Agent-directed Runtime Norm Synthesis

Andreasa Morris-Martin
University of Bath
Bath, UK
a.l.morris.martin@bath.ac.uk

Marina De Vos
University of Bath
Bath, UK
m.d.vos@bath.ac.uk

Julian Padget
University of Bath
Bath, UK
j.a.padget@bath.ac.uk

Oliver Ray
University of Bristol
Bristol, UK
csxor@bristol.ac.uk

ABSTRACT

To maintain fitness-for-purpose, the set of norms governing a MAS will typically need to evolve to reflect the changing needs of both participants and the environment. We put forward a conceptual framework to address this problem comprising dynamic institutions (sets of norms), that depend upon the formulation of new norms and the revision of existing norms, informed by the experiences of agents participating in the MAS. The objective is to allow participating agents to influence the revision of the norms governing the MAS, thereby taking a first step towards adaptable self-governance of socio-technical systems through explicit norms. This paper proposes a novel framework for revising at runtime the norms of a formally specified institution, directed by the agents in the MAS. The framework employs special-purpose synthesiser agents with partial observability of the state of the MAS to formulate new norms or revise existing ones, in response to requests from agents for changes to the institution. To demonstrate the feasibility of the framework, we capture a set of norms using the Inst*AL* institutional specification language and revise those norms using the XHAIL symbolic machine learning system. Building freely on Sergot's room scenario as a case study, we show how to synthesise norms that can resolve runtime institutional conflicts, and so establish the viability of a method for decentralised agent-directed runtime (online) revision of explicit norms.

KEYWORDS

norms; multiagent systems; institution; norm synthesis; decentralised; agent-direct; runtime; self governance

ACM Reference Format:
Andreasa Morris-Martin, Marina De Vos, Julian Padget, and Oliver Ray. 2023. Agent-directed Runtime Norm Synthesis. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Normative multiagent systems (MAS) provide a set of norms, referred to as an institution or normative system, that govern the behaviour of agents within the MAS. Normative MAS extend conventional MAS by incorporating an explicit representation of the normative information [12]. An institution needs stability to facilitate continued governance of these agents over time, however, if institutions remain static for longer than necessary, the norms comprising them may become partly or wholly irrelevant. In order to remain fit for purpose, the norms of the institution must evolve to reflect the changing needs of their environment and its participants. Acknowledging that norms need to change with the environment to remain relevant, we note that synthesis of norms at runtime is being studied more extensively [25]. In consequence, we put forward a conceptual framework to allow for dynamic institutions, that depend upon the formulation of new norms and the revision of existing norms, informed by the experiences of agents participating in the MAS. The objective is to allow participating agents to influence the revision of the norms governing them and the wider MAS, so taking a first step towards adaptable self-governance of socio-technical systems through explicit norms.

Our framework, which we coin agent-directed norm synthesis, is in line with two of Ostrom's eight principles for institutional design: participatory decision-making and defining rules that meet the needs of the local context. Hence, we allow the agents that operate within the MAS to be involved in the synthesis of norms that govern the MAS. The framework employs special-purpose synthesiser agents with partial observability of the state of the MAS to formulate new norms or revise existing ones, in response to requests from agents for changes to the formally specified institution. A participating software agent initiates this revision by requesting a change based on its (unsatisfactory) experience, which is used by the system to revise, if possible, its current set of norms to give the agent its desired outcome.

The contributions of this paper are two-fold. Firstly, we propose an agent-directed norm synthesis framework that based on requests and interactions of agents within the MAS, can synthesise norms to resolve conflicts reported by the institution. Secondly, we demonstrate how we utilise decentralised synthesisers at runtime to revise the norms of an institution through symbolic machine learning. To demonstrate the feasibility of the framework, we capture a set of norms using the Inst*AL* institutional specification language. We define rules that allow our institution to report on conflict situations that arise in the MAS. We then prove a mechanism to revise those norms at runtime using the XHAIL symbolic machine learning system using the agent experience data. This results in a revised Inst*AL* specification that resolves the conflict.

Building freely on Sergot's room scenario as a case study [33], we show how to synthesise norms that can resolve conflicts reported

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

by the institution at runtime. Additionally, we ensure that the modifications to the institution are acceptable for all the agents within the MAS and are not in conflict with the overall objectives of the MAS. Hence, we establish the viability of a method for decentralised agent-directed runtime (online) revision of explicit norms.

