agents then use their resources to reach the goal, and receive a score according to Eq. 1. **c** Alternatively, in the no-exchange outcome, B chooses to reject A’s proposal, and agents must move as close to the goal as they can with their initial resources. Here, this results in a lower score for both agents.

3.2 Utility Functions for Social Preferences

To design agent decision-making protocol for the CT environment, social preferences and possible actions were mapped to quantitative utility functions. In each case, the agent perceives their environment, and uses the available information to select an action. An action is selected if it is expected to maximise the utility associated with the agents social preferences, a function of the game scores expected to result from an action, calculated using the scoring function in Eq. 1. The way in which an agent uses the utility function depends on whether it is acting as a *Proposer* or *Responder*.

Let, x , be an arbitrary exchange outcome, e.g., the resources that each agent possesses after the an exchange. If we assume that an agent will always use their chips optimally to achieve the highest possible score, each exchange outcome x maps directly to a pair of scores $S - P(x)$ and $S - R(x)$ for the *Proposer* and *Responder* respectively for a given game set-up. We can therefore define our utility functions in terms of x .

An agent acting as *Proposer* uses a chosen utility function as a ranking criteria to select a proposal. The *Proposer* calculates the utility associated with each exchange outcome, x , from the set of all possible exchange-outcomes, X , then selects the outcome with the greatest utility, and sends the corresponding proposal that would result in that outcome if accepted by the *Responder*. A *Responder* will accept a proposal only if it maximises a utility-based acceptance criteria relative to the no-trade outcome \bar{x} , the random set of resources possessed by each agent if no-exchange takes place. Here, the expected score for the no-trade outcome, \bar{x} , can be denoted $S - P(\bar{x})$ and $S - R(\bar{x})$. A proposal is only accepted if the utility of the proposed exchange is greater than the utility of the no-exchange outcome for the *Responder*.

Utility functions for socially oriented decision-making protocols are outlined for CT [15,17] based on different social preferences. We adapted these utility functions to describe agent protocols for our implementation of CT:

Individual Benefit the utility is the proposer score

$$U - r(x) = S - R(x) \quad (2)$$

or the responder score.

$$U - p(x) = S - P(x) \quad (3)$$

Aggregate Benefit the utility is the cooperative score, the sum of the proposer and responder scores.

$$U - c(x) = S - P(x) + S - R(x) \quad (4)$$

Outcome Fairness (Advantage of Outcome) the utility is the advantage achieved by the responder.

$$U - a(x) = S - R(x) - S - P(x) \quad (5)$$

Trade Fairness (Advantage of Trade) the utility is the advantage achieved by the responder, relative to rejection.

$$U - f(x, \bar{x}) = (S - R(x) - S - R(\bar{x})) - (S - P(x) - S - P(\bar{x})) \quad (6)$$

It is important to note that these functions are written from the perspective of the *Responder* so that they are positive when the action benefits the *Responder*. When used by the *Proposer*, the subscripts P and R are switched.

3.3 Agent Decision-Making Policies

In this section, we adapt the social-preference-based utility functions outlined in Sect. 3.2 to construct decision-making policies corresponding with altruistic, selfish and cooperative SVOs, and positive and negative integral emotions. We use these policies to develop baseline *Stable-SVO* agents, which always make decisions according to a fixed SVO-based policy, and *Svoie* agents, which act according to an SVO-based policy by default, but may temporarily adopt an integral-emotion-based policy in response to game outcomes in CT.

SVO Policies. Baseline *Stable-SVO* agents were created such that each agent has one of three possible SVO traits: selfish, altruistic or cooperative. Each SVO describes a fixed decision-making policy with a utility function reflecting social outcome preferences.

Selfish A selfish agent takes actions which maximise their own payoff.

- Proposal Ranking Criteria:

$$\text{maximise } U - p(x) \quad (7)$$

- Response Acceptance Criteria:

$$\text{accept trade if and only if: } U - p(x) > U - p(\bar{x}) \quad (8)$$

Cooperative A cooperative agent takes actions which maximise mutual payoff.

- Proposal Ranking Criteria:

$$\text{maximise } U - c(x) \quad (9)$$

- Response Acceptance Criteria:

$$\text{accept trade if and only if: } U - c(x) > U - c(\bar{x}) \quad (10)$$

Altruistic An altruistic agent takes actions which maximise payoff to others.

- Proposal Ranking Criteria:

$$\text{maximise } U - r(x) \quad (11)$$

- Response Acceptance Criteria:

$$\text{accept trade if and only if: } U - r(x) > U - r(\bar{x}) \quad (12)$$

Integral Emotion Policies. We devise two integral emotions policies to capture temporary changes in social preferences resulting from positive or negative integral emotions. Here, the integral emotion policies describe social outcome preferences that are not captured in *Stable-SVO* policies. The negative emotion policy, *competitive equity aversion*, is one which is expected to result in achieving a higher score with the largest margin of difference between the agent and its opponent (“Advantage of Outcome”) or “unfair” proposal). Conversely, the positive emotion policy, *inequity aversion*, is one which will minimise the margin of difference between the resulting scores. These are distinct from SVO policies as they do not consider game score maximisation.

Positive Integral Emotion (Inequity Aversion). An agent with positive integral emotion valence takes “fair” actions that minimise the difference in payoff between themselves and others.

- Proposal Ranking Criteria:

$$\text{minimise } 1/(1 + |U - a(x)|) \quad (13)$$

- Response Acceptance Criteria:

$$\text{accept trade if and only if: } U - f(x, \bar{x}) < 0 \quad (14)$$

Negative Integral Emotion (Competitive Equity Aversion). An agent with negative integral emotion valence takes “unfair” actions that maximise the difference in payoff between themselves and others, and for which the payoff to themselves is greater than that to others.

- Proposal Ranking Criteria:

$$\text{maximise: } 1/(1 + |U - a(x)|) \quad (15)$$

- Response Acceptance Criteria:

$$\text{accept trade if and only if: } U - f(x, \bar{x}) > 0 \quad (16)$$

Internal Emotion State for *Svoie*. We adopted standard decision-making protocols for altruistic, cooperative, and selfish SVOs to form baseline *Stable-SVO* agents, where agents always make decisions which align with their SVO. We then introduced an integral emotion component to the *Stable-SVO* agents to produce a *Svoie* agent—an agent that has an SVO, as well as positive and negative integral emotion policies. We designed *Svoie* agents so that positive integral emotion would be associated with reaching the goal in a round of CT, and negative emotion with not reaching the goal. To encode integral emotion in *Svoie*, we define an internal state $E \in \{-1, -0.5, 0, 0.5, 1\}$ representing the current valence of the agent, e.g. the positiveness or negativeness of their integral emotion. This is an internal state that is updated based on goal achievement at the end of each game round. For simplicity, we allow E to take one of five

discrete states between -1 and 1 , however, a higher granularity or continuous implementation could be used.

In the CT game, goal achievement results in a step increase in E and conversely, goal non-achievement results in a step decrease. We use E to define the probability that an agent selects an integral-emotion-based policy. $E = 0$ represents a neutral emotion state, in which the agent always defaults to its baseline SVO decision-making policy. When $E = 0.5$ or $E = -0.5$, the agent will have a 50% chance of selecting the positive or negative emotion policy respectively, and when $E = 1$ or $E = -1$, the agent will always select the associated emotion policy. In this way, agents can exhibit varying degrees of emotion-based behaviour over many repeat interactions depending on how frequently their decision-making policy causes them to achieve or miss their goals. The state E is designed to reflect the “appraisal theory” of emotion [36], that posits that human emotions are internal phenomena, constructed through the appraisal of external events and stimuli, for example, by evaluating whether an event outcome aligns with personal goals or expectations.

4 Experiments and Results

We conducted simulation experiments using CT (Sect. 3.1) as a task environment. We repeat our experiments using four different agent societies, which we define based on the proportions of agents with different SVO trait:

altr-coop Agent society with equal number of altruistic and cooperative agents
altr-self Agent society with equal number of altruistic and selfish agents
coop-self Agent society with equal number of cooperative and selfish agents
mixed Agent society with number of altruistic, cooperative and selfish agents

Each simulation is run over 1,000 time steps. At each time step, each agent in the society is paired with another agent at random, and each pair of agents plays two rounds of CT and receives a score. We compare simulations of *Svoie* agent societies to simulations of *Stable-SVO* societies.

Stable-SVO Agents follow fixed decision-making rules associated with their SVO.

Svoie Agents act the same as *Stable-SVO* initially, and have an SVO trait, but may deviate from their stable SVO trait based on game outcomes.

We define metrics and hypotheses in Sect. 4.1, for evaluating whether the integral emotion mechanism introduced in *Svoie* has a beneficial effect on societal outcomes at the end of the simulations.

4.1 Evaluation Metrics and Hypotheses

We define and compute *Individual Welfare*, *Collective Welfare* and *Welfare Inequality* for evaluating simulated *Svoie* and *Stable-SVO* agent societies.

Welfare measures the success of agents in maximising their score. We calculate the mean score achieved by individual agents and across samples of agents to evaluate welfare.

Inequality measures inequality of outcomes between members of an agent society. We assess inequalities over distributions using the Coefficient of Variation (CoV) measure [32]. Whereas Gini Coefficient is used in other research to measure inequality, we select CoV for its simplicity, and because the distributions of individual measures are observed to be approximately normal in preliminary runs.

1. **Individual Welfare** The mean score an individual agent achieves over all time steps in a simulation run.
2. **Collective Welfare** The mean score over a sample of agents.
3. **Welfare Inequality** The CoV of the distribution of individual welfare of agents in a sample. The magnitude of this measure is smaller for more equal societies.

We evaluate two hypotheses corresponding to the evaluation metrics for simulated agent societies.

H1 *Svoie* gives greater collective welfare than *Stable-SVO* over all agents in a society.

H2 *Svoie* gives lower welfare inequality than *Stable-SVO* over all agents in a society.

4.2 Simulation Setup

We simulated a sequence of CT games, described in Sect. 3.1, between random pairs of agents in each multi-agent society. At each time step, all agents are randomly paired, and each pair of agents plays two rounds of CT. Each simulation was performed over 1,000 time steps with a population size of 300 to account for random variations in game set-up and agent pairings at each time step. For each game round, we record the scores achieved by each agent. At the end of each simulation, we compute the metrics listed in Sect. 4.1. For *Svoie* agents, we initialised integral emotion to $E = 0$, so that all agents start by using the policy associated with their SVO trait.

The results presented are derived from the average of three repetitions for each simulation. We conducted tests to identify significant differences in our evaluation metrics between *Svoie* and *Stable-SVO*, across entire societies and specific samples of agents with a particular SVO trait. We use a two sample t-test, and report the means, μ , and p-values, p , and measure effect size as Cohen's d [10].

4.3 Evaluation

To evaluate hypotheses H1 on collective welfare, and H2 on welfare inequality, we compared *Svoie* to *Stable-SVO* for the four societies described in Sect. 4: *altr-coop*, *altr-self*, *coop-self* and *mixed*.

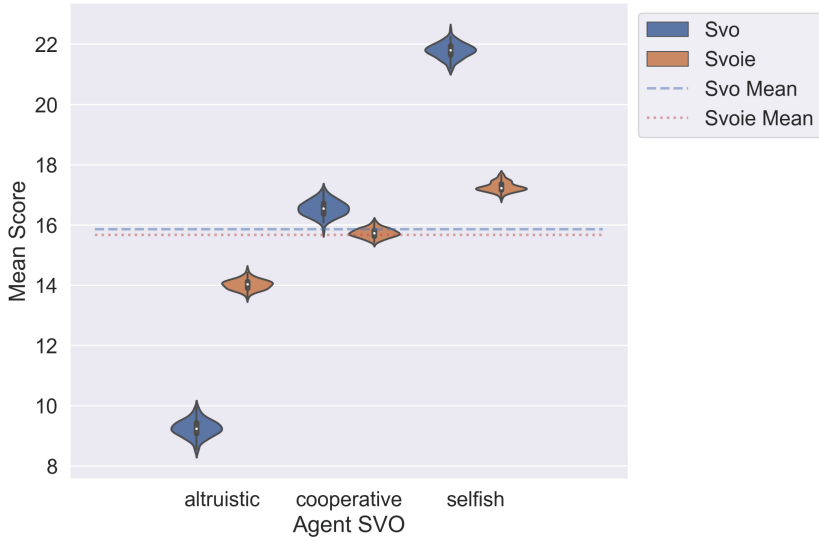
Table 1 compares population metrics measured for *Svoie* and *Stable-SVO* agent societies: (1) mean game score (Mean Score) achieved by all agents in a society, and in samples of agents with the same SVO, as a measure of the collective welfare achieved by those groups; (2) the coefficient of variation (CoV) of the distribution of mean welfare for individual agents in each group as a measure of welfare inequality. All results are calculated from three repeat runs.

Table 1. Comparison of the mean score and coefficient of variation in societies of *Stable-SVO* and *Svoie* agents with various combinations of SVO traits.

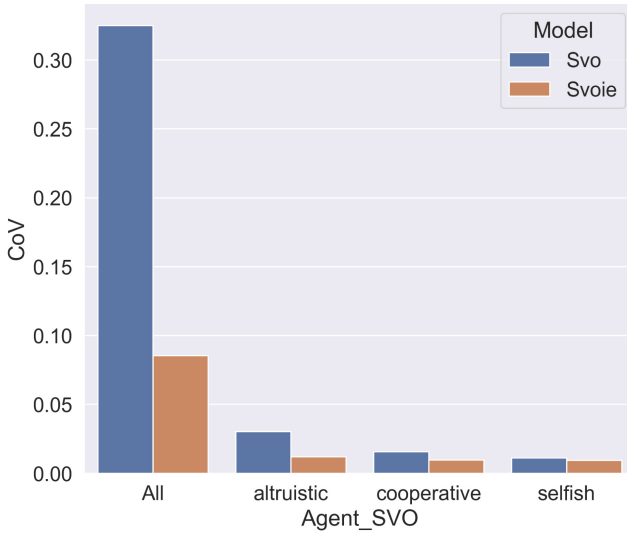
Configuration			<i>Stable-SVO</i>			<i>Svoie</i>		
Society	Sample SVO	Size	Mean score	Std	CoV	Mean score	Std	CoV
altr coop	all	300	16.299	2.435	0.149	15.754	0.807	0.051
	altr	150	13.877	0.206	0.015	14.966	0.169	0.011
	coop	150	18.720	0.221	0.012	16.541	0.177	0.011
altr self	all	300	15.257	7.358	0.481	15.456	1.676	0.108
	altr	150	7.917	0.279	0.035	13.791	0.153	0.011
coop self	self	150	22.597	0.271	0.012	17.121	0.181	0.011
	all	300	16.299	1.758	0.108	15.816	0.690	0.044
	coop	150	14.558	0.226	0.015	15.147	0.157	0.010
	self	150	18.040	0.219	0.012	16.486	0.169	0.010
mixed	all	300	15.863	5.149	0.324	15.664	1.332	0.085
	altr	100	9.269	0.271	0.029	14.016	0.170	0.012
	coop	100	16.527	0.243	0.015	15.731	0.160	0.010
	self	100	21.792	0.267	0.012	17.244	0.179	0.010

Our findings suggest that deviations from stable SVO traits in *Svoie* minimally impact collective welfare. We find that there is no significant difference in collective welfare in the mixed society, *Svoie* ($\mu = 15.664$) and *Stable-SVO* ($\mu = 15.863$), ($p=0.5436$, $d=0.0497$), or for the altr-self society, *Svoie* ($\mu = 15.456$) and *Stable-SVO* ($\mu = 15.257$) ($p=0.6739$, $d=0.034$). However, *Svoie* yields lower collective welfare in both the altr-coop society, *Svoie* ($\mu = 15.754$) and *Stable-SVO* ($\mu = 16.299$) ($p<0.001$, $d=0.305$), and in the the coop-self society, *Svoie* ($\mu=15.816$) and *Stable-SVO* ($\mu = 16.299$) ($p<0.001$, $d=0.371$), albeit with small effect size. Therefore, the societies of *Svoie* agents, which are more likely to seek fair or “inequity averse” actions in response to reaching goals and which are more likely to seek unfair “competitive equity averse” in response to missing goals, were found to perform roughly as well as societies of agents which only act according to their SVO.

Across all societies, we observed that *Svoie* agents significantly reduced welfare inequality compared to *Stable-SVO*, with a large effect size: (altr-coop: *Stable-SVO* $\mu = 16.299$, *Svoie* $\mu = 15.754$, $p<0.001$ $d=84.106$), (altr-self: *Stable-SVO*



(a) Mean welfare.



(b) Inequality.

Fig. 2. Comparison of welfare and inequality in societies of *Stable-SVO* and *Svoie* agents, with an equal number of agents with altruistic, cooperative and selfish SVO traits.

$\mu = 15.257$, *Svoie* $\mu = 15.456$, $p < 0.001$, $d = 39.543$), (self-coop *Stable-SVO* $\mu = 16.299$, *Svoie* $\mu = 15.816$, $p < 0.001$, $d = 73.007$), (*Stable-SVO* $\mu = 15.863$, *Svoie* $\mu = 15.664$, $p < 0.001$, $d = 32.384$). This is illustrated by the distributions of individual welfare (mean score) for samples of agents in the mixed society simulation, shown in Fig. 2a (a). We can see that the distributions of scores for each sample of agents with a particular SVO trait are further apart for the *Stable-SVO* simulations and closer together for the *Svoie* simulations, but the ordering of their performance is unchanged. For example, we observe that altruistic *Svoie* agents perform better than altruistic *Stable-SVO* agents, as they are likely to use unfair strategies in response to being taken advantage of, and selfish *Svoie* agents perform worse than selfish *Stable-SVO*, as they are likely to use fair strategies after taking advantage of others. Further, the width of the distributions of mean score for each SVO is reduced in the *Svoie* simulation, therefore welfare inequality within an individual SVO trait sample is reduced relative to *Stable-SVO* societies as well. This is reflected in the data shown in Table 1 which contains measurements of the mean score achieved by samples of agents with different SVO traits, and the coefficient of variation of the distributions of agent scores within those samples.

5 Limitations, Directions and Conclusions

We now discuss limitations and directions. Firstly, we model societies with heterogeneous SVO by generating populations of agents which can take one of either two or three distinct SVO traits, from altruistic, selfish and cooperative. In human societies, SVO varies continuously between individuals as described by the ring model [31]. Further, we assume an equal distribution of SVO traits in society, whereas in human societies, certain ranges of SVO are more common than others. Buckman et al. [7] implement a more realistic treatment of SVO in agent societies, by sampling agent traits from ranges of the SVO ring model found to be most prevalent in human society using relevant experimental data on SVO prevalence. We did not attempt to simulate realistic human societies, and were focused instead on modelling integral emotions alongside SVO to investigate how this would affect societal welfare and welfare inequality in a society of agents with different SVO policies. The three SVO policies we used in our work give different and non-overlapping distributions in welfare in our baseline simulations, and we therefore considered them to be appropriate for our purposes.

Secondly, we model integral emotion as the variable state E using several simplifying assumptions which prevent any direct comparison with integral emotion in real human behaviour. We only incorporate two integral-emotion-based policies, for positive and negative E respectively. These policies are based on human behaviours which have previously been associated with positive and negative emotions, however they do not follow any explicit model. Further, we assume only one environment trigger, goal-achievement or non-achievement, to be relevant for influencing emotion, whereas there is evidence that other factors influence emotion, e.g. fairness of outcomes [40, 44], which could be utilised in the

CT environment. We also only allow E to vary over five possible states, and the extent to which E changes is constant and chosen arbitrarily, preventing any differences in sensitivity to emotional stimuli between agents. We implemented *Svoie* agents as a coarse-grained model of SVO and integral emotion in agent societies, and did not seek to accurately model human behaviour. In this context, we found that societies of *Svoie* agents had lower welfare inequality compared to baseline *Stable-SVO* agents, and that collective welfare was preserved. These results suggest that by introducing transient changes in decision-making, triggered by task relevant events, agents can adapt their otherwise stable policies depending on the society they operate within.

Agent-based modelling of social decision-making will always require simplifying approximations and assumptions, and cannot accurately capture all aspects of human behaviour, but they are nevertheless useful for studying specific aspects and edge cases [14]. Research on integral emotions (discussed in Sect. 2.2) present the foundational idea behind our contributions—in a social decision-making context, people may make seemingly irrational choices in reaction to recent task outcomes, which may primarily be motivated by strong task-related integral emotions rather than by fixed values, or a rational effort to punish or reward another person due to perceived social inequity. We conduct simulation experiments to investigate the effects at the society level that result from acting according to a simplified and idealised model of this type of behaviour, when compared to acting rationally according to fixed preferences. The limiting and simplifying assumptions of our agent model mean that we cannot predict whether the effects that we observe would extend to real human societies. However, this simulation method offers a useful tool for modelling dynamic behaviour, and better understanding existing models of human behaviour. Understanding the interplay between emotions and social preferences in human decision-making is important for the development of autonomous agents which can understand human social norms, and act in accordance with human moral and ethical principles [1, 28, 38, 49].

There are many factors thought to exert a guiding influence on human behaviour, and models which seek to explain how these factors give rise to a variety of seemingly irrational patterns of behaviour observed in humans, such as predictable deviations from fixed preferences in games and other social contexts. Simulation methods have been applied in related works to study the possible adaptive and socially beneficial effects of different examples of these phenomena. For example, Kampik et al. [25] investigated the role of sympathy in cooperative behaviour, Sylwester et al. [45] have examined antisocial punishment, paying a cost to punish pro-social actors, as a form of social norm enforcement, and Köster et al. [29] demonstrate how the enforcement of arbitrary and inconsequential social norms may improve overall norm compliance in agent societies. Further, there is a rich body of existing work which explores the role of human factors on norm emergence in multi-agent systems [37, 41].