2 MOTIVATION

In practice, institutions governing MAS are typically static and manually updated by a system designer who must examine runs or traces of the MAS and recommend revisions of the norms based on problematic situations observed. This is a tedious and potentially error-prone process which relies solely on the designer to be able to recognise the problems in presumably large textual logs. There are numerous computational attempts at revising the norms in a MAS, for example [5, 8, 20–24], but each of these only consider a small number of deontic notions of norms at a time, less than five, with the majority only considering a single deontic notion. Other approaches to computational revision employ a method which selects a suitable subset of norms to be active concurrently from a set of static norms [7, 13, 34].

All of the approaches above synthesise norms in isolation, where they consider a single situation and a deontic concept that defines the behaviour of an agent(s) in that situation. In contrast, our framework considers a rule or the set of rules constituting a holistic institution and traces of a desired or undesired run. A further difference is that the norm change mechanism in the works cited above is typically syntactic rather than semantic [9]. With syntactic change, synthesis adds new norms or removes old norms resulting in a new set of deontic concepts. In contrast, semantic norm change affects counts-as rules or makes changes in the granting and revoking of deontic concepts. [6, 18, 19] also employ semantic change in their revision of a holistic institution, however different from our framework, they aim to identify and resolve conflicting rules between interacting institutions at design time and require manual involvement in the process.

Available norm synthesis methods revise the norms only to ensure that the norms continue to facilitate system goals even if that means preventing individual agents from accomplishing their goals. We believe there is no reason that individual agents should not be involved in the governance of the system and revisions should be possible for the benefit of the individual agents and their goals while ensuring that the revisions do not conflict with system goals. Agents should be able to direct or at least influence the norm synthesis. This paper's contribution is to provide the first framework for holistic and semantic norm synthesis directed by the agents themselves. We present an implementation to provide a proof of concept that this approach works.

3 A NORM SYNTHESIS FRAMEWORK

The work reported here follows the norm emergence framework set out in Morris-Martin et al. [26]. To make this paper self-contained, we summarise the key points and introduce the terminology used in what follows. The cited paper describes a four-stage norm-emergence framework (creation, propagation, adoption, emergence), where creation – the focus of this paper – comprises three stages (ideation, synthesis, decision), within which synthesis further comprises four stages (proposal, synthesis, deliberation, agreement) with two types of actors: participating agents and synthesiser agents. Each synthesiser agent is responsible for a set of participating agents and receives their revision requests. The synthesiser only has access to information pertaining to the agents it looks after, so may not have complete information. The synthesiser agents form a fully-connected network. In addition, there is an "Oracle" that provides a system and functional boundary for the revision process to which to appeal for final approval.

All of the examples of (prescriptive) norm revision we have found in the literature depend upon some external agency that observes the system and revises the norms to optimise them for the system's goals. In contrast, the novelty here is that it is the agents that identify the need for change and then delegate that task to synthesiser agents to fulfil both the coordination of multiple requests (some of which may be for the same thing), establish consensus about a particular revision and handle the mechanical details of the revision process. For the purposes of the experiments in section 5, the non-synthesiser agents are norm-compliant – that is they do not break norms to achieve their goals as could normally be the case in a MAS – then when they discover something they want to do but cannot, they report that information to initiate the revision request. The framework works with richer agent behaviours and sanctioning mechanisms. The norm-compliant agents used here are sufficient for the purpose of illustrating how revision information is captured and the revision task is brought about. Figure 1 captures the interaction and flow of information between the agent, synthesiser and revision mechanism to support the examination of the three stages of the creation process that now follows.

3.1 Ideation

Participating agents interact in the environment and have information about undesirable states that could result from their actions. Such information is built into their plans and percept handling mechanisms to inform their behaviours making them normative by design. They aim to maintain desirable institutional states and avoid undesirable ones, therefore these agents will attempt an action to resolve a detected problem and also report the situation to synthesisers. In the present system, agents detect a state that is undesirable and naively conclude that this unwanted state is a direct result of problems or inadequacies within the normative institution. They also undo their action with the goal that doing so will resolve the problem. While we do not hold to that, it is sufficient for our purposes of triggering a revision. More sophisticated analysis is a matter for (much) further work, but its outcome would be the same without loss of generality. Thus, the agent submits a request for a change or revision of the normative institution. The agent's goal or intention behind these requests is to ensure that their necessary actions always result in acceptable institutional states as they are norm-compliant by design, for reasons given earlier.