In our chosen simulation task environment, CT, random variations allow differences between the scenarios encountered by agents in each game, however the average performance for any agent is predictable over many time steps. This

work could be extended by investigating how integral emotions influence societal outcomes across multiple task environments, to understand the implications of integral emotion for regulating behaviour in a changing environment. Here, the societal effects of emotions and individual differences could be studied in the context of simulating the emergence and spread of norms in multi-agent systems which benefit survival.

Acknowledgements. DC was supported by the UK Research and Innovation (UKRI) Centre for Doctoral Training in Interactive Artificial Intelligence Award (EP/S022937/1). CH is a Leverhulme Research Fellow (RF-2021-533). NA thanks the University of Bristol for support. DC thanks Phillip Sloan for help with Colored Trails implementation.

References

1. Ajmeri, N., Guo, H., Murukannaiah, P.K., Singh, M.P.: Elessar: ethics in norm-aware agents. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 16–24. IFAAMAS, Auckland (2020). <https://doi.org/10.5555/3398761.3398769>
2. Andrighetto, G., Capraro, V., Guido, A., Szekeley, A.: Cooperation, response time, and social value orientation: a meta-analysis. PsyArXiv (2020). <https://doi.org/10.31234/osf.io/cbakz>
3. Balliet, D., Parks, C., Joireman, J.: Social value orientation and cooperation in social dilemmas: a meta-analysis. *Group Process. & Intergroup Relat.* **12**(4), 533–547 (2009). <https://doi.org/10.1177/1368430209105040>
4. Bogaert, S., Boone, C., Declerck, C.: Social value orientation and cooperation in social dilemmas: a review and conceptual model. *Br. J. Soc. Psychol.* **47**(Pt 3), 453–480 (2008). <https://doi.org/10.1348/014466607X244970>
5. Bono, S.A., van der Schalk, J., Manstead, A.S.R.: The roles of social value orientation and anticipated emotions in intergroup resource allocation decisions. *Front. Psychol.* **11**, 1455 (2020). <https://doi.org/10.3389/fpsyg.2020.01455>
6. Braz, L.F., Sichman, J.S.: Using the Myers-Briggs Type Indicator (MBTI) for modeling multiagent systems. *Revista de Informática Teórica e Aplicada* **29**(1), 42–53 (2022). <https://doi.org/10.22456/2175-2745.110015>
7. Buckman, N., Pierson, A., Schwarting, W., Karaman, S., Rus, D.: Sharing is caring: socially-compliant autonomous intersection negotiation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6136–6143. IEEE, Macao (2019). <https://doi.org/10.1109/IROS40897.2019.8967997>
8. Cheng, K.L., Zuckerman, I., Nau, D., Golbeck, J.: The life game: cognitive strategies for repeated stochastic games. In: Proceedings of the 3rd IEEE Int'l Conference on Privacy, Security, Risk and Trust and 3rd IEEE Int'l Conference on Social Computing, pp. 95–102. IEEE, Boston (2011). <https://doi.org/10.1109/PASSAT/SocialCom.2011.62>
9. Cheriyan, J., Savarimuthu, B.T.R., Cranefield, S.: Norm Violation in Online Communities—A Study of Stack Overflow Comments. In: Aler Tubella, A., Cranefield, S., Frantz, C., Meneguzzi, F., Vasconcelos, W. (eds.) *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*,

- vol. 12298, pp. 20–34. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-72376-7_2
10. Cohen, J.: *Statistical Power Analysis for the Behavioral Sciences*, 2nd edn. Lawrence Erlbaum Associates, Hillsdale, New Jersey (1988)
 11. Criado, N., Argente, E., Noriega, P., Botti, V.: Human-inspired model for norm compliance decision making. *Inf. Sci.* **245**, 218–239 (2013). <https://doi.org/10.1016/j.ins.2013.05.017>
 12. Crosato, L., Wei, C., Ho, E.S.L., Shum, H.P.H.: Human-centric autonomous driving in an AV-pedestrian interactive environment using SVO. In: *Proceedings of the IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*, pp. 1–6. IEEE, Magdeburg (2021). <https://doi.org/10.1109/ICHMS53169.2021.9582640>
 13. Dias, J., Mascarenhas, S., Paiva, A.: FATiMA Modular: towards an agent architecture with a generic appraisal framework. In: Bosse, T., Broekens, J., Dias, J., van der Zwaan, J. (eds.) *Emotion Modeling. Lecture Notes in Computer Science*, vol. 8750, pp. 44–56. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-12973-0_3
 14. Dignum, F.: Should we make predictions based on social simulations? *Int. J. Soc. Res. Methodol.* **26**(2), 193–206 (2023). <https://doi.org/10.1080/13645579.2022.2137925>
 15. Gal, Y., Grosz, B., Kraus, S., Pfeffer, A., Shieber, S.: Agent decision-making in open mixed networks. *Artif. Intell.* **174**(18), 1460–1480 (2010). <https://doi.org/10.1016/j.artint.2010.09.002>
 16. Gal, Y., Grosz, B., Kraus, S., Pfeffer, A., M. Shieber, S.: Colored trails: A formalism for investigating decision-making in strategic environments. In: *Proceedings of the 2005 IJCAI workshop on reasoning, representation, and learning in computer games. 19th International Joint Conference on Artificial Intelligence, IJCAI-05; Conference date: 30–07-2005 Through 05–08-2005*, pp. 25–30 (2005)
 17. Gal, Y., Pfeffer, A.: Modeling reciprocal behavior in human bilateral negotiation. In: *Twenty-Second Conference on Artificial Intelligence (AAAI-07). 22nd National conference on Artificial intelligence, AAAI-07; Conference date: 22–07-2007 Through 26–07-2007*, vol. 22, pp. 815–821. AAAI Press (2007)
 18. Grosz, B.J., Kraus, S.: The influence of social dependencies on decision-making: Initial investigations with a new game. In: *Proceedings of the 3rd International Joint Conference on Multi-agent Systems, AAMAS’04*, pp. 782–789 (2004)
 19. Harsanyi, J.C.: On the rationality postulates underlying the theory of cooperative games. *J. Confl. Resolut.* **5**(2), 179–196 (1961). <https://doi.org/10.1177/002200276100500205>
 20. Huang, X., Wu, W., Qiao, H.: Computational modeling of emotion-motivated decisions for continuous control of mobile robots. *IEEE Trans. Cogn. Dev. Syst.* **13**(1), 31–44 (2021). <https://doi.org/10.1109/TCDS.2019.2963545>
 21. Isen, A.M.: An influence of positive affect on decision making in complex situations: theoretical issues with practical implications. *J. Consum. Psychol.* **11**(2), 75–85 (2001). https://doi.org/10.1207/S15327663JCP1102_01
 22. Joireman, J.A., Shelley, G.P., Teta, P.D., Wilding, J., Michael Kuhlman, D.: Computer Simulation of Social Value Orientation: Vitality, Satisfaction, and Emergent Game Structures. In: Liebrand, W.B.G., Messick, D.M. (eds.) *Frontiers in Social Dilemmas Research*, pp. 289–310. Springer, Berlin, Heidelberg (1996). https://doi.org/10.1007/978-3-642-85261-9_16
 23. de Jong, S., Hennes, D., Tuyls, K., Gal, Y.: Metastrategies in the colored trails game. In: *AAMAS* (2011)

24. Kalia, A.K., Ajmeri, N., Chan, K.S., Cho, J.H., Adah, S., Singh, M.P.: The interplay of emotions and norms in multiagent systems. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 371–377. International Joint Conferences on Artificial Intelligence Organization, Macao (2019). <https://doi.org/10.24963/ijcai.2019/53>
25. Kampik, T., Nieves, J.C., Lindgren, H.: Explaining Sympathetic Actions of Rational Agents. In: Calvaresi, D., Najjar, A., Schumacher, M., Främbling, K. (eds.) Explainable, Transparent Autonomous Agents and Multi-Agent Systems. Lecture Notes in Computer Science, vol. 11763, pp. 59–76. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-30391-4_4
26. Kianpour, M., Øverby, H., Kowalski, S.J., Frantz, C.: Social Preferences in Decision Making Under Cybersecurity Risks and Uncertainties. In: Moallem, A. (ed.) HCI for Cybersecurity, Privacy and Trust. Lecture Notes in Computer Science, vol. 11594, pp. 149–163. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-22351-9_10
27. Kraus, S.: Negotiation and cooperation in multi-agent environments. *Artif. Intell.* **94**(1), 79–97 (1997). [https://doi.org/10.1016/S0004-3702\(97\)00025-8](https://doi.org/10.1016/S0004-3702(97)00025-8)
28. Kuipers, B.: Human-like morality and ethics for robots. In: AAAI Workshop: AI, Ethics, and Society (2016)
29. Köster, R., Hadfield-Menell, D., Everett, R., Weidinger, L., Hadfield, G.K., Leibo, J.Z.: Spurious normativity enhances learning of compliance and enforcement behavior in artificial agents. *Proc. Natl. Acad. Sci.* **119**(3), e2106028118 (2022). <https://doi.org/10.1073/pnas.2106028118>
30. Lerner, J.S., Li, Y., Valdesolo, P., Kassam, K.S.: Emotion and decision making. *Annu. Rev. Psychol.* **66**(1), 799–823 (2015). <https://doi.org/10.1146/annurev-psych-010213-115043>
31. Liebrand, W.B.G., McClintock, C.G.: The ring measure of social values: a computerized procedure for assessing individual differences in information processing and social value orientation. *Eur. J. Pers.* **2**(3), 217–230 (1988). <https://doi.org/10.1002/per.2410020304>
32. Maio, F.: Income inequality measures. *J. Epidemiol. Community Health* **61**, 849–52 (2007). <https://doi.org/10.1136/jech.2006.052969>
33. McClintock, C.G.: Social motivation—a set of propositions. *Behav. Sci.* **17**(5), 438–454 (1972). <https://doi.org/10.1002/bs.3830170505>
34. McClintock, C.G., Allison, S.T.: Social value orientation and helping behavior1. *J. Appl. Soc. Psychol.* **19**(4), 353–362 (1989). <https://doi.org/10.1111/j.1559-1816.1989.tb00060.x>
35. de Melo, C.M., Terada, K.: The interplay of emotion expressions and strategy in promoting cooperation in the iterated prisoner’s dilemma. *Sci. Rep.* **10**(1), 14959 (2020). <https://doi.org/10.1038/s41598-020-71919-6>
36. Moors, A.: Appraisal Theory of Emotion. In: Zeigler-Hill, V., Shackelford, T.K. (eds.) *Encyclopedia of Personality and Individual Differences*, pp. 1–9. Springer International Publishing, Cham (2017). <https://doi.org/10.1007/978-3-319-28099-8-493-1>
37. Morris-Martin, A., De Vos, M., Padget, J.: Norm emergence in multiagent systems: a viewpoint paper. *Auton. Agents Multi-Agent Syst. (JAAMAS)* **33**(6), 706–749 (2019)
38. Murukannaiah, P.K., Ajmeri, N., Jonker, C.M., Singh, M.P.: New foundations of ethical multiagent systems. In: Proceedings of the 19th International Conference

- on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1706–1710. IFAA-MAS, Auckland (2020). <https://doi.org/10.5555/3398761.3398958>. Blue Sky Ideas Track
39. Nardin, L.G., Balke-Visser, T., Ajmeri, N., Kalia, A.K., Sichman, J.S., Singh, M.P.: Classifying sanctions and designing a conceptual sanctioning process model for socio-technical systems. *Knowl. Eng. Rev. (KER)* **31**, 142–166 (2016)
 40. Pillutla, M.M., Murnighan, J.K.: Unfairness, anger, and spite: emotional rejections of ultimatum offers. *Organ. Behav. Hum. Decis. Process.* **68**(3), 208–224 (1996)
 41. Savarimuthu, B.T.R., Cranefield, S.: Norm creation, spreading and emergence: a survey of simulation models of norms in multi-agent systems. *Multiagent Grid Syst.* **7**(1), 21–54 (2011). <https://doi.org/10.3233/MGS-2011-0167>
 42. Schwarting, W., Pierson, A., Alonso-Mora, J., Karaman, S., Rus, D.: Social behavior for autonomous vehicles. *Proc. Natl. Acad. Sci.* **116**(50), 24972–24978 (2019). <https://doi.org/10.1073/pnas.1820676116>
 43. Sloan, P., Ajmeri, N.: Commitment-based negotiation semantics for accountability in multi-agent systems. In: *Proceedings of the 10th International Workshop on Engineering Multi-Agent Systems (EMAS)*, pp. 1–25. Springer, Virtual (2022). <https://doi.org/10.1007/s10472-023-09875-w>
 44. Straub, P.G., Murnighan, J.K.: An experimental investigation of ultimatum games: information, fairness, expectations, and lowest acceptable offers. *J. Econ. Behav. & Organ.* **27**(3), 345–364 (1995)
 45. Sylwester, K., Herrmann, B., Bryson, J.J.: Homo homini lupus? Explaining anti-social punishment. *J. Neurosci. Psychol. Econ.* **6**(3), 167–188 (2013). <https://doi.org/10.1037/npe0000009>
 46. Tzeng, S.T., Ajmeri, N., Singh, M.P.: Fleur: Social Values Orientation for Robust Norm Emergence. In: Ajmeri, N., Martin, A.M., Savarimuthu, B.T.R. (eds.) *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XV. Lecture Notes in Computer Science*, vol. 13549, pp. 185–200. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-20845-4_12
 47. Vilone, D., Realpe-Gómez, J., Andrighetto, G.: Evolutionary advantages of turning points in human cooperative behaviour. *PLoS ONE* **16**(2), e0246278 (2021). <https://doi.org/10.1371/journal.pone.0246278>
 48. Västfjäll, D., Slovic, P., Burns, W.J., Erlandsson, A., Koppel, L., Asutay, E., Tinghög, G.: The arithmetic of emotion: integration of incidental and integral affect in judgments and decisions. *Front. Psychol.* **7** (2016). <https://doi.org/10.3389/fpsyg.2016.00325>
 49. Woodgate, J., Ajmeri, N.: Macro ethics for governing equitable sociotechnical systems. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1824–1828. IFAAMAS, Online (2022). <https://doi.org/10.5555/3535850.3536118>. Blue Sky Ideas Track
 50. Yamagishi, T., Horita, Y., Takagishi, H., Shinada, M., Tanida, S., Cook, K.S.: The private rejection of unfair offers and emotional commitment. *Proc. Natl. Acad. Sci.* **106**(28), 11520–11523 (2009). <https://doi.org/10.1073/pnas.0900636106>
 51. Yamagishi, T., Li, Y., Takagishi, H., Matsumoto, Y., Kiyonari, T.: In Search of Homo economicus. *Psychol. Sci.* **25**(9), 1699–1711 (2014). <https://doi.org/10.1177/0956797614538065>
 52. Zheng, Y., Yang, Z., Jin, C., Qi, Y., Liu, X.: The influence of emotion on fairness-related decision making: a critical review of theories and evidence. *Front. Psychol.* **8**, 1592 (2017). <https://doi.org/10.3389/fpsyg.2017.01592>

Argumentation and Conventions



Towards Ethical Argumentative Persuasive Chatbots

Caren Al Anaissy¹^(✉), Srdjan Vesic², and Nathalie Nevejans³

¹ CRIL Université d'Artois & CNRS, Lens, France
alanaissy@cril.fr

² CRIL CNRS Univ. Artois, Lens, France
vesic@cril.fr

³ Research Center in Law, Ethics and Procedures, Faculty of Law of Douai,
University of Artois, Arras, France
nathalie.nevejans@univ-artois.fr

Abstract. Argumentative persuasive technologies are technologies that use argumentation in order to persuade the persuadee to believe in something or not, which can later lead the persuadee to perform an action or not. The use of such tools opens numerous ethical considerations. In this paper, we survey the literature on persuasion that might be useful for argumentative persuasive chatbots, we cover the existing legal framework and ethical principles and we critically analyze the new proposal for a regulation on artificial intelligence of the European Commission. We also show how to use argumentation to enhance explainability and transparency of the persuasion systems. We propose to show the graphical representation of the arguments used during the persuasion to the user at the end of the dialogue, containing the relations between the arguments (attacks, supports), their origin (source), who uttered them (i.e. the machine or the human participant) and the persuasive methods employed. Our approach has several benefits. Namely, it makes the system more transparent and enhances the human understanding of the system, which is a benefit per se. Furthermore, the fact that the system is transparent increases the trust of the user, which (apart from being one of the goals of AI in general) can increase the chance that the user is persuaded by the system. Finally, the user can give a feedback on the presented arguments (e.g. how much they believe the arguments are ethical), which can be later used to improve the persuasion system.

Keywords: Persuasive chatbots · Ethics of persuasive technology · Computational argumentation

1 Introduction

Persuasion aims to change people's attitudes or behaviours [16]. Persuasive technologies are very powerful tools because of the simulation they can create. They help shaping the perception and interpretation of reality to users by amplifying specific perceptions and reducing others. They also help shaping the users'

actions in reality by encouraging specific forms of actions and discouraging others [32]. Persuasive technologies can also convey social presence when interacting with the user. Many studies have identified persuasive strategies that can be used to influence users and enhance persuasion. Argumentative persuasive chatbots deploy argumentation graphs as their knowledge base. Some of the chatbots even use argumentation semantics for decision-making problems [9]. However, the persuasive acts employed by these chatbots might sometimes be considered morally and ethically unacceptable. Note that the chatbot itself cannot take responsibility for the methods and outcomes of the persuasive acts, because it is not capable of forming its own intentions or making its own choices. Hence, it is not a free moral agent [7].

Persuasion in argumentation has received a large amount of attention during the last years [6, 18–20, 30, 33]. Also, ethics in argumentation has been studied with the focus on the use of normative systems [5, 23, 28].

However, we can observe that the researchers and practitioners who develop argumentative persuasive chatbots are not always aware of the existing advances and the state of the art in the domain of persuasion as well as of the current legal framework and corresponding ethical considerations. This is why the first goal of our paper is to make a bridge between the existing knowledge in persuasion and ethics on one side and the practitioners who develop argumentative persuasive chatbots on the other side.

This is why the **first part of the paper** is devoted to a **survey** of the literature on persuasion related to argumentative persuasive chatbots. We cover the state of the art of persuasion for persuasive chatbots and the corresponding ethical guidelines (Sect. 2). Then, we survey the existing legal framework and its link with ethical principles and critically analyze the new proposal for a regulation on artificial intelligence of the European Commission (Sect. 3). We believe that this overview will be useful for researchers in argumentation who want to deploy an argumentative chatbot for persuasion since it allows them to quickly get knowledge about the most relevant approaches at one place.

The **second part of the paper** is devoted to the question: how can we increase transparency, trust and effectiveness of the argumentative persuasive chatbots? We propose a method that can enhance explainability and transparency of the persuasion systems (Sect. 4). The main idea is to show the graphical representation of the arguments used during the persuasion to the user at the end of the dialogue. This representation contains the relations between the arguments (attacks, supports), their origin (source), who uttered them (i.e. the machine or the human participant) and the persuasive methods employed. This approach has several advantages. Namely, it makes the system more transparent and enhances the human understanding of the system, which is a benefit per se. Furthermore, the fact that the system is transparent increases the trust of the user, which (apart from being one of the goals of AI in general) can increase the chance that the user will be persuaded by the system. Finally, the user can give a feedback on the presented arguments (e.g. how much they believe the arguments are ethical), which constitutes valuable data, which can later be used to better

understand the underpinnings of human reasoning and improve the persuasion system.

The paper is organised as follows. Section 2 presents related work in persuasion that is relevant for argumentative persuasive chatbots and corresponding ethical guidelines. Section 3 surveys the legal and ethical aspects. Section 4 explains the pillars of our idea to use the graphical argumentation-based representation in order to enhance transparency, trust and efficiency of persuasion systems. We then talk about possible future work and conclude.

2 State of the Art in Persuasion

This section provides background material for argumentative persuasive chatbots and the corresponding ethical guidelines.

2.1 Persuasive Strategies

In this subsection, we review some of the persuasive strategies that can be used to enhance persuasion. Fogg was able to identify and propose forty principles that persuasive technologies can use [16]. The forty principles are classified into six categories. The first three categories are the three persuasive roles computing technologies can play. Computers can behave as persuasive tools, persuasive media and persuasive social actors. The three other categories study how computers and web pages can be more persuasive through credibility, mobility and connectivity. Table 1 shows the different principles proposed by Fogg classified into the six categories. We discuss only the principles that seem most relevant for persuasive chatbots. We present briefly the principles related to the role of persuasive technologies as social actors. As Fogg explains, there exist five essential types of social cues that persuasive technologies can convey. First, physical cues can be transmitted through the physical characteristics. Fogg proposes the principle of Attractiveness: “A computing technology that is visually attractive to target users is likely to be more persuasive as well.” Second, persuasive technologies can use psychological cues to persuade. Fogg defines a person’s psychology as the group of emotions, preferences, motivations and personality. Humour and empathy can also be considered as psychological cues. For example, a chatbot possesses empathy when it acknowledges the user’s feelings, shows compassion and supports the user. The expressions like “I understand your feeling” and “I am sorry to hear that” are signs of having empathy towards the user. The principle of Similarity states: “People are more readily persuaded by computing technology products that are similar to themselves in some way.” Third, we have the language. The principle of Praise states that: “By offering praise, via words, images, symbols, or sounds, computing technology can lead users to be more open to persuasion.” Fourth, social dynamics such as giving praise, cooperation and reciprocity can also be used. The principle of Reciprocity proposed by Fogg states that: “People will feel the need to reciprocate when computing technology has done a favour for them.” Finally, adopting a social role can be considered

as a very effective persuasive technique. Specifically, adopting an authority role seems to be very effective for persuasion. The principle of Authority identified by Fogg is the following: “Computing technology that assumes roles of authority will have enhanced powers of persuasion.” We also present some of the principles related to “credibility and computers” which can be considered relevant for persuasive chatbots. Fogg explains that there are two element keys for credibility: trustworthiness and expertise. Trustworthiness represents how much truthful, fair and unbiased a source can be perceived. The principle of Trustworthiness states: “Computing technology that is viewed as trustworthy (truthful, fair and unbiased) will have increased powers of persuasion.” Expertise represents how much perceived knowledge, skill and experience a source can have. The principle of Expertise states: “Computing technology that is viewed as incorporating expertise (knowledge, experience and competence) will have increased powers of persuasion.” Fogg also points out to the fact that computing technology tends to lose credibility easily if commits a significant error. Once the credibility is lost, it may be hard to regain. Therefore, Fogg proposes the principle of (Near) Perfection that states: “Computing technology will be more persuasive if it never (or rarely) commits what users perceive as errors.”