3.2 Norm Synthesis

We now describe each of the sub-phases of the norm synthesis phase to show how the activities together achieve agent-directed norm synthesis.

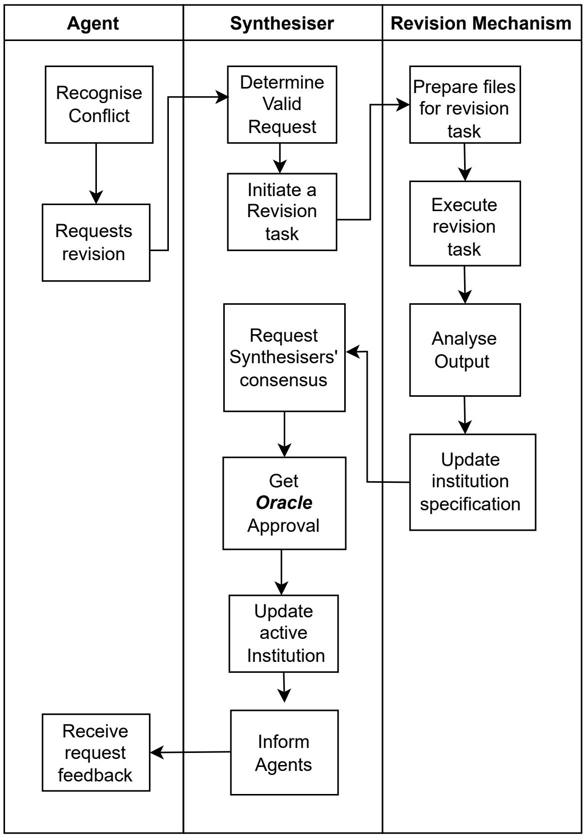

Figure 1: A summary of the agent-directed norm synthesis framework showing interactions between the main agents and the revision mechanism.

3.2.1 Proposal. At this stage, the participating agent sends a request to its synthesiser agent. Synthesisers handle requests that follow a defined message format where the agent provides the action attempted, the actual result of the action and the expected result, with the reason why this was expected. Upon receipt of a request, synthesisers apply a filtering mechanism to handle requests that do not warrant any action by the synthesiser, e.g. if the agent is attempting to report a problem that cannot or should not be resolved in the institution. In this case, the sending agent is advised that their request cannot be handled. Otherwise, the synthesiser will attempt either to handle the request, or queue it for later, if it is currently handling another. To handle a request, synthesisers begin a revision task according to the Synthesis stage that follows.

3.2.2 Synthesis. The framework in [26] sets out the requirements for revision guided by agent experiences but leaves open how the function shall be realised. Since that is mostly an implementation matter, we defer the details to section 4. The essential idea is the proposal information supplied by the agent is used to revise the institution without introducing conflicts with existing norms; that is the revision shall not affect anything other than making compliant the sequence of actions given in the proposal. The effect of the revision process is either to create a fresh institution that respects the proposal or no change, which latter means there is no revision that resolves the problem identified in the proposal. We draw attention to the fact that each synthesiser only has partial observability through the agents it looks after. Thus they are at risk of being unable to make decisions due to lack of information or of making a locally correct decision for the agents that communicate with them but a decision which is not correct for the system as a whole. This is discussed further in later sections. The number of synthesiser agents used is an implementation choice but there may be a performance impact due to how agents' behaviours are implemented.

3.2.3 Deliberation. If the revision process can synthesise a norm or revise the existing norms in the institution, the synthesiser performing the task must request that the remaining synthesisers determine whether they approve of the revision of the MAS. Synthesisers accept or reject a revision based on whether the revision allows the agents they are responsible for to successfully perform the action associated with the revision request. To answer this question, each of the other synthesisers query the institution for the effect that the action by one of its agents will have on the state where the action is executed using the proposed institution. The resulting state is examined to ensure the agent is able to complete the action successfully. If so, the synthesiser accepts the revision, otherwise it is rejected and the result notified to the originating synthesiser.

3.2.4 Agreement. The synthesiser that originates the revision determines whether to forward the proposed revision to the Oracle for a final decision on the basis of a majority vote of the other synthesisers. A positive vote signals the end of the norm synthesis phase but not the revision task. In the case of a negative vote, the synthesiser informs the requesting agent(s) that the revision was not successful. Once again, we have adopted a simple solution, for sake of clarity, but a more sophisticated consensus mechanism could be substituted.

3.3 Decision

The Oracle is represented as an agent and makes the final decision whether the proposed institution can replace the old one or not. This decision is intended to ensure that revisions proposed do not affect the objectives of the MAS, introduce conflicts or allow unwanted situations in the MAS. The Oracle agent achieves this by consulting a record of revisions that it will not accept. The acceptance of a revision, and the feedback to the synthesiser who made the request, marks the successful completion of a revision task. Similarly, the non-acceptance and accompanying feedback to the synthesiser marks the unsuccessful completion of a revision task.

4 IMPLEMENTATION

In order to demonstrate the feasibility of the framework described in the previous section, we instantiate it with Inst*AL*, the institutional specification language, for representing a set of norms and XHAIL, a symbolic machine learning system, for norm revision. Both use answer set programming as their base language. We implement a JASON project which incorporates the agents, institution and revision system in a MAS implementation. We discuss the tools used for this proof of concept implementation below.

4.1 Inst*AL*

Inst*AL* [29] is an institutional action language that is based on Event Calculus [17] and Action Language constructs, thereby exhibiting several attributes of both. Inst*AL* uses answer set programming (AnsProlog, ASP)[2] as its computational back-end where the Inst*AL* specification is translated into an ASP program along with additional grounding information, such as time points, values that particular literals may take, and agent actions, then the solver generates all traces of a given length that include all the actions provided. The Inst*AL* model is based on the concept of observable events which capture what occurs in the physical or virtual world and generates institutional events which then have an effect on the domain of the institution in question if necessary. An institution in Inst*AL* is modelled as a set of institutional states that evolve over time as a direct result of the occurrence of institutional events. Subsequently the state of an institution is defined as the set of fluents or facts that describe the institutional state at a given time instant.

We define our institution to recognise and report conflict situations that fully compliant normative agents would prefer to avoid. These indicate a state within the institution that the MAS prefers to not occur, but does not restrict the state from occurring through regimentation. Inst*AL*, as originally defined, uses an event-based normative model, i.e unwanted behaviour generates so-called violation events that can be used for enforcement. In defining state-based conflict situations that can be recognised and reported by the institution, we extend Inst*AL* to represent state-based norms by means of the non-inertial fluent construct already available in Inst*AL*. Non-inertial fluents present normative information to agents in the time step that the conditions for the (non-inertial) fluent are satisfied. These state norms are defined as a non-inertial rule that says that one fluent holds when the conjunction of a set of fluent(s) also hold, some of which may be negated. Multiple such definitions may be used to capture disjunctive conditions. Naturally, the fluent may not directly or indirectly depend on itself. We use these state-based norms to define normative information that represent undesirable states from the perspective of the institution and the agent.

4.2 XHAIL

We utilise Inductive Logic Programming (ILP), a symbolic machine learning technique, to revise the institution based on the experience of the agents requesting a change. ILP allows for theory revision by taking a modifiable non-monotonic theory and positive and negative examples as input and returns a set of possible revisions of the theory, called hypotheses, that satisfy the examples. We use XHAIL [30] as our ILP solver. XHAIL enables the learning and revision of clauses in normal logic programs [30]. In XHAIL, the learning approach is guided by user-specified language constraints (mode-declarations) and search biases to revise non-monotonic logic theories to make them consistent with a set of positive and negative examples in the presence of incomplete information.

In this setting, the institution is our revisable theory and a trace is used to produce the examples. The trace, provided by a requesting agent is at most seven (7) states long containing the state that the action took place and three (3) states before and after if available. The initiation of a revision task (See Figure 1, top right box) initiates the automatic generation of the input files for XHAIL. From the trace, we extract the relevant positive and negative facts about the institution that describe what we wish to see or not see; these are our positive and negative examples, respectively. We use the textual representation of the institution to extract mode-declarations for fluents/events that are related to the action involved in the request being performed. Custom mode-declarations allows us to constrain the revision process by imposing syntactic and semantic bias on the proposed hypothesis. A similar process though requiring manual involvement is implemented in [6, 18, 19].