Table 1. The forty persuasion principles proposed by Fogg for persuasive technologies.

Computers as persuasive tools	Computers as persuasive media	Computers as persuasive social actors
Reduction	Cause and effect	Attractiveness
Tunnelling	Virtual rehearsal	Similarity
Tailoring	Virtual rewards	Praise
Suggestion	Simulations in real-world context	Reciprocity
Self-Monitoring		Authority
Surveillance		
Conditioning		
Credibility and computers	Credibility and the world wide web	Mobility and connectivity
Trustworthiness	“Real World Feel”	Kairos
Expertise	Easy verifiability	Convenience
Presumed credibility	Fulfilment	Mobile simplicity
Surface credibility	Easy-of-use	Mobile loyalty
Reputed credibility	Personalization	Mobile marriage
Earned credibility	Responsiveness	Information quality
(Near) Perfection		Social facilitation
		Social comparison
		Normative influence
		Social learning
		Competition
		Cooperation
		Recognition

Cialdini was able to identify six influence principles: Reciprocity, Commitment and Consistency, Social Proof, Liking, Authority, Scarcity [11]. Among these principles, we explain briefly the ones that were not explained before. The

Commitment and Consistency principle states that humans tend to commit to their opinions, values and choices. The Social Proof principle states that people tend to imitate other people's ideas and actions. For example, if a person is considering buying a product online but they are not sure whether the product is good or convenient for them, they can check the product's reviews section. If most of the reviews are positive, the user tends to feel more confident in their decision in purchasing the product. The Liking principle is close to the Similarity principle proposed by Fogg, it states that people like others who are similar to them. The Scarcity principle states that people tend to value opportunities or things that become less available, hence scarce. Oinas-Kukkonen and Harjumaa proposed and developed a Persuasive Systems Design model [26] where twenty-eight persuasive strategies were listed for the design of persuasive technology. These principles were divided into four categories: primary task, dialogue, system credibility, and social support.

Wang et al. were able to identify ten persuasion strategies that are divided into two types, the persuasive appeal and the persuasive inquiry type. The persuasive appeal type consists of trying to appeal to the persuadee's psychology. The persuasive inquiry type consists of asking the persuadee personal questions to facilitate the persuasion [35]. The persuasive appeal strategies identified in this work are the following: The Logical appeal strategy consists of using evidence and reasons to convince the persuadee of the persuasion goal. The Emotional appeal strategy consists of evoking the persuadee's positive and/or negative emotions. The Credibility appeal strategy consists of citing information from objective sources in order to gain the persuadee's trust. The Foot-in-the-door strategy consists of asking the persuadee small requests first, then asking larger ones. The Personal story strategy consists of telling the persuadee stories about other people who were persuaded by the persuasion goal, focusing on the positive results of such persuasion. The Donation Information strategy consists of giving the persuadee information about the action or idea the persuader wants to convince them with. The persuasive inquiry strategies are the following: The Source-related inquiry strategy consists of asking the persuadee whether they are aware or not of the action or idea the persuader wants to persuade them with. The task-related inquiry strategy consists of asking the persuadee their own opinions and expectation concerning the persuasion goal. Finally the personal-inquiry strategy consists of asking the persuadee about their own personal experiences related to this persuasion goal.

2.2 Personalization in Persuasion

In this subsection, we briefly define personalized persuasion and we briefly review two works done in this field. Personalization plays an important role in enhancing persuasion. Personalized persuasion consists of using the user's personal information and background to enhance the outcome of persuasion [12, 22, 24, 27]. Apart from using the user's psychological cues to persuade them, personalization can appear in the form of trying to adapt the methods used during the

process of persuasion based on the user's psychological profile and/ or personal information. The goal of such adaptation is also the enhancement of the persuasion outcome. Kaptein et al. studied the effects of involving users in choosing a specific influence strategy for persuasion, disclosing the usage of such strategies and the use of multiple strategies simultaneously on user compliance to persuasive attempts [21]. The authors consider these results as guidelines for designing adaptive persuasion systems. Adaptive persuasion systems are persuasion systems that use different influence strategies based on the users' profiles in order to increase their influence on users. The authors proved that letting the user decide which strategy to adopt in order to persuade them is more effective than predicting the preference for a specific strategy based on behaviour or personality measures. The reason behind this is that if the user chooses which influence strategy to adopt, they will commit to choice that they have made. The authors have also proved that using a single preferred strategy is more effective than using the preferred strategy simultaneously with a non preferred one. Also, using the two strategies simultaneously was more effective than using the single non preferred strategy alone. The two persuasive strategies used in this work are the authority and the consensus strategies. Wang et al. studied how the variations of the persuasion strategies upon the user's psychological background, affect the persuasion outcome [35]. The results of the work done by Wang et al. [35] and Shi et al. [31] show that personalizing the persuasion strategies yields better persuasion outcomes.

2.3 Persuasive Chatbots

In this subsection, we briefly review four persuasive chatbots that were developed and designed. The first three chatbots use the Credibility Appeal (Trustworthiness and Expertise) strategy because all the arguments presented by the chatbots come from objective sources (scientific sources, governmental websites and experts) therefore the information presented by the chatbots are unbiased, fair and truthful. As for the fourth chatbot, it uses different persuasive strategies during the dialogue.

Altay et al. studied whether they can change people's opposite opinion regarding genetically modified food and genetically modified organisms by providing rebuttals to the counterarguments held against genetically modified organisms [2]. They have defined four conditions: the Control Condition consists of defining genetically modified organisms. The Consensus Condition informs about the important existence of the scientific consensus regarding the genetically modified organisms' benefits. The Counterarguments Condition presents first counterarguments against genetically modified organisms then rebuttals to these counterarguments, then rebuttals to the previous ones and so on. The user scrolls to check all the arguments which were presented in a clear dialogue structure. Finally, the Chatbot Condition consists of presenting all the counterarguments against genetically modified organisms to the user, the user can click on any counterargument, the rebuttal of the counterargument selected appears

progressively. The user has the option to come back to the initial counterarguments and click on another one. The four conditions were compared against each other. The Counterarguments Condition was found more persuasive than the Chatbot condition in the sense that spending more time reading when all counterarguments are available leads to more positive attitude changes than selecting only the most relevant counterarguments.

Hadoux and Hunter showed how preferences over types of concern can be used to enhance the persuasion in persuasive dialogues [17]. The notion of concern is defined as an issue raised or addressed by an argument. Among the empirical studies conducted, a group of participants chatted with two types of chatbots called the baseline chatbot and the preference-based chatbot. The structure of the dialogue is the same for both chatbots, the first argument is the one proposed by the system and called the persuasion goal, then a menu of counterarguments is proposed by the system to the user. The user can select a set of counterarguments; this set is called a menu move. Then the system can counter the selected arguments by a set of arguments called the posit move. The dialogue continues in the same manner until there is no argument attacking any argument of the last move or if the user ends the dialogue by a null argument. A null argument means that the user does not choose any of the counterarguments presented by the system. For the baseline chatbot, for each counterargument selected by the user, the chatbot selects a rebuttal among the arguments that attack the counterargument randomly. For the preference-based system, a set of the user's preferences over concerns is available, for each counterargument selected by the user, the chatbot selects the rebuttal with which the most preferred type of concern is associated. The results show that using preferences over concerns enhances the persuasiveness of the dialogue.

Chalaguine and Hunter designed a persuasive chatbot to persuade users to take the Covid-19 vaccine [10]. The authors use the same notion of concern as in [17]. The chatbot first presents the persuasion goal, then the user provides a counterargument manually. The chatbot predicts the user's concern raised in the counterargument and replies with the first not yet used rebuttal that addresses the same concern. In case the chatbot could not identify the concern of the user i.e. the prediction is less than 40% in confidence, it replies with one among three default rebuttals. There exist only three default rebuttals. The dialogue ends when the chatbot cannot identify a concern and all of the three default rebuttals were already used. The authors have shown that this interactive chatbot is more persuasive than a static web page in which the users read the ten most common rebuttals used by the chatbot.

Shi et al. developed a persuasive chatbot that understands the user's input based on neural network models [31] and replies from the human responses that were collected from the previous work [35]. The authors conducted experiments to study the effect of disclosing the chatbot's identity (bot or human) and the effects of using different type of inquiries (personal and/or non personal inquiries) on the persuasiveness of the dialogue. The chatbot salutes the user first, then the chatbot asks the user some questions based on the type of persuasive inquiry

the user was assigned to. After finishing from the persuasive inquiry module, the chatbot moves on to the persuasive appeal module where the chatbot dialogues with the user. At each step, the chatbot uses a different persuasive strategy and asks the user if they want to donate. If the user accepts to donate, that means that the chatbot succeeded in persuading the user to donate. If the user does not accept to donate, the chatbot uses another not yet used persuasive strategy to try to persuade the user to donate. The dialogue ends if the user agrees to donate or if the ten persuasive strategies were all used. Results showed that whether the chatbot was really human or not, it is the perceived identity of the chatbot by the user that matters. The persuasiveness is better when people think they are talking to human.

2.4 Argumentation Theory in Persuasive Dialogues

In this subsection, we review the main elements of a persuasive dialogue system introduced by Prakken [29]. Argumentation-based dialogues can be classified into six types based on their goal [4]. We have persuasion, negotiation, information seeking, deliberation, inquiry and quarrel. When it comes to argumentation-based dialogues, there are multiple rules to take into account. The communication language consists of the utterances the participants can make, the protocol consists of the conditions under which the participants can make the utterances, it also determines when the dialogue ends. According to Prakken [29], the main elements of a persuasive dialogue systems are the following:

- A dialogue goal which, in persuasion dialogues, is the resolution of a conflict of point of views between the participants.
- A topic language
- A logic for the topic language used, which can be monotonic or non-monotonic, it can be used to manage the dialogical consistency of participants.
- A communication language: As defined before, communication language consists of the allowed utterances to make. The most important ones are : claim ϕ , why ϕ , concede ϕ , retract ϕ , question ϕ and ϕ since S . As explained by Prakken [29], Claim ϕ means that the speaker asserts that ϕ is the case. Why ϕ means that the speaker challenges that ϕ is the case and asks for reasons why it would be the case. Concede ϕ means that the speaker admits that ϕ is the case. Retract ϕ means that the speaker declares that they are not committed (any more) to ϕ . Question ϕ means that the speaker asks another participant's opinion on whether ϕ is the case. Finally, ϕ since S means that the speaker provides reasons why ϕ is the case.
- A protocol: The protocol specifies the allowed moves at each step of the dialogue. The protocol specifies the dialogue's structure. We have unique-reply vs multi-reply, unique-move vs multi-move and immediate-reply vs non-immediate-reply, deterministic and fully deterministic vs non deterministic protocols.

- A set of participants with roles, internal beliefs and commitments. Usually the roles in a persuasion dialogue are proponent, opponent and neutral toward a well specified topic. Commitments are very important because they determine the end of the dialogue and its outcomes, and they can be used to oblige the participant to be dialogically consistent. Commitments are usually determined by claim ϕ , concede ϕ and retract ϕ .
- Effect rules: The effect rules determine the effects of the speech acts on the participants' commitments.
- Outcome rules: The outcome rules define the outcomes of a dialogue which are in a persuasive dialogue the winners and the losers of the dialogue.

2.5 Ethics of Persuasive Chatbots

In this subsection, we cover the ethical guidelines proposed in the literature for the design of persuasive technologies and we try to orient some of them towards the design of persuasive chatbots precisely. Berdichevsky and Neuenschwander explain that when it comes to persuasion, both persuader and persuadee take full moral responsibility for the outcome [7]. In order to evaluate the ethics of persuasion itself, one should evaluate the persuader's motivations, the methods they employed and the outcome of persuasion.

Although persuasive technology and persuasive people have same motivations and use similar methods and strategies, persuasive technology has more persuasive potential because of the simulations they can embed leading to more realism. The difference between persuasion through technology and through person-to-person interactions relies in the methods used for persuasion and also probably the outcome. The ethics of persuasion seems to be insufficient to guide the design and implementation of persuasive technology. The authors wanted to reconsider the ethical guidelines for traditional persuasion methods when being applied by technology and not humans, and for the outcome i.e. the persuasion goal.

Therefore, Berdichevsky and Neuenschwander proposed a set of eight ethical principles and guidelines for the design and implementation of persuasive technology, with the consideration that the designers should be only responsible for intended and unintended reasonably predictable outcomes [7]. The first two principles state that the intended outcome of any persuasive technology and the motivations behind it should never be considered unethical if the persuasion was done without the technology or if the outcome happened independently of the persuasion. The third principle states that the designers of such technologies should take responsibility for all reasonably predictable outcomes of their use. The "Dual Privacy" principles state that the creators of persuasive technology should respect users' privacy when it comes to accessing their online personal information and sharing it with a third party. The "Disclosure" principle states that the designers must be transparent and clear about the motivations, methods and intended outcomes of such technology. The "Accuracy" principle states that the persuasive technology should always be honest and credible. Finally, the "Golden" principle states that the designers of persuasive technology should

never use a persuasion goal that they themselves are not consent of being persuaded by it.

Verbeek [32] emphasised on the importance of integrating the ethical framework proposed by Berdichevsky and Neuenschwander [7] with the concept of “technological mediation” in order to better understand and predict unintended outcomes, and take these outcomes into consideration when designing persuasive technology. As mentioned before, technology tends to shape human perception and interpretation of reality by amplifying some perceptions while reducing others. It also shapes human actions in reality by encouraging specific forms of actions while discouraging others. Persuasive technology mediates these effects between users and their environments, so when technology is used the way their designers intended, it is possible to have unintended and unexpected outcomes, Verbeek proposed to focus on all the mediation effects by doing a moral reflection along deontological and utilitarian lines. The deontological point of view means respecting the moral principles while the utilitarian point of view means balancing between the desirability for something and its costs for all the people involved. This moral reflection will take into consideration the intended persuasions, the methods of persuasion used with the emerging forms of mediation, and the outcomes of the mediation which are the consequences of the persuasive and mediating role of the technology. The ethical guidelines that the authors proposed are the following:

- The intended persuasions of the technology-in-design must cause no harm for the people using persuasive technologies and those affected by them being used, these intended persuasions must benefit these people and be fair (justice) to them.
- The methods of persuasion and forms of mediation must be disclosed (respect for autonomy), cause no harm in terms of privacy and be fair to all people.
- As for the outcomes of mediation, the designers must do a moral imagination of all the possible mediating roles of technology in human actions and experiences and then assess these mediations along the deontological and utilitarian lines.

Fogg proposed to apply a stakeholder analysis to identify all people affected by a persuasive technology, and what each stakeholder has to gain or lose [16].

- List all of the stakeholders.
- List what each stakeholder has to gain.
- List what each stakeholder has to lose.
- Evaluate which stakeholder has the most to gain.
- Evaluate which stakeholder has the most to lose.
- Determine ethics by examining gains and losses in terms of values.
- Acknowledge the values and assumptions you bring to your analysis.

Note that values differ from a culture to another. Hence, creators of persuasive technology must be careful about the culture in which they are embedding this technology, because with every different culture, comes different ethical issues.

We list below a set of guidelines for the design of persuasive chatbots inspired by the guidelines proposed by Fogg [16] for the design of persuasive technology.

- Users of persuasive chatbots should not be distracted by the number of questions or the difficulty of arguments, because this can stand in the way of their focus on the content in the chatbot. The chat must not be complicated or very lengthy.
- Creators of persuasive chatbots should not consider that the user has experience in the domain of the goal with which we want to persuade the user.
- Creators of persuasive chatbots should not include in the chat any links to download an application or something else.
- The user must be able to stop at anytime they want, or ask for clarification. The creators should be also careful about the cases where the user must have the ability to ask for a human intervention.
- Not only the creators of persuasive chatbots take responsibility when it comes to errors and damage to the user, but also the company that bought this technology, distributed and promoted it. We may have different companies through time, they all can be responsible.
- Manipulation can happen when the chatbot expresses negative or positive emotions towards the user, presents arguments that appeal to the user’s positive/negative emotions, tells lies or false information, tells incomplete information, chats with children or mentally disabled people, presents threatening information or punishment. Negative emotions could be fear, angry, deception, impatience. Positive emotions could be celebration, rewarding, encouraging. Designers of persuasive chatbots should avoid manipulation at all costs. It is preferred that the chatbot does not express emotions at all or expresses the minimum and only for good cause.
- The chatbot should not be very sophisticated in a way that confuses the user whether the chatbot is a human or robot. The user must know that they are chatting with a robot.
- Designers of persuasive chatbots must test and supervise these chatbots when used to observe if there are any unintended outcomes that were not recognised before, or happen to a small number of people. This is how they should deal with unintended and unpredictable outcomes. They should also keep track of the conversations between the chatbot and the users, with the user’s knowledge.
- Persuasive chatbots should not provide offers, promotions, advertisements or branding. They should be designed exclusively for persuasion.

Creators of persuasive chatbots are also invited to consult the guidelines for developers of conversational AI proposed by Microsoft [13].

3 Persuasive Chatbots Between Ethics and Law

In this section, we present and discuss the legal issues that impact the design and implementation of chatbots, specifically persuasive chatbots. On April 21, 2021 the European Commission has published a proposal for a regulation on artificial intelligence [15], called the AI Act, which is currently under discussion and will soon be adopted [1]. The AI Act proposes a gradation of legal constraints

according to the risks presented by the AI system. These risks are those relating to health, safety, fundamental rights and environment. AI systems are therefore classified into the following categories:

- “Unacceptable Risk” (social scoring, subliminal techniques, biometric categorisations, “real-time” remote biometric identification systems in publicly accessible spaces, etc.), the use of which is banned, sometimes with some exceptions.
- “High Risk” (Annex III cites biometrics, management of critical infrastructure, educational and vocational training, employment, workers management and access to self-employment tools, access to essential public and private services, etc.), the use of which must follow strict obligations and requirements so that the AI system can be placed on the market in the European Union.
- “Low Risk” (AI systems intended to interact with people (i.e. Chatbots), deep fakes, emotion recognition systems, etc.), the use of which requires compliance with an obligation of transparency (Article 52).
- “Minimal Risk” (e.g. Video games and spam filters based on AI), for which the AI Act does not impose any specific obligation.

The AI Act is one of the first texts in the world that will impose legal obligations for chatbots, when the text currently under discussion is voted on. The 2021 Proposal contained very few obligations regarding chatbots. Indeed, users brought to interact with a chatbot only needed to know that they were discussing with a machine in order to be able to choose whether to continue the discussion or not. However, the latest versions are much more precise [1]. On the one hand, the amendments to the Proposal explain how to provide the information. Article 52.1 now states that: “Providers shall ensure that AI systems intended to interact with natural persons are designed and developed in such a way that the AI system, the provider itself or the user informs the natural person exposed to an AI system that they are interacting with an AI system in a timely, clear and intelligible manner, unless this is obvious from the circumstances and the context of use.” The information must therefore be provided either by the provider itself or by the chatbot, or by the professional user. This information must also be provided in a way that is clear, intelligible and at the most late at the time of the first interaction (Article 52.3, b) so that the person can choose not to use it, unless the fact that they are interacting with a chatbot is obvious to the taking into account the circumstances and the context of use. In addition, this text now also requires that the provider indicates which functions are AI enabled, if there is human oversight, and who is responsible for the decision-making process, as well as the possibilities to object against the application and to seek judicial redress against decisions taken by or harm caused by AI systems. However, Article 52.1 relates to chatbots in general, but not specifically to persuasive chatbots. Therefore, this information obligation may not be sufficient to protect users in this context.

In order to guarantee more complete transparency for the users of a persuasive chatbot, it seems important that the providers also inform them of the persuasion strategy underlying their system. For example, it is relevant that the

users of an authority-based persuasion chatbot are informed of the designers' goals. The AI Act does not directly address this concern. This therefore means that if this text remains as it is on this subject, people interacting with a persuasive chatbot could be incompletely protected. The ethical approach is a response to this concern, because it makes it possible to reinforce the law in order to do what is well beyond what is only legal. European policies relating to the ethics of AI also adopt this vision [25].

We therefore argue that simply informing the users that they are interacting with a chatbot is insufficient for a persuasive chatbot. It is therefore crucial to provide them with additional ethical information:

- Users must be informed about the nature of the persuasion system used.
- Users should be made aware of the potential effects of the persuasive system used, particularly if the persuasive effect could be enhanced.

Can the designers of a persuasive chatbot be likely to infringe the rights and freedoms of users? We believe that if the chatbot adapts its method of persuasion according to the gender, racial origins, or religious beliefs of users, it might risk to behave in a discriminatory way. While the previous versions of the AI Act neglected these risks for rights and freedoms, the latest amendments reveal the desire to integrate the ethical principles of AI which had only been mentioned in the Guidelines or other non-binding texts of the European Union [25]. Thus, the new Article 4, a) concerns the “General principles applicable to all AI systems” which must be respected by all operators, including the provider and the professional user (i.e. deployer for the latest versions of the AI Act), whose AI system falls within the scope of the AI Act. This is indeed the case of providers of persuasive chatbots which, as low-risk AI systems, must respect six new fundamental principles which are:

- Human agency and oversight
- Technical robustness and safety
- Privacy and data governance
- Transparency
- Diversity, non-discrimination and fairness
- Social and environmental well-being

If these principles end up being definitively voted on, we can therefore think that a chatbot will have to respect the principles of non-discrimination and equity. However, it should be borne in mind that the goals of the designers of the persuasive chatbot may only be discriminatory in appearance. Indeed, it is quite conceivable that designers adapt their method of persuasion, for example, to the age or level of education of the user, which would then be a simple way for designers to be better understood or for the chatbot to be more easily used by specific users.

If the chatbot is gendered or humanised, it is still likely to infringe the rights and freedoms of users in two cases. In the first case, certain human aspects can reinforce the persuasive effect. For example, the chatbot can appear in the

form of a gentle face of a grandmother who softens the users. Designers must be aware of this and minimise these characteristics. In the second case, gender or certain human aspects can be potentially sexist or discriminatory depending on the uses that are made of them and the goals that designers pursue. One of the solutions could be to minimise the human characters to avoid the problems of sexist or discriminatory biases. However, it is possible for a non-gendered persuasive chatbot without human characteristics to be less persuasive for the purposes pursued by the designer. In this case, a compromise must be made between the values to be respected and the goals to be achieved by the chatbot.

Designers of a persuasive chatbot can still infringe people’s rights and freedoms if the chatbot is designed to manipulate users, for example by leading them in a certain direction without their knowledge or saying things that are false or truncated. The latest version of the AI Act, which now prohibits the use of deliberately manipulative or deceptive techniques in Article 5.1, a) as an unacceptable risk AI system, seems to come closer to the objectives pursued by a persuasive chatbot. In reality, the prohibition concerns cases where the manipulative technique would seriously harm the person: “the placing on the market, putting into service or use of an AI system that deploys subliminal techniques beyond a person’s consciousness or purposefully manipulative or deceptive techniques, with the objective to or the effect of materially distorting a person’s or a group of persons’ behaviour by appreciably impairing the person’s ability to make an informed decision, thereby causing the person to take a decision that that person would not have otherwise taken in a manner that causes or is likely to cause that person, another person or group of persons significant harm.” Therefore, the use of a persuasive chatbot, except for the purpose of achieving the extreme results referred to in Article 5.1, a), is not prohibited under the current state of the AI Act. The question is nonetheless also very delicate from an ethical point of view, since the motivations of designers can vary considerably. However, we believe that the majority of the designers have good intentions and want to use the chatbot for users’ own good. As an example, take the the chatbots that try to persuade users to practice a sporting activity or to limit the consumption of alcohol or sugar.

4 Ethical Design via Explainability

There is an increasing interest in Explainable AI over the last few years in order to tackle the ethical challenges that arise from the use of AI-based technologies. Vilone and Longo list the existing definitions in the literature of the notions related to the concept of explainability [34]. We believe that we can respect the ethical guidelines for persuasive chatbots [7] by using argumentation for the explainability of persuasive chatbots. In this work, we use mostly two notions of explainability: understainability and correctability. Understainability means the capacity of a method for explainability to make a model understandable while correctability means the capacity of a method for explainability to allow end-users make technical adjustments to an underlying model [34]. Our method

consists of showing an argumentation graph to the user after the dialogue: that graph highlights the persuasive methods and the sources of information used by the chatbot, and the degrees to which the user finds the chatbot’s arguments ethically acceptable. Before explaining our method, we briefly present Dung’s abstract argumentation framework.

4.1 Dung’s Abstract Argumentation Framework

In abstract argumentation [3,8,14], arguments are considered defeasible entities where all information related to these arguments are abstracted away except for the relations of attacks between them. Dung’s argumentation framework [14] is one of the attempts used to formalise reasoning i.e. to represent systems of arguments and their relations, determine which arguments are acceptable.

Definition 1. *An abstract argumentation framework AF is a pair $\langle Ar, att \rangle$ where Ar is a finite and non-empty set of arguments and $att \subseteq Ar \times Ar$ is an attack relation (\rightarrow).*

Figure 1 shows an example of an abstract argumentation framework with arguments represented by nodes, and relations of attacks among them.

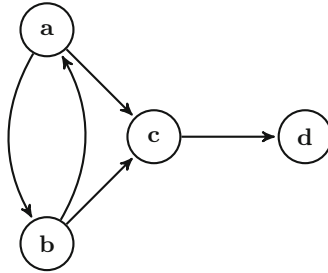


Fig. 1. Example of an abstract argumentation framework

4.2 An Argumentation-Based Approach Towards Explainability

Our method consists of labelling each argument presented by the chatbot by a persuasion strategy (if it exists) and by the source of the information presented in the argument. If the chatbot uses natural language processing to generate the arguments presented to the user, this process is called post-labelling because the chatbot labels the arguments after they were presented to the user. In the other case where the chatbot has already a knowledge base i.e. well defined arguments in its system, then each argument must be pre-labelled. For both cases, the user chats with the persuasive chatbot.

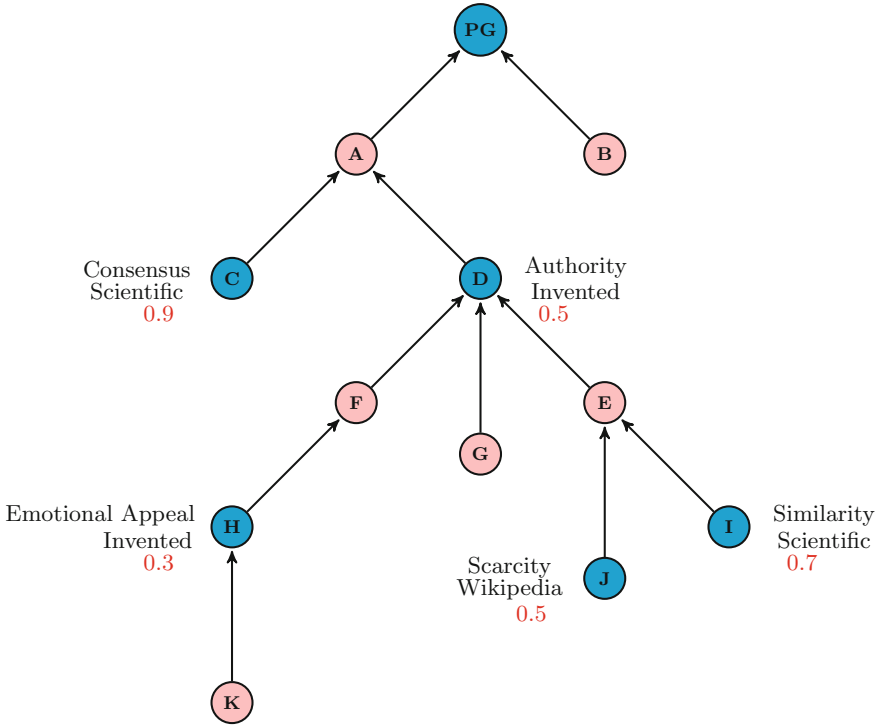


Fig. 2. Example of an abstract argumentation graph representing the dialogue between the user and the chatbot. “PG” stands for persuasion goal. The nodes in blue represent the chatbot’s arguments while the nodes in pink represent the user’s arguments. The first row of labels represents the persuasive strategies used by the chatbot. The second row represents the source of the information presented by the chatbot. The third row represents the degrees assigned by the user

When the dialogue ends, the chatbot shows an argumentation graph that consists of all the arguments that were presented during the dialogue by both sides, with the relations (attacks) between them. By showing this argumentation graph, the chatbot shows the persuasion strategies that were used during the process of persuasion to the user. Hence, the chatbot discloses all the methods employed in the dialogue. It also shows the source of the information it provided in each argument. The information can be extracted from scientific sources, crowd-sourcing, online forums, governmental websites, personal communication with experts, etc. Also, it can be generated by the chatbot i.e. invented.

Adopting this method allows the user to assess the accuracy of the information that was given by the chatbot and to possibly detect if the chatbot lied or stated false information. By implementing this method, we answer to the question : How did the chatbot try to persuade the user?

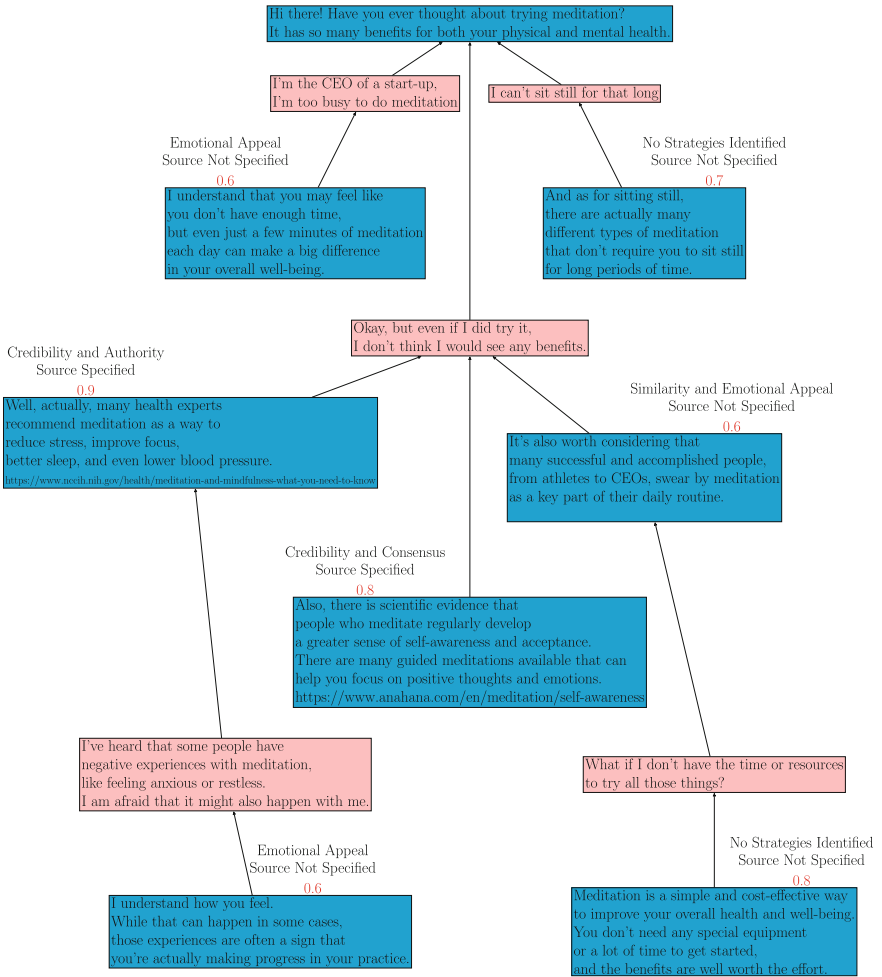


Fig. 3. Example of an abstract argumentation graph representing the dialogue between the user and the chatbot. The chatbot tries to persuade the user to do meditation. The nodes in blue represent the chatbot's arguments while the nodes in pink represent the user's arguments

We also allow the user to input for each argument presented by the chatbot, a degree that ranges between 0 and 1. Each degree associated with a specified argument must represent the user's belief in the argument being (somehow) ethical. We let this assignment be spontaneous and intuitive in order to be able to represent the user's actual beliefs and preferences over what is considered ethical and what is not. Assigning a degree to each argument presented by the chatbot will let us know the set of preferences of the user over the persuasive strategies and over the sources of information provided. Hence, we can build a

recommendation system which predicts the set of preferences of a user based on their personal information and/or personality measures. This way, the chatbot can be considered ethically adaptive. Also, this can help the designers of the chatbot to eliminate from the chatbot’s knowledge base the persuasive strategies or even the arguments that were assigned a very low degree of being ethical. Figures 2 and 3 show examples of abstract argumentation graphs presented to the user after the dialogue with the chatbot.

5 Conclusion

In this paper, we studied ethical argumentative persuasive chatbots. In the first part of the paper, we reviewed the state of the art of persuasion for argumentative persuasive chatbots and the ethical guidelines for the design of such systems. First, we provided background material for the persuasive strategies that can be used by persuasive chatbots to enhance the persuasion. Then we discussed how personalization used in persuasion can be helpful to improve the persuasive effect. We also reviewed four argumentative persuasive chatbots where we focused on the dialogue structure, and we briefly studied argumentation in persuasive dialogues. Finally, we presented the state of the art of ethics in persuasive technologies and we made a list of the ethical guidelines that designers of persuasive chatbots are invited to respect. We also discussed the legal constraints related to design and implementation of persuasive chatbots and we showed how ethics could complement the legal framework in order to better respect the user’s freedoms and rights.

In the second part of the paper, we proposed to use argumentation to display the persuasive strategies employed by the chatbot and the source of the information presented by the chatbot to the user. This way, the chatbot discloses the persuasive methods it used and provides to the user more transparency by providing for them the source of the information presented in the arguments. We also proposed to assess how much ethical each argument presented by the chatbot is, by letting the user input how much they believe each argument is considered ethical. This way, to eliminate the arguments that have very low degrees in the next dialogue, and we can ethically adapt the arguments presented by the chatbot to the user’s preferences.

Acknowledgements. This work benefited from the support of the AI Chair project Responsible AI (ANR-19-CHIA-0008) (<https://ia-responsable.eu/>) and the project AGGREEY (ANR-22-CE23-0005), both from the French National Research Agency (ANR).

References

1. Proposal for a Regulation of the European Parliament and of the Council laying down Harmonised Rules on Artificial Intelligence and amending certain Union Legislative Acts (AI Act), April 21, 2021. See the successive evolutions of the Proposal: General approach of the European Parliament and of the Council, November 11, 2022, Draft Compromise Amendments on the Draft report of the European Parliament and of the Council, May 16, 2023, and Draft European Parliament Legislative Resolution, June 14, 2023
2. Altay, S., Schwartz, M., Hacquin, A.S., Allard, A., Blancke, S., Mercier, H.: Scaling up interactive argumentation by providing counterarguments with a chatbot. *Nat. Hum. Behav.* 1–14 (2022)
3. Amgoud, L., Doder, D., Vesic, S.: Evaluation of argument strength in attack graphs: foundations and semantics. *Artif. Intell.* **302**, 103607 (2022)
4. Baroni, P., Gabbay, D., Giacomin, M., van der Torre, L.: Handbook of Formal Argumentation. College Publications. <https://books.google.be/books?id=-OnTswEACAAJ> (2018)
5. Bench-Capon, T., Modgil, S.: Norms and value based reasoning: justifying compliance and violation. *Artif. Intell. Law* **25**(1), 29–64 (2017)
6. Bench-Capon, T.J.: Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.* **13**(3), 429–448 (2003)
7. Berdichevsky, D., Neuenschwander, E.: Toward an ethics of persuasive technology. *Commun. ACM* **42**(5), 51–58 (1999)
8. Besnard, P., Hunter, A.: Elements of Argumentation. MIT Press (2008)
9. Bistarelli, S., Taticchi, C., Santini, F.: A chatbot extended with argumentation. In: *AI³@ AI* IA* (2021)
10. Chalaguine, L., Hunter, A.: Addressing popular concerns regarding Covid-19 vaccination with natural language argumentation dialogues. In: *European Conference on Symbolic and Quantitative Approaches with Uncertainty*, pp. 59–73. Springer (2021)
11. Cialdini, R.: Influence: The Psychology of Persuasion. Business Library. <https://books.google.be/books?id=mJidPwAACAAJ> (1984)
12. Ciocarlan, A., Masthoff, J., Oren, N.: Actual persuasiveness: impact of personality, age and gender on message type susceptibility. In: *International Conference on Persuasive Technology*, pp. 283–294. Springer (2019)
13. Corporation, M.: Responsible bots: 10 guidelines for developers of conversational AI. <https://www.microsoft.com/en-us/research/publication/responsible-bots/> (2018). [Online; accessed 4 Nov 2018]
14. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.* **77**(2), 321–357 (1995)
15. Ebers, M., Hoch, V.R., Rosenkranz, F., Ruschemeier, H., Steinrötter, B.: The European commission’s proposal for an artificial intelligence act—a critical assessment by members of the robotics and AI law society (rails). *J* **4**(4), 589–603 (2021)
16. Fogg, B.J.: Persuasive technology: using computers to change what we think and do. *Ubiquity* **2002**(December), 2 (2002)
17. Hadoux, E., Hunter, A.: Comfort or safety? gathering and using the concerns of a participant for better persuasion. *Argum. & Comput.* **10**(2), 113–147 (2019)

18. Hadoux, E., Hunter, A., Corrége, J.B.: Strategic dialogical argumentation using multi-criteria decision making with application to epistemic and emotional aspects of arguments. In: International Symposium on Foundations of Information and Knowledge Systems, pp. 207–224. Springer (2018)
19. Hunter, A.: Towards a framework for computational persuasion with applications in behaviour change. *Argum. & Comput.* **9**(1), 15–40 (2018)
20. Hunter, A., Polberg, S.: Empirical methods for modelling Persuadees in dialogical argumentation. In: 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 382–389. IEEE (2017)
21. Kaptein, M., Duplinsky, S., Markopoulos, P.: Means based adaptive persuasive systems. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp. 335–344 (2011)
22. Kaptein, M., Markopoulos, P., De Ruyter, B., Aarts, E.: Personalizing persuasive technologies: explicit and implicit personalization using persuasion profiles. *Int. J. Hum. Comput. Stud.* **77**, 38–51 (2015)
23. Liao, B., Slavkovik, M., van der Torre, L.: Building jiminy cricket: an architecture for moral agreements among stakeholders. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp. 147–153 (2019)
24. Liao, M., Sundar, S.S.: How should AI systems talk to users when collecting their personal information? effects of role framing and self-referencing on human-AI interaction. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–14 (2021)
25. Nevejans, N.: What place for ai ethics in consumer protection in the light of the AI act and beyond?. In: Governance of Artificial Intelligence in the European Union. What impact on consumers?, pp. 183–203. Bruylant editions (2023)
26. Oinas-Kukkonen, H., Harjumaa, M.: Persuasive systems design: key issues, process model, and system features. *Commun. Assoc. Inf. Syst.* **24**(1), 28 (2009)
27. Orji, R., Busch, M., Dijkstra, A., Reisinger, M., Stibe, A., Tscheligi, M.: Personalization in persuasive technology. In: Adjunct Proceedings of the 11th International Conference on Persuasive Technology, pp. 96–99 (2016)
28. Pigozzi, G., van der Torre, L.: Arguing about constitutive and regulative norms. *J. Appl. Non Cl. Log.* **28**(2–3), 189–217 (2018)
29. Prakken, H.: Formal systems for persuasion dialogue. *Knowl. Eng. Rev.* **21**(2), 163–188 (2006)
30. Rosenfeld, A., Kraus, S.: Strategical argumentative agent for human persuasion. In: ECAI 2016, pp. 320–328. IOS Press (2016)
31. Shi, W., Wang, X., Oh, Y.J., Zhang, J., Sahay, S., Yu, Z.: Effects of persuasive dialogues: testing bot identities and inquiry strategies. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2020)
32. Verbeek, P.P.: Persuasive technology and moral responsibility toward an ethical framework for persuasive technologies. *Persuas.* **6**, 1–15 (2006)
33. Verheij, B., et al.: Grounded semantics as persuasion dialogue. *Computational Models of Argument: Proceedings of COMMA 2012* 245, 478 (2012)
34. Vilone, G., Longo, L.: Notions of explainability and evaluation approaches for explainable artificial intelligence. *Inf. Fusion* **76**, 89–106 (2021)
35. Wang, X., Shi, W., Kim, R., Oh, Y., Yang, S., Zhang, J., Yu, Z.: Persuasion for good: towards a personalized persuasive dialogue system for social good (2019). [arXiv:1906.06725](https://arxiv.org/abs/1906.06725)



Uncertain Machine Ethical Decisions Using Hypothetical Retrospection

Simon Kolker^(✉) , Louise Dennis , Ramon Fraga Pereira ,
and Mengwei Xu 

Department of Computer Science, University of Manchester, Manchester, UK
{simon.kolker,louise.dennis,ramon.fragapereira,
mengwei.xu}@manchester.ac.uk

Abstract. We propose the use of the hypothetical retrospection argumentation procedure, developed by Sven Ove Hansson to improve existing approaches to machine ethical reasoning by accounting for probability and uncertainty from a position of Philosophy that resonates with humans. Actions are represented with a branching set of potential outcomes, each with a state, utility, and either a numeric or poetic probability estimate. Actions are chosen based on comparisons between sets of arguments favouring actions from the perspective of their branches, even those branches that led to an undesirable outcome. This use of arguments allows a variety of philosophical theories for ethical reasoning to be used, potentially in flexible combination with each other. We implement the procedure, applying consequentialist and deontological ethical theories, independently and concurrently, to an autonomous library system use case. We introduce a preliminary framework that seems to meet the varied requirements of a machine ethics system: versatility under multiple theories and a resonance with humans that enables transparency and explainability.

Keywords: Machine ethics · Uncertainty · Argumentation · Moral theory

1 Introduction

Autonomous machines are an increasingly prevalent feature of the modern world. From spam filters [28] and fraud detectors [3], to drivers [32], medical practitioners [43] and soldiers [40], machines are being developed to automate tasks. Any decision affecting real people has the potential for ethical impact. Therefore machines are increasingly recognised as ethical agents. Moor [34] categorises such agents as either *implicitly* or *explicitly ethical*. Implicit ethical agents are built and situated by humans to have a neutral or positive effect, like an ATM machine; they do not utilise concepts of right and wrong in their internal decision making. As autonomous systems make more decisions with more responsibility, they need to reason about ethics *explicitly*. Allen et al. identify two strategies

for designing explicitly ethical systems [4]: *bottom-up* approaches train systems to make ethical decisions with learning techniques based on data from human decision making; *top-down* approaches encode principles and theories of moral behaviour (often drawn from philosophy) into rules for a selection algorithm, generally using techniques from the field of symbolic Artificial Intelligence (AI). In this paper, we propose and implement a top-down, explicitly ethical approach.

When an action is taken in the real world, its exact results are typically uncertain. As such, a top-down machine ethics system needs a mechanism for handling uncertainty over outcomes. There are mechanisms for handling uncertainty in AI, including Bayesian methods, Dempster–Shafer theory, fuzzy logics and others [36]. Nevertheless, it is currently unclear how they might integrate with machine ethics; there may be unanticipated philosophical implications.