After the execution of the ILP task and analysis of the output, there is either a revised Inst*AL* file for the synthesiser to propose as the new institution or an empty file, indicating that the revision task is complete but no revision can be determined. We visualise the revision of an institution defined in Inst*AL* in Figure 2.

4.3 JASON

We implement our MAS as a JASON [4] project. JASON is an agent development language which is an implementation of AgentSpeak that facilitates the development of custom environments in Java and belief-desire-intentions (BDI) agents which participate within the environments. JASON also provides the architecture supporting the execution of these agents in a simulation environment. We implement the environment for our MAS utilising a custom environment class called StepSynchedEnvironment which extends the TimeSteppedEnvironment provided by JASON with additional capabilities for handling durative tasks. The TimeSteppedEnvironment available from JASON allows all agents' tasks to complete within a single timestep, irrespective of the task length. Durative tasks are able to extend across multiple timesteps.

The environment designed for our MAS provides the following facilities: (i) Institution integration: the environment allows for the execution of participating agent's actions in the institution and the transition to a new institution state. (ii) State storage: the details of the facts held in a state and the events observed and occurred in that state are stored in a data structure to provide trace information for synthesis tasks. (iii) Analysis of institutional output: following every participating agent's action on the institution, the output of the new state is analysed and the percepts provided to the acting agent. (iv) Preparation of revision files: the environment interacts with a utility class that is able to prepare the files needed for each revision task. (v) Revision: all of the functionality associated with revision tasks, including the execution of an ILP task in XHAIL. (vi) Recording of institutional revisions: the environment records the timestep and the updated institution file whenever the institution is changed. (vii) Execution of actions by agents: the set of tasks that the agents within a MAS can execute and the effect that those actions have on the environment.

5 EXPERIMENTS AND RESULTS

5.1 Case Study

We define a case study inspired by Sergot's room scenario [33]. We simulate agents utilising a set of disconnected rooms in a building where agents can enter and leave rooms once they have the necessary permissions to do so. Rooms have a maximum capacity and some rooms have special purposes, e.g. they only allow access to certain types of agents or agents performing a certain role. In

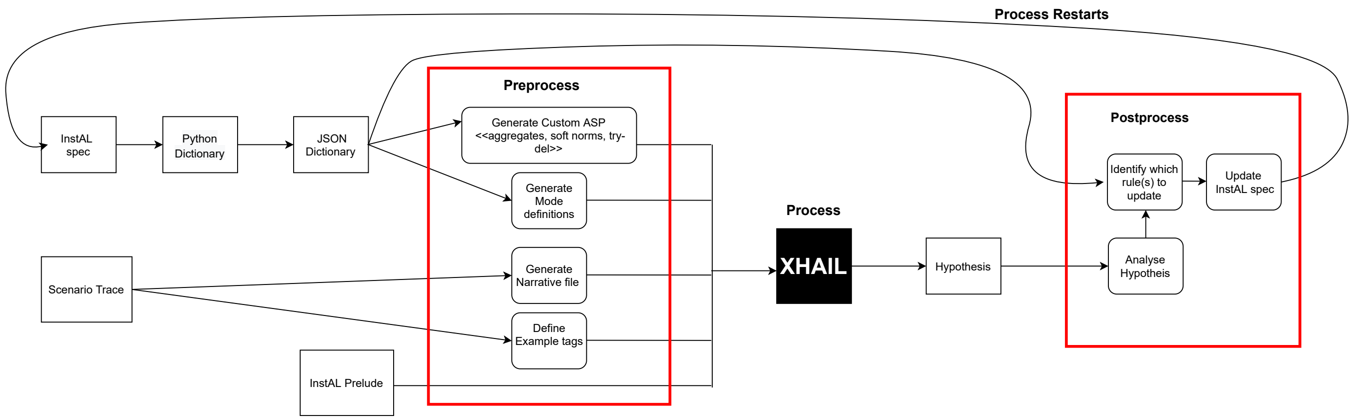

Figure 2: The implementation of the revision process of an Inst*AL* specification in XHAIL using agent experience data

our institution, agents have permission to enter rooms once they enter the MAS. Once in a room, the agent loses permission to enter all rooms and gains permission to leave the room that they are currently in. The "rooms" institution allows us to investigate several scenarios allowing for modifications to the existing institution, based on how agents use the MAS. These scenarios are described in the experiments discussed below.