Instead, we opted to operationalise and implement Sven-Ove Hansson’s hypothetical retrospection procedure [26]. Originating in Philosophy, the procedure was designed to guide ethical reasoning under uncertainty. It favours no specific ethical theory, but systematises the foresight argument pattern, extending an assessor’s perspective to judge decisions by the circumstances in which they were made. Therefore, arguments can be grounded in a variety of ethical theories. Over the past ten years, the field of machine ethics has implemented many such theories [41], yet there is no consensus over which is most effective. Philosophy too has not agreed which is morally correct, leaving implementers to choose from the perspective of stakeholder requirements and preferences. Thus, a mechanism for handling uncertainty that adapts to different ethical theories is desirable.

We outline the procedure via an example from Hansson [26]. Suppose an agent is given the choice between an apple and flipping a coin. If the coin lands heads, they win a free holiday to Hawaii. If the coin lands tails, they get nothing. Selecting the coin is clearly a valid choice. How might this decision be justified? Under hypothetical retrospection, we list each possible outcome: choosing the apple; choosing to toss the coin and winning the Hawaii holiday; choosing to toss the coin and losing. Next, we *hypothetically retrospect* from each outcome’s endpoint. Intuitively, the objective is to find an action whose outcomes do not lead the agent to *regret* the ethical implications of their action.¹ First, consider the coin’s outcomes: after winning the holiday, there cannot be regret since the Hawaii holiday is the best outcome; after losing the coin flip, the agent has nothing which is the worst outcome, but there is no regret since the agent justifies that they had a good chance of winning Hawaii, which is far better than an apple. Now, consider choosing the apple. Here, the agent regrets that they missed a chance of a holiday worth far more than an apple. We saw that choosing the coin did not lead to such regret. Therefore, the procedure advises we pick the coin, matching our intuition.

This paper operationalises the hypothetical retrospection procedure, and the foresight argument pattern it is based on. We implement and evaluate it with

¹ We recognise there is little ethical impact in this decision, besides maximising utility. It serves as an abstract example where one decision openly defeats another.

moral theories from Philosophy. We consider Deontology, which specifies a set of actions that are strictly forbidden [2], and a theory of consequentialism, which specifies an action is good if its consequences maximise good for the greatest number of people [35]. We illustrate our approach with the novel scenario of an autonomous library system. We demonstrate the system’s potential for explainability and versatility, while discussing issues and future work.

In Sect. 2, we will cover related work in the area and highlight this paper’s contribution. In Sect. 3, we will cover background on symbolic argumentation and uncertainty in Ethical Philosophy. In Sect. 4 we will recap Hansson’s description of hypothetical retrospection; in Sect. 5 we overview our problem formalism, including notation, the representation of probability and the argumentation model; Sect. 6 we describe our algorithm and implementation. Section 7 describes our test case of the autonomous library system, its formalism, and our results. Finally, in Sect. 8 we will identify the system’s potential benefits and its shortfalls left for future work.

2 Related Work

This is not the first attempt at building a top-down explicitly ethical machine. Tolmeijer et al. presents an exhaustive survey of implementations as of 2020, but finds the effect of uncertainty is rarely addressed [41]. Dennis et al. developed a framework suggesting how an autonomous system should act in unforeseen circumstances, with no positive outcomes. However, it does not address uncertainty between the likelihood of outcomes [20]. Probabilistic reasoning, such as Bayesian networks [39] and Markov models [19], has been applied to machine ethics, mostly with regards to maximising expected utility [17]. There are a number of criticisms of this approach which we will touch on in Sect. 3. Killough et al. goes further, architecting agents sensitive to utility risk and reward, with an ability to dynamically adjust risk-tolerance for the environment [30].

This paper is interested in a framework that incorporates a variety of philosophical ethical theories and allows for the combination of multiple theories, such as Deontology [2], Contractualism [8] and Virtue Ethics [27]. Different philosophical theories can advise on different courses of action, not only in tricky dilemma situations but sometimes even in situations where the moral choice seems intuitively obvious. There has been some work within machine ethics on comparing and combining different theories. For instance, Sholla et al. weights different principles and then uses fuzzy logic to decide between their recommendations [38]. Ecoffet and Lehman [23] use a voting procedure in which different ethical theories vote on recommendations but they struggle with the difficulty of comparing utilitarian theories that return a score for actions with deontological theories that tend to return a judgement that the action is either permissible or impermissible. Our framework enables a flexible approach in which the construction of an argument can treat all ethical theories equally, or allow one to have precedence over another. The HERA project [31] is of interest here—while it does not combine ethical theories it provides a single framework in which many

theories can be formalised and operationalised, allowing their recommendations to be compared. Cointe et. al [18] do something similar with an Answer-Set Programming approach though focused, in this case, on enabling the agent to make moral judgements about others. These systems could, potentially, be integrated into our argumentation framework to supply judgements on the rightness of an action and its consequences from the perspective of a particular moral theory.

Atkinson and Bench-Capon have developed a framework for ethical argumentation [9]. Like our work, assessments of action's outcomes are modelled as arguments. However, Atkinson and Bench-Capon's work remains concerned with epistemic conflicts between arguments (i.e. disputes between the truth of argument's circumstances) and annotates attacks and defends within the argumentation framework with values, aligning it with the philosophical theory of Virtue Ethics. Our work pivots away, focused purely on the ethical conflicts between arguments. We can assume epistemic truth because arguments are based only on potential, purely hypothetical, versions of events, each created from a single, shared set of information. This allows us to address moral conflict directly. It also lets us build uncertainty into the argumentation mechanism, instead of delegating it to a detail of argument attacks.

3 Background

The effect of uncertainty on machine ethics has been relatively unexplored largely due to the lack of research on how uncertainty impacts ethics in general. As Altham explains, there seems to be a gap in moral theory for uncertain situations [5]. He postulates this could be due to a belief among philosophers that no special principles are required; Moral Philosophy decides the virtues and it is up to Decision Theory to decide how they should be maximised under uncertainty.

Hansson shows that Utilitarian theories are straightforward in this regard [26]. These theories judge decisions based on numeric utilities assigned to their consequences. Expected utility Utilitarianism uses probabilities as weights to discount the utility of improbable outcomes. Hansson critiques this adaptation for the same reason as actual Utilitarianism: its assumption that outcomes can be appraised in terms of a single number (or at least done so both easily and accurately) often produces unintuitive outcomes. In the Apple-Coin scenario from Sect. 1, although it is evident that a trip to Hawaii holds more value than an apple, the extent of the difference in value remains uncertain. Adding more apples, such as 100, 1000, or 1001, does not necessarily make the deal any more appealing. In other words, apples and holidays are not proportionally comparable. There is no method of assigning relative utilities to all possible states. Brundage briefly surveys other critiques against consequentialist theories. First, they fail to account for personal social commitments, i.e. to friends and family. Second, they do not consider individual differences and rights, tending to favour the majority over any minority. Lastly, they place excessive demands on individuals to contribute to others [14].

Traditional Deontological systems [2] are made of principles which should never be violated. Hansson shows that any form of probabilistic absolutism,

where an action is not permitted if there is any chance of a rule violation, would be too restrictive. Therefore, an approach involving probability thresholds is often suggested. Here, an action is only forbidden when the probability that it violates a law exceeds some limit. The exact value of this limit is open for debate. It is tempting to suggest the limit should have some relation to the action's potential benefits, but this could soon reduce to some elaborate form of Utilitarianism, adamantly against the essence of the original theory.

Noticeably, most humans do not consciously rely on one philosophical, moral theory to make their decisions [13]. Nor do we think it is our place to choose a single theory to apply to machine ethics. As such, one of Hansson's key contributions is providing an argumentation procedure that can frame multiple, possibly conflicting theories rationally. To model this, we look to the study of abstract argumentation. Dung creates a framework of logically generated, non-monotonic arguments [22]. They can discredit each other with attacks, modelled as a binary relation between the arguments. Dung goes on to specify properties of a well-founded framework; he gives procedures for believing arguments based on their membership to framework extensions. This paper will take only the simple structure of Dung's framework. We leave it to Hansson's philosophy to define attacks and select arguments.

4 Hypothetical Retrospection

Hypothetical retrospection systematises ethical decision making with uncertain outcomes such that its judgements resonate with humans. In this section, we overview Hansson's description of the procedure from [26], before we operationalise it in Sect. 5.

Much of moral philosophy can be interpreted as an attempt to extend a decision maker's perspective. In promoting empathy, we invoke a perspective extending argument pattern to consider other's perceptions of our actions. For cases of uncertainty, Hansson argues it is helpful to extend our perspective with future perceptions of our actions. This means viewing, or hypothetically retrospecting on, a choice from the endpoint of its major foreseeable outcomes. As a result, the hypothetical outcomes, or the *potential branches of future development*, can be used to build resonate arguments about what to do in the present. Although Hansson proposes moral arguments that go beyond utility, duty or rights based calculations, the procedure is compatible with many theories of Ethics.

Hansson determines each action's branches of future development like a search problem. Theoretically, a decision's effects may be infinitely complex and far-reaching. The major search principle, therefore, is to find the most probable future developments which are the most difficult to defend morally. This will increase the chance of considering unethical scenarios. Branches should be developed to an endpoint sufficiently far to capture all morally relevant information. Intermediate information must be captured too: rule violations occurring before the point of retrospection still need to be considered. Additionally, and for the

sake of comparison, branches should be described with the same type of information where possible.² Hansson sees no reason not to create alternate branches based on the uncertainty of the decision maker’s own future choices, considering human’s inability to control their future actions. Whether an autonomous system has uncertainty over its future actions depends on the nature of the agent and its application architecture.

Our implementation assesses actions assuming their potential branches are provided. In future work, a planning algorithm could be adapted to the requirements above. For instance, the Probabilistic Planning Domain Definition Language (PPDDL) [42] is able to formalise different stochastic planning settings, e.g., Markov Decision Process (MDP) [25], Stochastic Shortest Path problems (SSP) [12], and Fully Observable Non-Deterministic planning [16]. This was superseded recently by the Relational Dynamic Influence Diagram Language (RDDL) [37] which has been adopted by the International Probabilistic Planning Competition (IPPC)³ and is thus the target input language for many planning implementations.

Using their potential branches, actions can be assessed with a selection of ethical theories. Hansson stresses we are not to assess actions in isolation; assessments are purely comparative. This is because decisions are not made in isolation. Given a choice between actions A and B, choosing A is choosing A-instead-of-B. Building action assessments from comparisons ensures all morally relevant information is taken into account.

Actions are compared by hypothetically retrospecting from the endpoint of each action’s potential branches of future development. We search for an action which never leads an agent to morally regret its choice in retrospect. Hansson argues against the term *regret* since it is considered a psychological reaction; humans often feel regret for actions they did not commit, or that they could not have known were wrong. By regret, therefore, we mean that the decision making was logically flawed under retrospection. As a result, we use the term *negative retrospection* to reflect this more technical definition. By hypothetically retrospecting between actions’ branches, we search for an action which does not lead to negative retrospection, or has full acceptability among its branches. If no such action exists, one should be selected that maximises acceptability in its most probable branches.

Therefore, Hypothetical Retrospection’s decisions are based on relevant ethical information using moral arguments that resonate with humans.

² The way in which consequences are discussed here may seem to exclude non-consequentialist theories. Hansson emphasizes that this is not the case. In his approach, consequences are broadly defined and their *information* includes agency, virtue intentions, and any other information necessary for moral appraisal.

³ <https://ataitler.github.io/IPPC2023/>.

5 Formalism

We define an ethical decision problem as a tuple $\langle A, B, S, U, F, I, m \rangle$, composed of an ethical environment and a set of available actions, each with a set of potential branches of future development.

An environment's ethically relevant properties are represented by the set S of Boolean variables; the set I defines the initial truth assignment to S , before actions are taken. For example, in the Coin-Apple scenario there are three state variables in S : s_1 represents whether or not we have an apple, s_2 whether or not we have gambled, and s_3 is whether or not we won a trip to Hawaii. In the initial state I , all these variables are false.

Ethical information for consequentialist and deontological theories are formalised with sets U and F . To capture the issue from Sect. 1, where different event outcomes have an immeasurably greater/lower utility, we have introduced the notion of utility classes.

Definition 1. (*Utility Class*) A utility class is an unordered set of individual utility assignments represented as tuples of $\langle s_k, \phi, v \rangle$, where s_k denotes a state variable in S and $v \in \mathbb{R}$ represents the variable's utility when assigned Boolean value ϕ .

The ordered set U contains utility classes in descending order of importance. Where $i < j$, all the positive utilities in u_i are considered greater than any utility in u_j ; all the negative utilities in u_i are considered less than any utility in u_j . To reiterate, the absolute utilities in lower indexed classes are considered immeasurably greater. In the Coin-Apple example, there are two utility classes in U . The first contains the utility assignment, $\langle s_3, True, 1 \rangle$ representing a utility of 1 for getting the Hawaii holiday. The second class has utilities immeasurably lower. It contains one assignment, $\langle s_1, True, 1 \rangle$ representing a utility of 1 for getting the apple.

The set F describes the states forbidden by a given deontological theory. This is not the same as defining a negative utility in U since utilities can be outweighed by a greater positive utility. In deterministic decision making environments, forbidden states can not be outweighed. They could represent, for instance, that someone was deceived, that a law (e.g., trespass) was broken, and so on—any action or outcome that can not be justified. The formalism assumes that the high-level rules have been translated into domain-level rules, applicable to the state variables in S .

Definition 2. (*Forbidden State*) A Forbidden State is a tuple $\langle s, \phi \rangle$ where $s \in S$ is a state variable forbidden from being assigned the Boolean value ϕ .

In the Coin-Apple scenario, F could contain a forbidden state, $\langle s_2, True \rangle$ representing a rule against gambling.

With an environment of ethical values, we define set A of available actions and set B of all potential branches of future development. We define a mapping, m , that associates every action with its potential branches of future development.

Each branch, $b \in m(a)$ is an ordered sequence of *events* that could occur after action a .

Definition 3. (*Event*) An event is a tuple of $\langle s, \phi, p \rangle$ where $s \in S$, ϕ is the new Boolean value of s , and p is the probability that the event occurs.

An event therefore represents the change in value of one state variable in S . A branch is a sequence of events that can occur after the action is taken.

For the Coin-Apple example, there are two available actions in A . Action a_1 represents choosing the apple. It maps to one branch $b_1 \in m(a_1)$, containing one event, $\langle s_1, True, 1 \rangle$ —if we choose to have an apple, we gain an apple; we have not gambled nor won a holiday to Hawaii. Action a_2 represents flipping the coin. It maps to two branches, $b_2, b_3 \in m(a_2)$. The branch b_2 contains one event, $\langle s_2, True, 1 \rangle$ —we gambled, but we have no apple and no holiday to Hawaii. The branch b_3 is the sequence of events $\langle s_2, True, 1 \rangle$ then $\langle s_3, True, 0.5 \rangle$ —first we gambled, then we won a holiday to Hawaii. The Coin-Apple problem is shown in Fig. 1.

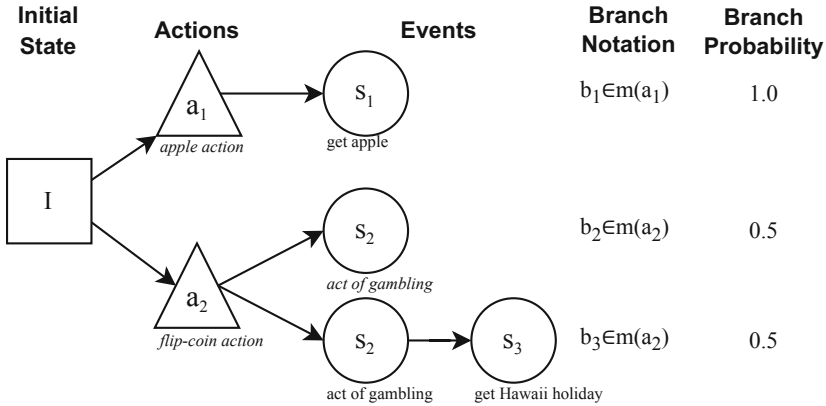


Fig. 1. Diagram for Coin-Apple scenario. Event nodes represent True assignment to a state variable. Actions map to a set of branches, represented by rows of event nodes. Probability of conjunction of branches' events given under branch probability.

We define the ethical decision problem and a permissible action. The definition of acceptability depends on the ethical theories under consideration (see Sect. 5.2).

Definition 4. (*Ethical Decision Problem*) An ethical decision problem is a tuple of $\langle A, B, S, U, F, I, m \rangle$ where A stands for a set of available actions, B the set of all potential branches of future development, S the set of Boolean state variables, U an ordered set of *utility classes*, F a set of forbidden state assignments, I the initial assignment of Boolean values to the variables in S , representing the initial state, and $m : A \rightarrow \mathcal{P}(B)$ (where \mathcal{P} is the powerset function) is a mapping of actions to potential branches of development.

Definition 5. (*Permissible Action*) Given an ethical decision problem, defined as a tuple of $\langle A, B, S, U, F, I, m \rangle$, a permissible action is an action, $a \in A$, such that for all potential branches of future development $b \in m(a)$, there is acceptability over their events in state space S . If no such actions exist, action a is permissible if it maximises the cumulative probability of its acceptable branches.

5.1 Probability Representation

In many scenarios, while a person may have an intuition that some events are more probable than others, their exact probabilities are unknown. This is most common when interacting with humans and complex systems. Our implementation supports the use of estimative as well as exact probability estimates. Kent found that intelligence reports tend to use *poetic* words like *probable* or *unlikely* [29]. The issue is that people have different interpretations of their meaning. Kent defined a relation for poetic words to mathematical probability ranges, as given in Table 1 from [29]. Our implementation supports both estimative and exact probabilities.

Table 1. Mathematical to poetic relation from Kent’s estimative probability [29].

100% Certainty			
The General Area of Possibility	93%	Give or take 6%	Almost certain
	75%	Give or take 12%	Probable
	50%	Give or take 10%	Chances about even
	30%	Give or take 10%	Probably not
	7%	Give or take 5%	Almost certainly not
0% Impossibility			

5.2 Argumentation Model

Hansson does not give steps for comparing action’s potential branches of future development in [26]. For our implementation, we chose to build comparative moral assessments with a simple argumentation network, based partially on the work of Atkinson et al. [10]. Here, arguments are generated logically from an *argument scheme*. For an action $a \in A$, selected in initial state I , resulting in the branch $b \in m(a)$ with probability p , the following argument is generated:

“From the initial state I, it was acceptable to perform action a, resulting in consequences b with probability p.”

For notation, this is written $Argument(b)$. We view this as a default argument that any action is acceptable. In our running example, the retrospective argument below is generated for b_3 , tossing the coin and winning the Hawaii holiday.

“From the initial state I , where $s_1 = s_2 = s_3 = \text{False}$, it was acceptable to perform the action a_2 , resulting in consequences with $s_2 = s_3 = \text{True}$ with probability 0.5.”

To determine an argument’s validity, we search for attacks from other actions’ arguments. Incoming attacks imply negative retrospection for not choosing an attacking action. To formalise Hansson’s retrospection, we generate attacks by posing critical questions on arguments’ claims [10]. For the branches $b_1 \in m(a_1)$, $b_2 \in m(a_2)$ and any generic moral principle, the following critical questions are asked for $\text{Argument}(b_1)$ to attack $\text{Argument}(b_2)$.

CQ1 *Did b_2 violate a moral principle that b_1 did not?*

CQ2 *Did a_2 hold a greater probability of breaking the moral principle than a_1 ?*

$\text{Argument}(b_1)$ only attacks $\text{Argument}(b_2)$ if both of these questions are answered positively. They represent negative retrospection for missing the chance to avoid violating a principle. The critical questions are asked both ways between all arguments supporting different actions, for every moral principle under consideration. The time and space complexity of answering the questions will differ for different theories. The desired ethical theories have to be encoded into the critical questions relative to a domain. For Utilitarianism and a generic deontological do-no-harm principle critical questions are embedded as follows:

- Utilitarian CQ1: *Did b_2 bring greater utility value than b_1 ?*
- Utilitarian CQ2: *Did a_2 expect greater utility value than a_1 ?*
- Do-no-harm CQ1: *Did b_2 cause harm where b_1 did not?*
- Do-no-harm CQ2: *Did a_2 expect greater probability of causing harm than a_1 ?*

After searching for attacks on all branches, an action should be selected with complete acceptability. If no such action exists, an action should be selected with maximal acceptability, i.e. summing the probability of each non-attacked argument and selecting an action with a maximal sum.

6 Implementation

We outline our implementation in Algorithm 1. Given an ethical decision problem, all actions are compared by their potential branches of future development (lines 2–4). There is a hypothetical retrospective argument made from the perspective of each branch in favour of its action. Attacks are generated between arguments by asking two critical questions based on an ethical theory. For our implementation we use a utilitarian and a deontological theory (lines 5–6), detailed later in Algorithms 2 and 3. Attacked branches are marked as such (lines 7–13). An action’s acceptability defaults to 1 and is subtracted by the cumulative probability of attacked branches. The action with maximum acceptability is selected (lines 17–25).

Algorithm 1 Arguments action's potential branches of future development. Returns index of action with maximum acceptability.

Input Ethical Decision Problem $\langle A, B, S, U, F, I, m \rangle$

Output Permissible Action $a \in A$

```

1: array attacked  $\leftarrow [False, \dots, False]$  of size  $length(B)$ 
2: for each  $a_i, a_j$  in  $\{(a_i, a_j) | a_i, a_j \in A \text{ and } a_i \neq a_j\}$  do
3:   for each  $b_k$  in  $m(a_i)$  do
4:     for each  $b_l$  in  $m(a_j)$  do
5:        $uTarget \leftarrow$  Target in Utilitarian CQs ( $b_k \in m(a_i), b_l \in m(a_j), U$ )
6:        $dTarget \leftarrow$  Target in Deontological CQs ( $b_k \in m(a_i), b_l \in m(a_j), F$ )
7:       if  $dTarget == uTarget$  and  $dTarget$  is not None then
8:          $attacked[uTarget] \leftarrow True$ 
9:       else if  $dTarget \neq uTarget$  and  $dTarget$  is None then
10:         $attacked[uTarget] \leftarrow True$ 
11:       else if  $dTarget \neq uTarget$  and  $uTarget$  is None then
12:         $attacked[dTarget] \leftarrow True$ 
13:       end if
14:     end for
15:   end for
16: end for
17: array acceptability  $\leftarrow [1, \dots, 1]$  of size  $length(A)$ 
18: for each  $a_i \in A$  do
19:   for each  $b_k \in m(a_i)$  do
20:     if  $attacked[k]$  then
21:        $acceptability[i] \leftarrow acceptability[i] - Probability(b_k)$ 
22:     end if
23:   end for
24: end for
25: return  $\leftarrow \arg \max_i(acceptability[i])$ 

```

Algorithm 2 embeds the theory of Utilitarianism into the critical questions. As explained in Sect. 5, branches are made from a list of events which each change a Boolean state variable with some probability. Variable utilities are defined by a set of utility classes, with assignments in lower indexed classes immeasurably greater. Algorithm 2 compares two potential branches and returns the index of a branch if it is defeated by the other branch through the critical questions. It is invoked by Algorithm 1 on line 5. Algorithm 2 counts from the lowest utility class upwards to find the first class where branch utilities are unequal. If found, critical question 1 is answered positively. The branch with the greater utility becomes the *attacker*, the other is the *defender* (lines 1–9). If utilities are equal through all classes, there are no attacks (lines 10–12). Otherwise, the defender branch attempts to use the foresight argument to defend itself: for each lower indexed class, if the defender's action has greater expected utility, defence is successful and there is no attack (lines 13–17). If the attacker action has greater or equal expected utility across all classes, defence fails and critical question 2 is positive. Thus, the defender branch is attacked (line 18).

Algorithm 2 For two potential branches of future development, finds target with lower utility in utility classes and no greater utility expectation to defend.

Input Action Branches $b_k \in m(a_i), b_l \in m(a_j)$, Utility Classes U

Output Index of Attacked Branch x

```

1: for  $c \leftarrow 0$  to  $\text{length}(U)$  do
2:    $\text{value}[i] \leftarrow$  Utility of  $b_k$  in  $U[c]$ 
3:    $\text{value}[j] \leftarrow$  Utility of  $b_l$  in  $U[c]$ 
4:   if  $\text{value}[i]$  is not  $\text{value}[j]$  then
5:      $\text{attacker} \leftarrow \arg \max_x (\text{value}[x])$ 
6:      $\text{defender} \leftarrow \arg \min_x (\text{value}[x])$ 
7:     break
8:   end if
9: end for
10: if  $\text{attacker}$  is None then
11:   return  $\leftarrow$  None
12: end if
13: for  $\text{lowerc} \leftarrow 0$  to  $c$  do
14:   if Expected Utility of  $a_{\text{attacker}}$  in  $U[\text{lowerc}] <$  Expected Utility of  $a_{\text{defender}}$  in
      $U[\text{lowerc}]$  then
15:     return  $\leftarrow$  None
16:   end if
17: end for
18: return  $\leftarrow$  defender

```

Algorithm 3 shows Deontology embedded into the critical questions, similar to Algorithm 2. Algorithm 3 iterates across the set of forbidden assignments and checks the events in either for a violation (lines 1–3). See Sect. 5 for forbidden assignments. If one branch has a violation that the other does not, then critical question 1 is positive (line 4 and 9). To defend itself, the violating branch’s action must have a greater probability of not making the assignment. If this is not true, critical question 2 is positive and the index of the violating branch is returned (lines 4–13). If no branch is attacked, neither index is returned (line 15).

Our implementation has no planning element, searching for action’s branches as discussed in Sect. 4. This is left for future work. Instead, we pass an ethical decision problem to an implementation of Algorithm 1 and a permissible action is output. We implement a web app with Flask and Python 3.8.9 to graph retrospection and alter utilities and deontological laws. The source code is available on GitHub at <https://github.com/sameysimon/HypotheticalRetrospectionMachine>.

Algorithm 3 For two potential branches of future development, finds target which breaks a deontological law with no greater expectation otherwise.

Input Action Branches $b_k \in m(A_i), b_l \in m(A_j)$, Forbidden States F

Output Index of Attacked Branch x

```

1: for each  $\langle s, \phi \rangle$  in  $F$  do
2:    $violation[i] \leftarrow$  Do events in  $b_k$  set  $s = \phi$ 
3:    $violation[j] \leftarrow$  Do events in  $b_l$  set  $s = \phi$ 
4:   if  $violation[i]$  and not  $violation[j]$  then
5:     if Probability of  $s = \phi$  in  $m(a_j) <$  Probability of  $s = \phi$  in  $m(a_i)$  then
6:       return  $\leftarrow i$ 
7:     end if
8:   end if
9:   if  $violation[j]$  and not  $violation[i]$  then
10:    if Probability of  $s = \phi$  in  $m(a_i) <$  Probability of  $s = \phi$  in  $m(a_j)$  then
11:      return  $\leftarrow j$ 
12:    end if
13:   end if
14: end for
15: return  $\leftarrow$  None

```

7 Autonomous Library Test Case

To demonstrate our implementation, we present an uncertain ethical decision problem and discuss our implementation's selected action given five sets of ethical considerations.

Suppose a student logs onto their University's autonomous library to revise for a test the next morning. All the other students started revision a month ago. As the student constructs various search terms for a recommendation, the system recognises that all other students have taken out the same book, implying it is very useful. Should the autonomous library use this data to recommend the book, allowing the student to revise quicker on the night before the test? If other students find out, they may feel unfairly treated; students who wait for a reference would get the same credit as those who find it themselves.

We model the scenario as an ethical decision problem, $\langle A, B, S, U, F, I, m \rangle$, with two actions in A mapping to ten branches in B , acting across four state variables in S . For action a_1 , to *recommend* the book, student data is compromised, the truth of which is represented by Boolean variable s_1 . Given a recommendation, there is a 0.6 chance the book is used, represented by s_2 . If they have the book, there is a 0.7 chance they will pass, s_3 , otherwise without the book there is a 0.3 chance they will pass, s_3 . Finally, there is a 0.05 chance other students will find out their data was compromised, s_4 . If the system ignores the book,

with action a_2 , there is a 0.3 chance the student will pass, again represented as s_3 .⁴ Figure 2 is a decision tree labelled with probabilities and branch notation.

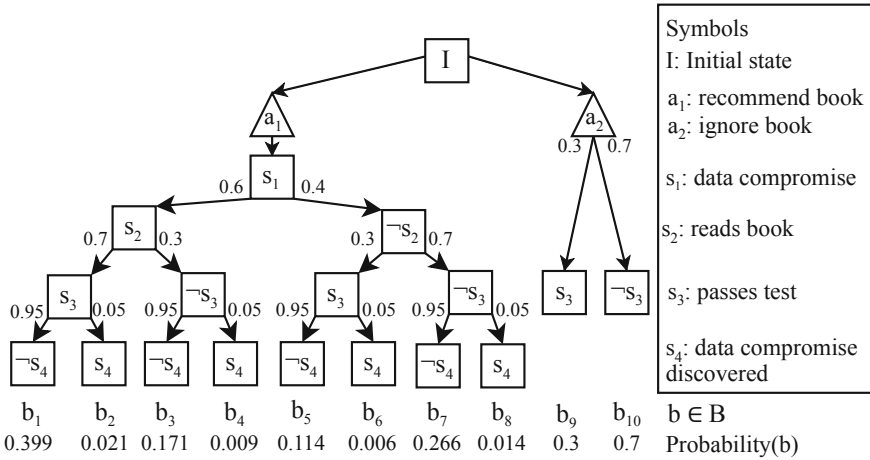


Fig. 2. Decision tree of possible events in Autonomous Library problem. Triangles represent actions and boxes variable assignments, \neg represents *False* assignment.

An argument is generated from each branch’s endpoint, representing positive retrospection. Using the argument scheme from Sect. 5.2, $Argument(b_1)$ is the following:

“From the initial state, I, where $s_1 = s_2 = s_3 = s_4 = False$, it was acceptable to perform the action, a_1 , resulting in consequences with $s_1 = s_2 = s_3 = True$ and $s_4 = False$, with probability 0.399.”

The argument claims it was acceptable to recommend the book, resulting in a data protection violation (s_1), the student reading the book (s_2) and passing the test (s_3), with the data breach kept a secret ($s_4 = False$), at a probability of 0.399.

7.1 Consequentialism with One Assignment

First we test our implementation considering the ethical theory of consequentialism. We set U to have one utility class with one utility assignment, $\langle passesTest, 1, True \rangle$. The only value is the student passing. Intuitively, the action maximising the probability of passing should be chosen; hypothetical retrospection agrees. The argumentation graph in Fig. 3 shows the retrospection.

⁴ There is discourse on whether a decision to act should be judged the same as a decision not to act [24]. We consider ignoring the book an action, an act of discrimination for example, which is assessed the same as the act to recommend.

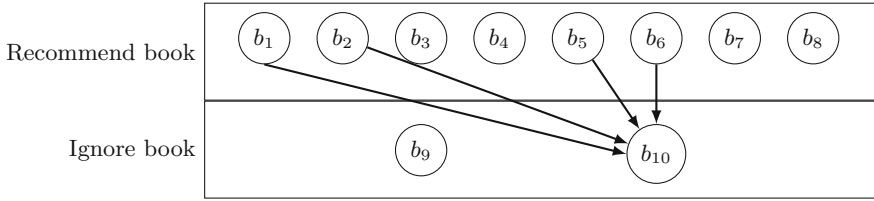


Fig. 3. Graph of retrospection between hypothetical branches of development with only the utility of the student passing in consideration. Incoming edges on an argument represent negative retrospection for not selecting the attacking argument’s action.

Every branch has acceptability, except $b_{10} \in m(a_2)$ where the student fails after the system chooses *ignore*, with 0 utility and 0.3 probability (*probably not* in Kent’s words). This branch has a lower utility than the four *recommend* branches where the student passes: $b_1, b_2, b_5, b_6 \in m(a_1)$. They cause $Argument(b_{10})$ to answer critical question 1 positively when attacked by these arguments. Since *recommend* has a greater utility expectation, or a greater probability of the student passing, $Argument(b_{10})$ cannot defend itself in critical question 2. Thus, there is no reason to select *ignore*; from the perspective of b_{10} ’s endpoint there is negative retrospection. There are no other attacks. Therefore by hypothetical retrospection action a_1 , *recommend*, should be selected.

7.2 Consequentialism with Two Equal Assignments

Now we consider two utility assignments of the same class: $\langle passesTest, 1, True \rangle$ and $\langle othersFindOut, -1, True \rangle$. This invokes the risk of others finding out their data was used, with others finding out judged as bad as the student passing is good. Retrospection is shown in Fig. 4.

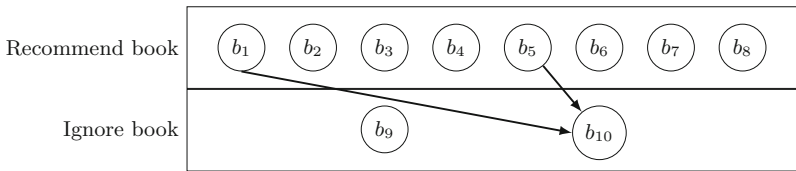


Fig. 4. Graph of retrospection between hypothetical branches of development with the cost of others finding out data was compromised equaling the utility of the student passing.

Again, only branch $b_{10} \in m(a_2)$ has negative retrospection, when the student fails after the system chooses to *ignore* the book. This time only two of *recommend*’s branches have greater utility, $b_1, b_5 \in m(a_1)$. Action *recommend* still has a greater utility expectation, so *ignore* cannot be defended in critical question 2. Therefore, *recommend* is selected.

7.3 Consequentialism with Unequal Assignments

The utility of students discovering the data compromise can be lowered such that *recommend*'s expected utility is lower than *ignore*'s, for example with the assignment $\langle \textit{othersFindOut}, -5, \textit{True} \rangle$. Now, attacks fire the other way, displayed in Fig. 5. When *recommend* is chosen and other students find out, as in $b_2, b_4, b_6, b_8 \in m(a_1)$, the utility is lower than *ignore*'s branches. This answers critical question 1 positively for attacks on these branch's arguments. There is no defence since *ignore* has a greater utility expectation so critical question 2 is positive. *Recommend* can lead to the highest utility branches with b_1 and b_5 , but unlike before, b_{10} defends citing its higher utility expectation. Thus, *ignore* is selected with full acceptability.

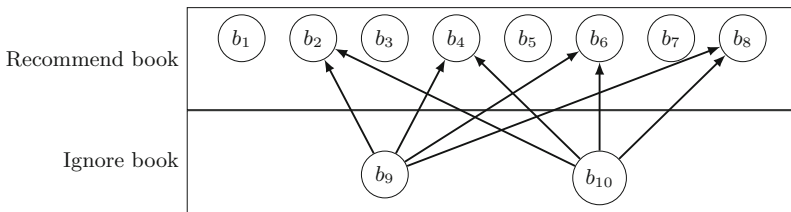


Fig. 5. Graph of retrospection between hypothetical branches of development with the cost of others finding data was compromised outweighing the utility of passing.

Deciding utilities is difficult without further details, i.e. the student's grades, data preferences, etc. Ideally, branches would be developed until enough morally relevant information is described, but this is not always computationally viable. Even so, exact utilities are subjective. We confront this issue with utility classes. Supposing *othersFindOut* has utility immeasurably lower than *passesTest*, we form two classes. The first has assignment $\langle \textit{othersFindOut}, -1, \textit{True} \rangle$; the second has $\langle \textit{passesTest}, 1, \textit{True} \rangle$. The resulting retrospection is the same as in Fig. 5, with the cost of others' knowledge outweighing the benefits of passing.

7.4 Deontology with Consequentialism

Finally we consider a deontological theory against the misuse of others' data. This could be the UK Law, requiring under the Data Protection Act that personal data is to only be used for specified, explicit purposes [1]. Otherwise, there could be a violation of the Doctrine of Double Effect, having four conditions [33]: 1. that the action in itself from its very object be good or at least indifferent; 2. that the good effect and not the evil effect be intended; 3. that the good effect be not produced by means of the evil effect; 4. that there be a proportionately grave reason for permitting the evil effect. If we consider non-consensual use of students' data as bad and helping a student to pass the exam to be good, then the fact that the bad effect is required in order to bring about the good effect

breaks the third condition above, and, therefore, is not permissible. We build on our first test in Sect. 7.1 which selected *recommend* with utility assignment $\langle \text{passesTest}, 1, \text{True} \rangle$. Adding forbidden state $\langle \text{dataProtectionViolation}, \text{True} \rangle$ to F results in the retrospection shown by Fig. 6. Every argument from *ignore* attacks every argument from *recommend* since *ignore* avoids violating the law.

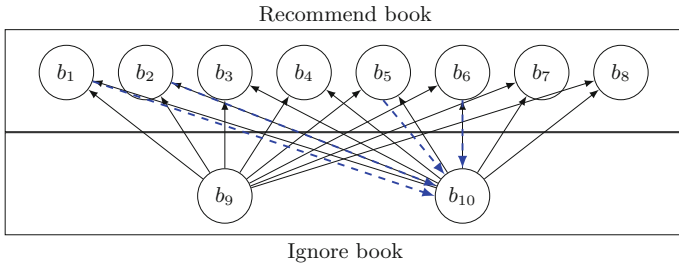


Fig. 6. Graph of retrospection between potential branches of development with one Consequentialist assignment and one Deontological law. Consequentialist attacks are dashed blue; Deontological attacks are solid black.

Under our previous Consequentialism, *recommend* is still chosen with the same attacks on $\text{Argument}(b_{10})$ as before. This conflict represents a moral dilemma, where no choice is normatively inferior to another [26]. The aim is to maximise acceptability amongst the most probable branches. Since all arguments from *recommend* are attacked, there is 0 acceptability for that action; one argument from *ignore* is attacked with 0.7 probability meaning *ignore* is selected with the maximum acceptability of 0.3.

8 Discussion

Our goal here is to extend the typical approach to machine ethics, which is the assessment of a single action from the perspective of a single ethical theory, often without any account of probability or uncertainty. We have formalised Hansson’s hypothetical retrospection procedure, systematising moral assessments as comparisons between consequences. This forms richer judgements beyond the evaluation of utilities. Furthermore, our moral assessments are comparisons between retrospective justifications of hypothetical consequences. One might ask how this differs from directly analysing the properties of consequences? For machines, it gives a procedure for selecting actions and providing justifications. For humans, it offers a resonance that allows us to make clearer judgements [26]. It also allows us, in the future, to build on existing work for evaluating actions from the perspective of individual ethical theories and combining those judgements into arguments. Essentially our proposal extends, rather than replaces, existing mechanisms for evaluating actions against a single ethical theory.

The retrospective procedure formalised by the critical questions resembles real life discussion: a claim against an argument and a chance to refute. Say someone takes action a_2 in preference to a_1 and a principle is broken. Retrospective argumentation through the critical questions produces a dialogue similar to the following:

1. You should have chosen a_1 because it didn't break this moral principle.
2. No, because there is a greater probability of breaking some other principle with a_1 . If I was given the decision again, I would make the same choice.

Real life discussion may not be so civil, but if facts were agreed upon, this is the logical dialogue. Resonance with real life has utility for agent transparency and explainability, important for ethical AI [11] and stakeholder buy-in.

The implementation is theory-neutral, allowing multiple principles and theories to be considered at once, more analogous to human decision-making. Implementational work remains, not least the integration into a planning system to generate branches, but also evaluation against a wider range of ethical theories (e.g. Virtue Ethics) to see how easily they answer the critical questions. We also wish to develop the evaluation of action's consequences along branches, not just at the branches end—for instance, if someone is made unhappy as a consequence of some action, but then we compensate them by the end of the branch, can we ignore that we caused them (albeit temporary) unhappiness?

Implementations of hypothetical retrospection could be integrated into more general agent reasoning either as modules on top of an existing autonomous system, possibly similar to Arkin's governor architecture [7]. Cardoso et. al have, for instance, considered how such ethical governors might integrate with BDI agents [15]. Alternatively hypothetical retrospection could be implemented as a general decision-making process in which, for instance, the extent to which an action enables an agent to achieve or maintain goals could be included together with the arguments based upon ethical theories. Systems of this kind—in which all reasoning is encompassed within the ethical reasoning system can be seen in, for instance, the GenEth System [6] where “maintain readiness” is treated as an ethical duty or the HERA system [31] where in [21] the system defaults to utilitarianism to decide among actions all of which are considered equally valid according to some ethical theory.

Our current implementation has a fairly simple approach to the integration of ethical theories. Some theories are directly incompatible, potentially leading to “worst of both worlds” solutions. Additionally, the use of utility classes needs careful handling. When utilities are of a greater class, they are prioritised, no matter how remote their probabilities. Extending the Coin-Apple scenario, suppose an agent is offered a free apple every day—as opposed to some number of apples all at once, or suppose the chance of winning the Hawaii holiday is extremely low, or both. The justification for sacrificing a lifetime supply of apples for a small chance of a holiday is considerably weaker than sacrificing one apple for a 50/50 chance of a holiday. Expected utility clearly has a part to play, even if the calculation of such utilities is non-trivial. The difficulty in estimating utilities, and the fact that utilities may depend upon unknown factors such as a

person's financial situation, mean there is uncertainty in the evaluation of state utilities which our framework currently does not address.

There will be some computational complexity in searching and representing actions' potential branches of future development. In Sect. 4, we note Hansson's principles for optimising search but it remains to be seen if this can be practically implemented to keep planning tractable for common problems.

Nevertheless we believe the hypothetical retrospection framework practically handles many of the issues in machine ethics—particularly the handling of uncertainty and the lack of any real agreement on the best moral theory.

Open Data Statement

This work is licensed under a Creative Commons Attribution 4.0 International License. The tools/examples shown in this paper and instructions on reproducibility are openly available on GitHub at: <https://github.com/sameysimon/HypotheticalRetrospectionMachine>.

Acknowledgements. We would like to thank the University of Manchester for funding and EPSRC, under project Computational Agent Responsibility (EP/W01081X/1).

References

1. Data protection. Ministry of Justice. <https://www.gov.uk/data-protection>. Accessed 07 July 2023
2. Alexander, L., Moore, M.: Deontological Ethics. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2021 edn. (2021)
3. Alhaddad, M.M.: Artificial intelligence in banking industry: a review on fraud detection, credit management, and document processing. *ResearchBerg Rev. Sci. Technol.* **2**(3), 25–46 (2018)
4. Allen, C., Smit, I., Wallach, W.: Artificial morality: top-down, bottom-up, and hybrid approaches. *Ethics Inf. Technol.* **7**(3), 149–155 (2005)
5. Altham, J.E.: Ethics of risk. In: *Proceedings of the Aristotelian Society*, vol. 84, pp. 15–29. JSTOR (1983)
6. Anderson, M., Anderson, S.L., Berenz, V.: A value-driven eldercare robot: virtual and physical instantiations of a case-supported principle-based behavior paradigm. *Proc. IEEE* **107**(3), 526–540 (2019). <https://doi.org/10.1109/JPROC.2018.2840045>
7. Arkin, R.C.: Governing lethal behavior: Embedding ethics in a hybrid deliberative/reactive robot architecture. In: *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, pp. 121–128 (2008)
8. Ashford, E., Mulgan, T.: Contractualism. In: Zalta, E.N. (ed.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2018 edn. (2018)
9. Atkinson, K., Bench-Capon, T.: States, goals and values: revisiting practical reasoning. *Argum. Comput.* **7**(2–3), 135–154 (2016). <https://doi.org/10.3233/aac-160011>

10. Atkinson, K., Bench-Capon, T., McBurney, P.: Justifying practical reasoning. In: Proceedings of the Fourth International Workshop on Computational Models of Natural Argument (CMNA 2004), pp. 87–90 (2004)
11. Balasubramaniam, N., Kauppinen, M., Hiekkanen, K., Kujala, S.: Transparency and explainability of ai systems: ethical guidelines in practice. In: Requirements Engineering: Foundation for Software Quality: 28th International Working Conference (REFSQ), pp. 3–18. Springer (2022)
12. Bertsekas, D.P., Tsitsiklis, J.N.: An analysis of stochastic shortest path problems. *Math. Oper. Res.* **16**(3), 580–595 (1991)
13. Bialek, M., Neys, W.D.: Dual processes and moral conflict: evidence for deontological reasoners’ intuitive utilitarian sensitivity. *Judgm. Decis. Mak.* **12**(2), 148–167 (2017)
14. Brundage, M.: Limitations and risks of machine ethics. *J. Exp. & Theor. Artif. Intell.* **26**(3), 355–372 (2014)
15. Cardoso, R.C., Ferrando, A., Dennis, L.A., Fisher, M.: Implementing ethical governors in BDI. In: Alechina, N., Baldoni, M., Logan, B. (eds.) *Engineering Multi-Agent Systems*, pp. 22–41. Springer International Publishing, Cham (2022)
16. Cimatti, A., Pistore, M., Roveri, M., Traverso, P.: Weak, strong, and strong cyclic planning via symbolic model checking. *Artif. Intell.* **147**(1–2), 35–84 (2003)
17. Cloos, C.: The utilibot project: an autonomous mobile robot based on utilitarianism. *AAAI Fall Symposium - Technical Report* (2005)
18. Cointe, N., Bonnet, G., Boissier, O.: Ethical judgment of agents’ behaviors in multi-agent systems. In: Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, pp. 1106–1114. *AAMAS ’16*, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2016)
19. Davis, M.H.: *Markov Models and Optimization*. Routledge (2018)
20. Dennis, L., Fisher, M., Slavkovik, M., Webster, M.: Formal verification of ethical choices in autonomous systems. *Robot. Auton. Syst.* **77**, 1–14 (2016)
21. Dennis, L.A., Bentzen, M.M., Lindner, F., Fisher, M.: Verifiable machine ethics in changing contexts. *Proc. AAAI Conf. Artif. Intell.* **35**(13), 11470–11478 (2021). <https://ojs.aaai.org/index.php/AAAI/article/view/17366>
22. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artif. Intell.* **77**(2), 321–357 (1995)
23. Ecoffet, A., Lehman, J.: Reinforcement learning under moral uncertainty. In: Meila, M., Zhang, T. (eds.) *Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 139, pp. 2926–2936. PMLR (2021). <https://proceedings.mlr.press/v139/ecoffet21a.html>. Accessed 18–24 July 2021
24. Foot, P.: The problem of abortion and the doctrine of the double effect. *Oxford Rev.* **5**, 5–15 (1967)
25. Hansen, E.A., Zilberstein, S.: Lao^{*}: a heuristic search algorithm that finds solutions with loops. *Artif. Intell.* **129**(1–2), 35–62 (2001)
26. Hansson, S.: *The Ethics of Risk: Ethical Analysis in an Uncertain World*. Springer (2013)
27. Hursthouse, R., Pettigrove, G.: Virtue Ethics. In: Zalta, E.N., Nodelman, U. (eds.) *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edn. (2022)

28. Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K., Alazab, M.: A comprehensive survey for intelligent spam email detection. *IEEE Access* **7**, 168261–168295 (2019). <https://doi.org/10.1109/ACCESS.2019.2954791>
29. Kent, S.: Words of estimative probability. *Stud. Intell.* **8**(4), 49–65 (1964)
30. Killough, R., Bauters, K., McAreavey, K., Liu, W., Hong, J.: Risk-aware planning in bdi agents. In: *International Conference on Agents and Artificial Intelligence*, vol. 2, pp. 322–329. SciTePress (2016)
31. Lindner, F., Bentzen, M.M., Nebel, B.: The hera approach to morally competent robots. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6991–6997. IEEE (2017)
32. Ma, Y., Wang, Z., Yang, H., Yang, L.: Artificial intelligence applications in the development of autonomous vehicles: a survey. *IEEE/CAA J. Autom. Sinica* **7**(2), 315–329 (2020). <https://doi.org/10.1109/JAS.2020.1003021>
33. Mangan, J.T.: An historical analysis of the principle of double effect. *Theol. Stud.* **10**(1), 41–61 (1949)
34. Moor, J.H.: *The Nature, Importance, and Difficulty of Machine Ethics*, pp. 13–20. Cambridge University Press (2011)
35. Mosdell, M.: Act-Consequentialism. *Encyclopedia of Global Justice*, pp. 2–2. Springer, Netherlands (2011)
36. Saffiotti, A.: An ai view of the treatment of uncertainty. *Knowl. Eng. Rev.* **2**(2), 75–97 (1987)
37. Sanner, S., et al.: Relational dynamic influence diagram language (rddl): Language description. Unpublished ms. Australian National University, vol. 32, p. 27 (2010)
38. Sholla, S., Mir, R.N., Chishti, M.A.: A fuzzy logic-based method for incorporating ethics in the internet of things. *Int. J. Ambient Comput. Intell. (IJACI)* **12**(3), 98–122 (2021)
39. Stephenson, T.A.: *An introduction to bayesian network theory and usage*. Technical report, Idiap (2000)
40. Szabadföldi, I.: Artificial intelligence in military application-opportunities and challenges. *Land Forces Acad. Rev.* **26**(2), 157–165 (2021)
41. Tolmeijer, S., Kneer, M., Sarasua, C., Christen, M., Bernstein, A.: Implementations in machine ethics: a survey. *ACM Comput. Surv. (CSUR)* **53**(6), 1–38 (2020)
42. Younes, H.L.S., Littman, M.L.: Ppddl1.0: An extension to pddl for expressing planning domains with probabilistic effects. In: *Technical Report -Carnegie Mellon University* (2004)
43. Yu, K.H., Beam, A.L., Kohane, I.S.: Artificial intelligence in healthcare. *Nat. Biomed. Eng.* **2**(10), 719–731 (2018)



Towards Convention-Based Game Strategies

Shuxian Pan^{1,2}(✉)  and Carles Sierra¹ 

¹ Artificial Intelligence Research Institute (IIIA-CSIC), Campus de la UAB, 08193 Bellaterra, Barcelona, Spain

² School of Computing Technologies, RMIT University, Melbourne, VIC 3001, Australia

{pshuxian,sierra}@iiaa.csic.es

<https://www.iiaa.csic.es>

Abstract. To effectively develop cooperative multiagent systems, we introduce an architecture that facilitates the agents' dynamic adoption of conventions. It expands an existing agent model's action selection architecture with a component that uses Natural Language Processing techniques. This component embeds conventions into agent interaction strategies to improve the predictability of other agents' actions if all agents adopt the same conventions in their strategies.

Keywords: Norm extraction · Multiagent systems · Cooperative AI

1 Introduction

Conventions can be defined as recurrent behaviour patterns of human communities [2] that increase the predictability of interaction outcomes. In an AI context, conventions can coordinate agents' actions in multiagent systems and simplify the agents' decision-making machinery. In general, not all agents are necessarily willing or capable of adhering to the same conventions. However, in a cooperative multiagent system, we may assume that the agents will agree to adhere to the same conventions to improve their collective performance.

In the literature, different terms refer to these agent behaviour patterns, often used to determine whether a specific action is “correct” and sometimes represent typical behaviour in a society. The term *convention* is often related to patterns that result from an agreement among members of a given community or culture. The term *norm* is often associated with legal aspects of behaviour and contains rewards or sanctions. The general term *rule of behaviour* is also commonly used. In this paper, we will use the term *convention*, but our proposal could be applied to norms or rules, as they all share the same basic structure. If we consider the concepts of rules and conventions, the boundary between them in real life can be pretty vague since the conventions can easily be settled as rules with the agreement between the agents.

Previous research has demonstrated that conventions considerably enhance a multiagent system’s overall performance. Conventions may have either an external or an internal origin. Machine learning methods, such as Reinforcement Learning, have identified particular agents’ communication patterns as *intrinsic* conventions [14, 20, 22, 40]. *Extrinsic* human social conventions are quite often imposed over human communities or multiagent systems [18, 23, 24, 43].¹ Conventions are frequently hand-coded in those previous works, representing a costly effort for engineers. Any modification to the system’s conventions often requires manual checks of their soundness. Although previous work has introduced conventions into multiagent systems in an automated style [35], to our knowledge, no attempt has been made to develop an automatic NLP pipeline to embed conventions into a system. This paper introduces an architecture based upon [33] primarily concerned with processing natural language conventions. We believe it will facilitate the creation of convention-based multiagent systems and give users control over multiagent behaviour. To illustrate the components of the architecture, we will use the board game Hanabi. Nonetheless, we will also attempt to present the architecture, particularly the NLP component, in as general terms as possible. Our vision is to “program” agents by *declaring* in natural language the game rules and the strategic behaviour that the agent should show.

In short, the contributions of this paper are: (1) to propose a generalisation of the decision-making component of an existing architecture [33], (2) to discuss the use of an NLP pipeline for norm extraction, and (3) to explore the combination of ontologies and convention patterns to represent conventions formally. This paper is divided into four sections. Section 2 describes the research problem and our focus, Sect. 3 introduces the relevant background knowledge, Sect. 4 presents our preliminary proposal for the architecture, and Sect. 5 discusses future work. This paper is submitted as a short paper presenting ongoing work.

2 Problem

Our research objective is to study “how agents adapt their strategies to conventions in a cooperative multiagent system.” To achieve this objective, we need to design an architecture with the following aspects/requirements:

- **Conventions.** Conventions are usually expressed in natural language and can be represented in a logical formalism [9, 16, 17]. Although conventions often lack the sanctioning aspect of (legal) norms, their structure is similar: both impose constraints on human or agent behaviour [1]. Thus, in our architecture, we focus on using NLP techniques for *norm extraction* (see Sect. 3.1) to process the conventions. We aim to automatically translate natural language conventions into a machine-readable representation for our agent model. Section 4.1 will outline this mechanism.

¹ Some of these works focus on *policies* rather than conventions. These two concepts are similar, although policies sometimes have a more probabilistic flavour [19]: there is the option that an agent probabilistically chooses an alternative to the action recommended by the policy when *exploring* the space.

- **Knowledge Representation.** Agents must have a *model of the environment* to observe the actions of others and their consequences. To achieve their common goal, agents must also be able to understand, that is, find explanations for, the actions made by other agents. Hence, our agent model must include some *Theory of Mind* (ToM) representation [33] (see Sect. 3.2).
- **Reasoning.** The architecture of [33] includes a component to determine the agent’s next action. This component contains a set of conventions to select an action from among several possibilities. We follow the same path in our architecture with some adaptations (see Sect. 4.4). For instance, decision rules in [33] contain priorities expressed with natural numbers. The higher the number, the higher the priority. Since [33] only shows examples requiring a few simple conventions, it is enough to make them hand-coded. However, for larger sets of conventions, possibly more complex, we will need (semi-)automated approaches for priority determination.

We will use the game Hanabi as a testbed. Hanabi is a cooperative board game for two to five players that will serve as an illustrative example for this paper. The game’s goal is to build card stacks in a specific order. There are five distinct colour stacks, each containing up to five cards that must be played in order from 1 to 5; the more cards correctly played, the better the final score. The players cannot view their cards, only those of their fellow players. The game actions are: providing a hint on a card held by another player (so-called “clue card”, saying its colour or number), discarding a card, and playing a card. There is a series of conventions that complement these rules. For instance, H-Group Conventions² are conventions organised and published by Hanabi players. Table 1 shows some of them. This paper will mainly use the two conventions labelled “Chop” in Table 1 as examples. See the following illustration of the use of a convention.

Example: Alice has no clued cards. Bob has cards in the first, third and fifth slots clued. Considering only the game rules, players could play or discard any card. However, when following the “Chop” conventions, Alice *should* discard her *chop* card in her fifth slot, and Bob should discard the card in his fourth slot.

Table 1. Some conventions extracted from the H-Group Conventions.

Labels	Conventions
Chop	The right-most unclued card in a player’s hand is called their “chop” card When a player needs to discard, they should discard their chop card
Types of Clues	Players are only allowed to give two types of clues: a Play Clue (meaning to play the focused card) and a Save Clue (meaning to save the focused card for later)
Play Clues	Play Clues can be given with either a colour or number clue
Save Clues	Save Clues can only be given to chop cards

² <https://hanabi.github.io>.

3 Background

3.1 Norm Extraction

This paper’s primary focus is on conventions. Hence, how to process conventions is the critical part of the architecture we want to discuss in detail in this paper. We will adapt existing state-of-the-art norm extraction techniques. Norm extraction is a sub-task of natural language processing that involves recognising and extracting norm structures from natural language text using (semi-)automatic approaches. Most past research has been conducted in the context of norm extraction from legal documents. Even though the definitions of norms and conventions are slightly different, from an NLP perspective, the differences are such that we can apply legal norm extraction techniques to conventions. However, we know that certain procedures were developed to address specific concerns of legal norms that may not be needed to process conventions. Unlike legal texts, the structure and semantics of conventions, particularly those for Hanabi, can be pretty simple and limited.

Recent norm extraction techniques and a general overview were evaluated in [13]. Norm extraction, like other NLP tasks, usually begins with text preprocessing. Several existing NLP toolkits and pipelines (e.g. NLTK,³ or Stanza⁴) provide automated preprocessing techniques, including tokenisation, removal of stop words and punctuation, and lemmatisation. In addition, some particular syntactic structures of the input text need to be modified, such as lists of items with enumerations, colons and numerous references which contains punctuation and alpha-numeric characters (prevalent in legal texts), to avoid failure of the sentence processing [46]. After this preprocessing, subsequent steps consist of parsing and/or tagging, the standard techniques in NLP. Some studies applied pre-trained general parsers, such as the Stanford parser⁵ [11, 41, 46], or the Berkeley parser⁶ [41]. Based on various grammar systems, parsers generate the grammar tree structure of the sentence over words (dependency) or phrases (constituency). Conversely, tagging is the annotation process that can attach syntactic, semantic, or logical features to words. The tagging and parsing processes can be performed simultaneously by the same tool or by tagging before parsing. For instance, we can automatically annotate words with part-of-speech (POS) tags before the parsing process starts. Some specific annotations, such as deontic information of legal norms, can only be done manually. Additional knowledge sources, such as Word Net [11] and Wikipedia [25, 37], were explored to supplement the semantic representation. In [41], the method was more sophisticated, containing a task-specific dictionary and vectorisation of sentences. Once annotated data is collected, machine learning or symbolic methodologies can incorporate norms into an AI system. One example is using the tax code as training data for a complicated multi-layer convolutional neural network (CNN)

³ <https://www.nltk.org>.

⁴ <https://stanfordnlp.github.io/stanza/index.html>.

⁵ <https://nlp.stanford.edu/software/lex-parser.shtml>.

⁶ <https://github.com/slavpetrov/berkeleyparser>.

to classify sentences into several deontic categories [32]. Another example is training a classifier based on the syntactic/semantic features in the norm sentences to extract specific elements from them [15]. Symbolic applications include NL2KR [16] and the Candc and Boxer tool chain [11]. Although their processes are not identical, their primary goal is to produce formal representations of input sentences adopting Combinatory Categorical Grammar (CCG). However, neither tool is maintained, so users should anticipate compatibility issues when applying them. As for the evaluation, although [13] created the gold standard for semantic parsing in the legal domain, the size of this test set is limited. Like the other test sets for legal norm extraction, they have to be annotated by experts. For metrics, recall, precision, and accuracy were commonly used [6, 12, 36, 41].

3.2 Theory of Mind

Theory of Mind (ToM) is an ability acquired through social interaction. To comprehend others' actions, humans need to create models of the beliefs of others. This ability can be further nested. For instance, not just the beliefs that agent i holds about the beliefs of j , but also the beliefs that agent i holds about the beliefs of j about the beliefs of k and so on. The former example is a first-order ToM statement, and the latter a second-order ToM statement. Although a higher order implies a deeper degree of comprehension, the rise in complexity will usually offset any gains [44, 45], so it is vital to consider and control the depth. Reference [33] offers a thorough introduction to the previous research. Reference [10] introduced the (potential) application of ToM in AI but also indicated that many existing approaches neglected or over-simplified the mental states of agents, which is critical for the human mind and their mental process. In [33], ToM focuses on deriving explicit beliefs, so the mental states are not involved. For our architecture, it will be the same. Our agents require this capacity to predict the actions of others and act upon that prediction.

3.3 Hanabi

Hanabi has been proposed as a challenging game⁷ to explore the limits of machine learning or rule-based systems [7, 21, 30, 40, 42, 43]. For example, a new game-play setting named *other-play* [42] (implemented from [22]), or an adversarial mechanism to self-play [43]. In these works, the agents either followed *different* conventions or played with human players. We do not discuss them in this article since, as stated previously, we focus our work on agents that play together and adhere to the same conventions. Thus, there is no need to address potential convention conflicts.

4 Architecture

Figure 1 illustrates the complete architecture we propose. Reference [33] serves as the inspiration. In our approach, a cooperative agent receives as input: informa-

⁷ A detailed literature review is provided by [4].

tion about the environment, messages from other agents, and a list of conventions to be followed when making decisions. A series of modules process the input so the action selection component can determine the action to take. These modules are divided into two blocks: “NLP” and “Agent Decision-making”. The following subsections describe the modules in detail.

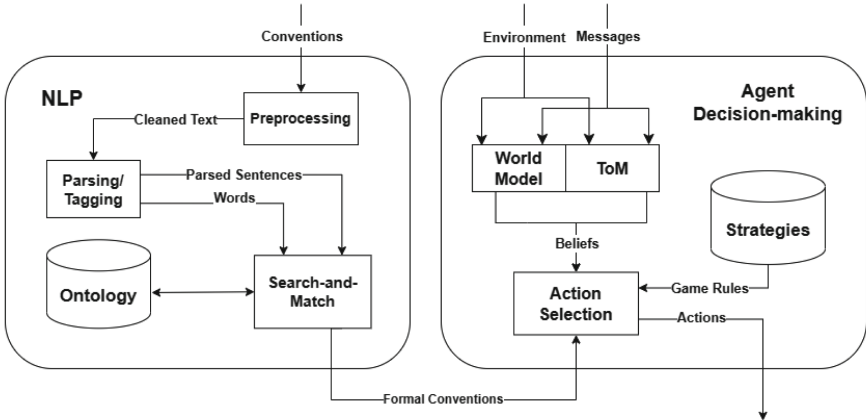


Fig. 1. The architecture for an agent in the convention-based cooperative multiagent system. Note that the “NLP” block works offline, meaning the processing will be done only once before the agent decision-making process occurs.

In the “NLP” block, we find the modules constituting the pipeline for processing natural language conventions and generating their formal representations. In the “Agent Decision-making” block, there are modules concerned with the agent’s knowledge of the world (World Model) and the module building explanations for the actions of others (Theory of Mind—ToM). These two modules contain the agent’s beliefs and methods to update them.

4.1 NLP

The “NLP” block’s goal is to translate natural language conventions into a machine-readable formalism so that the agent can adapt its strategies. As stated in Sect. 3.1, adapting an off-the-shelf translation system is not feasible as all previous systems we are aware of are not maintained anymore [11, 16]. Therefore, we must adapt some of the methods and ideas of these systems to develop our processing pipeline. This pipeline includes a preprocessing step, a parser (with a tagging/annotating step), a database for ontologies, and a novel algorithm that generates the formal representation of conventions.

The “Preprocessing” module, as discussed in Sect. 3.1, will contain the techniques that must be applied for a general norm extraction task. For instance, H-Group Conventions are published in HTML format, so the text needs to be

extracted from the HTML code for those conventions as input. We can either manually extract them or apply existing web content extraction tools such as GOOSE,⁸ which was reported with the best performance for English newspaper text [3].⁹

In the “Parsing/Tagging” module, existing parsers like Stanford Parser can be used to retrieve the semantic and syntactic features from the conventions. We can assume that the parsing output will neither contain a vast vocabulary nor a complex sentence structure. The terms in the conventions will thus refer to the limited ontology of a particular domain, e.g. cards, colours, or numbers in Hanabi, so the vocabulary is naturally tiny. Similarly, conventions are relatively simple rules to be understandable by the public, e.g. conventions in Table 1. Differently from legal norms, the conventions we aim at are thus straightforward. Given this simplicity, we consider first-order logic expressive enough to formalise conventions. More concretely, [34] proposed a representation language called “Agent Situation Language” (ASL) used to represent the rules of games. We will explore using this language, or an extension of it, to represent conventions, as conventions have a similar expressive power to game rules. We might explore using Jason as the interpreter of ASL since ASL is similar to Agentspeak [38], a programming language used to represent some Hanabi conventions in [33].

“Ontology” is an ontology database, which works closely with the “Search-and-Match” module for formal conventions’ generation. A simpler mechanism of “Search-and-Match” would be to analyse the frequent words from the input and their semantic and syntactic features, which need no ontology. For example, when considering the tokens extracted from a convention, the verb phrases (VP) are naturally mapped into predicates, and noun phrases (NP) are naturally mapped into predicate arguments. Therefore, a simple rule can be: from a sentence $S = NP V ADJ$ generate $V(NP, ADJ)$. For instance, from “The card colour is red”, we obtain the predicate instance $is(card_colour, red)$. A more complex rule can combine a constituency tree with a dependency tree. This combination will help determine which words should be placed together as predicates and arguments. A similar method is proposed in [28]: they wrote the predicate in a slightly different way to our proposal, as $dependency(governor, dependent)$, and created categories for different semantic rules. Nevertheless, ontologies are much richer structures that allow us to represent complex knowledge within a particular domain [5, 8, 31]. Thus, if there is an existing ontology in the particular domain of the conventions or a reference ontology, we can use it to determine the meaning of the words.¹⁰ Applying an existing ontology can also reduce the work of creating formal representations for conventions from scratch. Back to the sentence “The card colour is red”, instead of creating a rule to capture the

⁸ <https://github.com/goose3/goose3>.

⁹ Another tool with the best performance in [3] was Newspaper. Unlike GOOSE, it was primarily designed for newspaper texts and cannot extract structured text.

¹⁰ If no ontology exists, then we can generate one from the text using parsing and concepts/relation extraction rules, which consider semantic and syntactic features of the words [26].

features, we can search the ontology for concepts like “card” and “colour” and use the known relation between them, that is, cards have colours, and red is a common colour for cards. With these concepts and relations, we can formalise this sentence more efficiently. The intuitive meaning behind the name “Search-and-Match” is that the algorithm will use the input (words and tags in parsed sentences) as keywords to search the same (or similar) concepts in the “Ontology” database, match the keywords with the concepts, decide which relations can be found between the keywords, and generate the formal representation of the convention based on these relations.

However, some utterances, words or phrases represent features requiring formalising domain-dependent solid knowledge. This knowledge can be expressed as logical formulae patterns that include ontological elements from “Ontology”, and some of the features extracted from the text of the convention. The “Search-and-Match” module will first consult the “Ontology” with those utterances, words or phrases from its input, and try to match them with ontology concepts based on their semantic similarity (e.g. the semantic features they share or the similarity between their semantic features). If the existing ontology in the database does not contain these (domain-specific) concepts, this module should be able to update or enrich the ontology. For instance, consider the expression “right-most card” in Table 1. It is a relative concept: its meaning may refer to either a physical position (e.g. fifth slot in Hanabi) or a “chop” (see the example in Sect. 2). The ontology should be able to help select the second (logical) meaning as the actual meaning of the expression “right-most card”. Here is the example for *discard chop (right-most unclued) card* code that the ontology should provide:

```

if convention_concept(right_most_card) and
  next_option([discard(_,_,X),discard(_,_,Y),discard(_,_,Z)]) and
  clued(_,_,Z) and ~clued(_,_,Y) and ~clued(_,_,X) and X < Y
then next_action(discard(_,_,Y))

```

That in English would read like “if the convention text mentions the word `right_most_card`, the strategic component doubts about discarding one of three cards, and only one of them is clued, then the card to discard will be the one of the non-clued two that is in the right-most position.”

Such schemes can be grouped into structures representing the semantic similarity of the concepts. For instance, the code patterns for “right-most” and “left-most” concepts can be placed close to each other, possibly hanging from “position”, even if “position” does not appear in the conventions. For example, see the pattern associated with the concept of “position”. It contains a variable `OP` as a placeholder for an operator that can be later instantiated once we univocally determine the concrete position expressed in the convention:

```

[position(OP) for convention([position, place, location])]:=
if next_option([discard(_,_,X),discard(_,_,Y),discard(_,_,Z)]) and
  clued(_,_,Z) and ~clued(_,_,Y) and ~clued(_,_,X) and X OP Y

```

```
then next_action(discard(_,_,Y))
```

If a word in the convention (e.g. “right_most” or “right_place”) is found by the ontology as related with the concept “right_most_card”, the ontology can then provide the instance **position(<)**, or **position(>)** instead if the concept “left_most_card” is found. Such structure can be expressed in the language for formal conventions as:

```
[right_most_card(<) is_a position(OP)
  for convention([right_most, on_the_right, right_place])]
```

A complex ontology structure might have programming patterns for some of its concepts but not necessarily for all. If no programming patterns can be obtained from the ontology, a request for an update may be issued.

When importing an existing domain-independent ontology, we must particularise it to the context of an application, e.g. Hanabi, adding programming patterns. For instance, from a node in an ontology like “position,” we can add as leaves more domain-specific concepts (“left_most_card” and “right_most_card”) and associate them with the rules shown above.

In short, the “Search-and-Match” module performs two operations. On the one hand, it updates ontologies by adding concepts appearing in the conventions under analysis and associating formal convention programming patterns with them. This ontology can be enriched (e.g. by training a long short-term memory neural network for updating ontologies when new concepts are introduced [39]). On the other hand, it instantiates the parameters of existing formal convention programming schemes with particular values coming from the annotated words from the NLP conventions analysis.

4.2 World Model

The “World Model” is the module that represents knowledge about the environment. It contains Precepts, Domain-related Clauses, and Impossibility Clauses, all representing different kinds of information.

Factual information (facts) about the environment is represented as literals. The initialisation of an agent’s set of beliefs can thus be obtained from observing the environment. In our running example, these facts include the colour and number of cards other players hold. Though each agent does not have explicit knowledge of its cards, they can make inferences about the cards’ colour or number. The system’s possible states and actions are limited since there are just two types of clauses (about colour and number). “Domain-related” clauses indicate relationships among literals and specify more sophisticated characteristics; for example, a “playable” card must have a number which is one unit above the number on the card of the same colour that is on the stack on the table (e.g. red 3 is a “playable” card when there is a red 2 on the top of the red stack). “Impossibility clauses” express the circumstances where two literals cannot both be true. For example, the impossibility clause for “two different cards cannot occupy

the same slot” will have its condition become true when more than one card is assigned to the same slot.

4.3 ToM

The “ToM” module represents the beliefs that one agent has about the beliefs of others. It combines Theory of Mind clauses and Abducible Clauses. Theory of Mind clauses explicitly represent agents’ beliefs about facts of the environment. In contrast, Abducible clauses represent the possible beliefs they might have about facts of the environment.

The Theory of Mind Clauses are based on belief chains (see Sect. 3.2). Beliefs are encoded as literals of the type **believes**(**Ag**, **F**) in [33], which are true when the observer believes that an observee **Ag** is aware of a fact **F**. In our context, the concept of *fact* is equivalent to *belief* in the general definition of ToM because the observation of actions and the state of the environment are the only things we plan to use. As the environment is not fully observable (e.g. player’s cards are hidden from them), beliefs do not necessarily correspond to reality. This is so because agents will update beliefs via querying specific ToM clauses based on abduction, and abduction does not necessarily provide truthful consequences. A relevant source of belief updates is those abductive consequences that an agent *i* derives from the beliefs it holds about the beliefs of another agent *j* on *i*, that is, on itself. In that case, an agent becomes an observer of itself through the eyes of another agent, observer and observee simultaneously.

Example: Alice clues one of Bob’s cards, telling him its colour is red. When doing so, Alice can infer how Bob might interpret that piece of knowledge in terms of Bob’s beliefs about Alice’s beliefs leading to her telling Bob the colour of the card. For instance, Bob may infer that Alice is giving him a save clue (see Table 1) not to discard the card since this card is his current chop card.

Similarly to [33], we will leave out of our architecture any mechanism to determine beliefs about actions to be taken by other agents. We do so because of the high complexity of this kind of reasoning. However, we think a ToM representation with such capability would improve our agent architecture. We will consider it as future work.

Abducible Clauses complement the Theory of Mind Clauses. They add *potential* beliefs to the knowledge base as long as they do not contradict any pre-existing beliefs. Note that these clauses are domain-specific.

Example: Alice is currently holding the belief *I have a red or blue card in the third slot*. If Alice’s abduction mechanism produces *I have a red card in the third slot*, which is not contradictory to the current belief, she may (defeasibly) infer that the card’s colour is red and act accordingly.

4.4 Action Selection

The **SelectAction** function in [33] relies on *Action Selection Clauses* written in AgentSpeak. These clauses represent the actions the agents might take and the

beliefs they need to hold to take these actions. All the clauses also contain priority information encoded as a natural number. In [33], the game rules were manually coded as an environment in Java, while some conventions were manually programmed as Action Selection clauses. In the same work, the `SelectAction` function implements hard-coded strategies and takes Action Selection Clauses as input. After ranking the Action Selection Clauses based on their priorities, the function checks the clauses in order starting from the highest priority ones. If the clause’s body is true according to the beliefs the agent is holding, and the potentially abducible beliefs, then the action suggested by the clause is selected; the remaining clauses are not considered. The game rules further verify the feasibility of this selected action. If the verification fails, the agent will take a default action (which can be defined by the developer).

Our architecture will modify the hard-coded strategies described and implemented in the `SelectAction` function by a customisable component. Our “Action Selection” module will receive three kinds of input: the rules of the game from the “Strategies” database written in ASL, the Action Selection Clauses in [33] rewritten in ASL, and the other formal conventions, also written in ASL, from the “Search-and-Match” module. Note that the `SelectAction` function written in ASL might have a different structure than the one written in AgentSpeak.

5 Discussion and Future Work

This short paper presents the initial ideas for an NLP-based agent architecture capable of processing conventions expressed in natural language. We have illustrated the architecture using examples from the card game Hanabi. Our next objective is to implement an agent following this architecture and putting it to work playing with other agents. We will check if our NLP correctly interprets the conventions when our agent plays with other agents that have the conventions hardwired in their strategy.

First, We will adapt and extend the model in [33] with additional modules for NLP. There will likely be some modifications to the original model, such as for the `SelectAction` function. The game rules and some conventions were manually coded in the original architecture. Thus, that implementation will be our testbed against which we will test the correct workings of our NLP component. We will also consider modifying the language ASL proposed in [34]. For instance, as discussed in Sect. 2, manual annotation of rule priorities may not be the best solution. We will work with real-world conventions to determine whether an alternative approach for ranking is required. In addition, apart from the already mentioned H-Group Conventions, other sources of conventions might be used. An example can be conventions generated from non-natural language data (e.g. records of game playing [18, 23]). We are planning to use an existing dataset pairing natural language sentences with their first-order logic representation [27, 29]. Our objective is to have a pairing between conventions and their formal representation. However, as it is unlikely that we can have a large set of such pairings, because they require a lot of manual work and the number of conventions is not

very large, we would like to explore transfer learning techniques over the pairings in [27, 29]. We also plan to create a more general version of our architecture, adaptable to any convention-based cooperative scenario. Also, we plan to extend the architecture to formalise norms that restrict agents' behaviour, for instance, by limiting the set of available actions.

One of the challenges we face is the preprocessing of the conventions. Even though an engineer can create specific patterns to rewrite sentences into simplified English, we want to automate the procedure using existing tools. Submitting prompts to large-language models for generating simplified sentences or a rephrasing of the sentence can be one of the possible solutions. Another challenge is the reliability of the module's output, whether it represents the convention precisely enough for the system to process and for the agents to follow. In this case, we need a mechanism to generate several formal representations as candidates and select them based on their performance in the system.

Although there are many appealing research topics in the study of conventions, such as how to model reaction mechanisms to deal with agents that break conventions, norm emergence [35], or the dynamics of conventions during game-playing, we will limit the scope of our research to the topics mentioned in the previous paragraphs.

Acknowledgements. This research is conducted under the REDI Program, a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 101034328. This paper reflects only the author's view and the Research Executive Agency is not responsible for any use that may be made of the information it contains. This research also receives support from the VALAWAI project (Horizon #101070930) and the VAE project (Grant no. TED2021-131295B-C31) funded by MCIN/AEI/10.13039/501100011033.

References

1. Ågotnes, T., Van Der Hoek, W., Rodríguez-Aguilar, J.A., Sierra, C., Wooldridge, M.J.: On the logic of normative systems. In: IJCAI, vol. 7, pp. 1175–1180 (2007)
2. Balke, T., da Costa Pereira, C., Dignum, F., Lorini, E., Rotolo, A., Vasconcelos, W., Villata, S.: Norms in MAS: Definitions and Related Concepts. In: Andrighetto, G., Governatori, G., Noriega, P., van der Torre, L.W.N. (eds.) Normative Multi-Agent Systems, Dagstuhl Follow-Ups, vol. 4, pp. 1–31. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2013). <https://doi.org/10.4230/DFU.Vol4.12111.1>. <http://drops.dagstuhl.de/opus/volltexte/2013/3998>
3. Barbaresi, A., Lejeune, G.: Out-of-the-box and into the ditch? multilingual evaluation of generic text extraction tools. In: Language Resources and Evaluation Conference (LREC 2020), pp. 5–13 (2020)
4. Bard, N., Foerster, J.N., Chandar, S., Burch, N., Lanctot, M., Song, H.F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., Dunning, I., Mourad, S., Larochelle, H., Bellemare, M.G., Bowling, M.: The hanabi challenge: a new frontier for ai research. *Artif. Intell.* **280** (2020). <https://doi.org/10.1016/j.artint.2019.103216>

5. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. ScientificAmerican.com (2001)
6. Boella, G., Di Caro, L., Robaldo, L.: Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines. In: Theory, Practice, and Applications of Rules on the Web: 7th International Symposium, RuleML 2013, pp. 218–225. Springer (2013)
7. Canaan, R., Gao, X., Togelius, J., Nealen, A., Menzel, S.: Generating and adapting to diverse ad-hoc partners in hanabi. *IEEE Trans. Games* (2022)
8. Consortium, W.W.W.: Owl 2 Web Ontology Language Document Overview, 2nd edn. <https://www.w3.org/TR/owl2-overview/>
9. da Costa Pereira, C., Tettamanzi, A.G., Villata, S., Liao, B., Malerba, A., Rotolo, A., van Der Torre, L.: Handling norms in multi-agent system by means of formal argumentation. *J. Appl. Logics-IFCoLoG J. Logics Appl.* **4**(9), 1–35 (2017)
10. Cuzzolin, F., Morelli, A., Cirstea, B., Sahakian, B.J.: Knowing me, knowing you: theory of mind in ai. *Psychol. Med.* **50**(7), 1057–1061 (2020)
11. Dragoni, M., Villata, S., Rizzi, W., Governatori, G.: Combining nlp approaches for rule extraction from legal documents. In: 1st Workshop on MIning and REasoning with Legal texts (MIREL 2016) (2016)
12. Ferraro, G., Lam, H.P.: Nlp techniques for normative mining. *FLAP* **8**(4), 941–974 (2021)
13. Ferraro, G., Lam, H.P., Tosatto, S.C., Olivieri, F., Islam, M.B., van Beest, N., Governatori, G.: Automatic extraction of legal norms: Eevaluation of natural language processing tools. In: Sakamoto, M., Okazaki, N., Mineshima, K., Satoh, K. (eds.) *New Frontiers in Artificial Intelligence*. pp. 64–81. Springer International Publishing (2020)
14. Foerster, J., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., Bowling, M.: Bayesian action decoder for deep multi-agent reinforcement learning. In: *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 1942–1951. PMLR (2019)
15. Gao, X., Singh, M.P.: Extracting normative relationships from business contracts. In: *AAMAS*, pp. 101–108. Citeseer (2014)
16. Gaur, S., Vo, N.H., Kashihara, K., Baral, C.: Translating simple legal text to formal representations. In: *New Frontiers in Artificial Intelligence: JSAI-isAI 2014 Workshops*, pp. 259–273. Springer (2015)
17. Governatori, G., Rotolo, A.: A conceptually rich model of business process compliance. In: *Proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling*, vol. 110, pp. 3–12. Citeseer (2010)
18. Gray, J., Lerer, A., Bakhtin, A., Brown, N.: Human-level performance in no-press diplomacy via equilibrium search (2020). [arXiv:2010.02923](https://arxiv.org/abs/2010.02923)
19. Harsanyi, J.C.: Games with randomly disturbed payoffs: a new rationale for mixed-strategy equilibrium points. *Internat. J. Game Theory* **2**(1), 1–23 (1973)
20. Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., Silver, D.: Rainbow: combining improvements in deep reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* **32**(1) (2018). <https://doi.org/10.1609/aaai.v32i1.11796>. <https://ojs.aaai.org/index.php/AAAI/article/view/11796>
21. Hu, H., Lerer, A., Cui, B., Pineda, L., Brown, N., Foerster, J.: Off-belief learning. In: *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 4369–4379 (2021)

22. Hu, H., Lerer, A., Peysakhovich, A., Foerster, J.: “other-play” for zero-shot coordination. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 4399–4410 (2020)
23. Hu, H., Wu, D.J., Lerer, A., Foerster, J., Brown, N.: Human-ai coordination via human-regularized search and learning (2022). [arXiv:2210.05125](https://arxiv.org/abs/2210.05125)
24. Jacob, A.P., Wu, D.J., Farina, G., Lerer, A., Hu, H., Bakhtin, A., Andreas, J., Brown, N.: Modeling strong and human-like gameplay with kl-regularized search. In: Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 9695–9728 (2022)
25. Kaptein, R., Serdyukov, P., De Vries, A., Kamps, J.: Entity ranking using wikipedia as a pivot. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. p. 69–78. CIKM ’10, Association for Computing Machinery, New York, NY, USA (2010). <https://doi.org/10.1145/1871437.1871451>
26. Kumar, N., Kumar, M., Singh, M.: Automated ontology generation from a plain text using statistical and nlp techniques. *Int. J. Syst. Assur. Eng. Manag.* **7**, 282–293 (2016)
27. Levkovskiy, O., Li, W.: Generating predicate logic expressions from natural language. In: SoutheastCon 2021, pp. 1–8. IEEE (2021)
28. Longo, C.F., Longo, F., Santoro, C.: Caspar: Towards decision making helpers agents for iot, based on natural language and first order logic reasoning. *Eng. Appl. Artif. Intell.* **104**, 104269 (2021)
29. Lu, X., Liu, J., Gu, Z., Tong, H., Xie, C., Huang, J., Xiao, Y., Wang, W.: Parsing natural language into propositional and first-order logic with dual reinforcement learning. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 5419–5431 (2022)
30. Lupu, A., Cui, B., Hu, H., Foerster, J.: Trajectory diversity for zero-shot coordination. In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 7204–7213 (2021)
31. McGuinness, D.L., Van Harmelen, F., et al.: Owl web ontology language overview. <https://www.w3.org/TR/2004/REC-owl-features-20040210/>
32. Michel, M., Djurica, D., Mendling, J.: Identification of decision rules from legislative documents using machine learning and natural language processing. In: Proceedings of the 55th Hawaii International Conference on System Sciences, pp. 6247–6256 (2022)
33. Montes, N., Osman, N., Sierra, C.: Combining theory of mind and abduction for cooperation under imperfect information. In: Baumeister, D., Rothe, J. (eds.) *Multi-Agent Systems*, pp. 294–311. Springer International Publishing, Cham (2022)
34. Montes, N., Osman, N., Sierra, C.: A computational model of ostrom’s institutional analysis and development framework. *Artif. Intell.* **311**, 103756 (2022). <https://doi.org/10.1016/j.artint.2022.103756>
35. Morales, J., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A., Vasconcelos, W., Wooldridge, M.: Online automated synthesis of compact normative systems. *ACM Trans. Auton. Adapt. Syst. (TAAS)* **10**(1), 1–33 (2015)
36. Olson, T., Forbus, K.D.: Learning norms via natural language teachings (2022). [arXiv:abs/2201.10556](https://arxiv.org/abs/2201.10556). <https://api.semanticscholar.org/CorpusID:244305883>
37. Pehcevski, J., Vercoustre, A.M., Thom, J.A.: Exploiting locality of wikipedia links in entity ranking. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) *Advances in Information Retrieval*, pp. 258–269. Springer, Berlin, Heidelberg (2008)

38. Rao, A.S.: Agentspeak(1): Bdi agents speak out in a logical computable language. In: Van de Velde, W., Perram, J.W. (eds.) *Agents Breaking Away*, pp. 42–55. Springer, Berlin, Heidelberg (1996)
39. Sanagavarapu, L.M., Iyer, V., Reddy, R.: A deep learning approach for ontology enrichment from unstructured text (2021). [arXiv:2112.08554](https://arxiv.org/abs/2112.08554)
40. Shih, A., Sawhney, A., Kondic, J., Ermon, S., Sadigh, D.: On the critical role of conventions in adaptive human-ai collaboration (2021)
41. Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., Dann, J.: Automated extraction of semantic legal metadata using natural language processing. In: *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pp. 124–135 (2018). <https://doi.org/10.1109/RE.2018.00022>
42. Treutlein, J., Dennis, M., Oesterheld, C., Foerster, J.: A new formalism, method and open issues for zero-shot coordination (2023)
43. Tucker, M., Zhou, Y., Shah, J.: Adversarially guided self-play for adopting social conventions (2020)
44. de Weerd, H., Verbrugge, R., Verheij, B.: Higher-order social cognition in the game of rock-paper-scissors: A simulation study. In: Bonanno, G., van Ditmarsch, H., van der Hoek, W. (eds.) *Proceedings of the 10th Conference on Logic and the Foundations of Game and Decision Theory*, pp. 218–232 (2012)
45. de Weerd, H., Verheij, B.: The advantage of higher-order theory of mind in the game of limited bidding. In: van Eijck, J., Verbrugge, R. (eds.) *CEUR Workshop Proceedings*, vol. 751, pp. 149–164 (2011)
46. Wyner, A., Peters, W.: On rule extraction from regulations. In: *Legal knowledge and information systems*, pp. 113–122. IOS Press (2011)

Author Index

- Ajmeri, Nirav 118
Al Anaissy, Caren 141
Alvarez-Napagao, Sergio 95
- Baldoni, Matteo 55
Blackledge, Buster 37
- Charpenay, Victor 55
Ciortea, Andrei 55
Collins, Daniel E. 118
Cranefield, Stephen 21, 55
d'Inverno, Mark 77
- Dennis, Louise 161
Di Scala, Daan 3
- Fraga Pereira, Ramon 161
- Gnatyshak, Dmitry 95
- Houghton, Conor 118
- Kolker, Simon 161
- Le Renard, Noan 37
- Marin Gutierrez, David 95
Masoud, Hazem 37
Mertzani, Asimina 37
- Nevejans, Nathalie 141
Noriega, Pablo 77
- Padget, Julian 55, 77
Pan, Shuxian 182
Papaioikonomou, Antonios 37
Pitt, Jeremy 21, 37
- Scott, Matthew 37
Sengupta, Abira 21
Sierra, Carles 182
Singh, Munindar P. 55
- Vázquez-Salceda, Javier 95
Verhagen, Harko 77
Vesic, Srdjan 141
- Xu, Mengwei 161
- Yolum, Pinar 3