For the purposes of this paper, we present base agents who are the primary actors in the MAS. We have done experiments with agents with other characteristics and with mixed populations, but for the revision functions explored here the base agents are sufficient. The task of a base agent is to explore the rooms in the institution, therefore it will randomly select one of the available rooms to enter first. It attempts to enter this room, by executing an enter room action. Base agents are fully norm compliant and always report problem situations they perceive, then act to resolve those problems as they naively conclude that this unwanted state is a direct result of problems or inadequacies within the normative institution brought about by their last action. In our case study, any violation perceived in a room counts-as an unwanted situation, which the base agent will act to resolve by leaving the room and reporting to the synthesiser. An agent attempting to enter a room can observe one of three possible outcomes: (i) the agent is allowed to enter the room and there are no issues reported, (ii) the agent is allowed to enter the room and but there are additional issues reported, and (iii) the agent is prevented from entering the room, and a `deniedEntry` event is triggered. The agent reports to its synthesiser in the case of either the last two possible outcomes.

5.2 Experiments

We conduct experiments to determine whether it is feasible to utilise the partial knowledge of synthesisers for a revision versus using global knowledge for our proof of concept. We define global knowledge as all the data about institutional states for the selected seven timesteps in the supplied trace, inclusive of data about all the agents in the MAS and all domain facts (fluents). Partial knowledge still includes all domain facts but then only those institutional states in which the actor supervised by the synthesiser is active and then only data about those agents supervised by the (same) synthesiser, that is the reporting agent's synthesiser siblings.

Initially, we experimented with global knowledge to drive the revision task, but it clearly contains much superfluous information, whereas the partial knowledge has much less that is not germane to the situation, making the revision process considerably faster. Having established that using partial mostly gives the same results as global (see section 7.1 for further discussion of this issue), the experiments reported here only use partial knowledge, but the full experimental record and implementation set up are available on GitHub [1].

Below we demonstrate, using scenarios from the case study, how agent-directed norm synthesis is able to provide solutions to problems encountered in the MAS. We show that the revision mechanism is able correctly to (i) remove unnecessary rules in the the institution, (ii) add missing but necessary rules, (iii) add an exception to an existing rule and (iv) handle two problems situations in one revision, one being the removal of a conflicting clause in a valid rule and the removal of a rule. The above are the only possible revisions of a set of norms: specialisation (add rule conditions), generalisation (remove rule conditions), addition and deletion of a rule, which the following sections now address.

5.2.1 Experiment 1: Removing a rule. We pose a situation whereby the institution penalises early arrivals: the first agent that enters a room receives an early bird violation. The aim of this experiment is to see whether agents are able to detect and report the early bird problem and whether the revision mechanism is able to recommend the removal of this rule. The simulation has two rooms: room1 and room2 both of whose maximum capacity is eight (8). We utilise the following agents: (i) two (2) synthesiser agents (ii) one coordinator agent (iii) one oracle agent, and (iv) eight (8) base agents. The simulation is run 8 times.

The results show, as expected, that for every experiment, a large proportion of agents are able to complete their goals, which is to explore both rooms, without requiring a change to the institution,

[1]The implementation set up is available at https://github.com/instsuite/agent-directed-norm-synthesis and https://github.com/andreasam-87/agent-directed-norm-synthesis the latter also has the experimental record

since at most two (2) agents at a time would receive this feedback. The agents who do receive the early bird violation feedback, leave the room, submit a request and await a response from a synthesiser agent before attempting to complete their goals. For this revision, as we have not provided an exception case, the only logical solution is to remove the rule, which is what we aim to achieve, as shown in the code snippet:

```
1 %Old Rule below
2 earlyBirdViol(P,L) when occupancy(L,X), equal(X,1),
      in_room(P,L), revise;
3 %New Rule below
4 %%% DELETED RULE %%%
```

where line 4 is output by the revision task to indicate the action it has taken. Six out of the eight experiments saw a successful revision task whereby the proposed revision was the removal of the rule, which was accepted by the Oracle and the remaining agents were able to complete their goals. Both of the remaining experiments, saw the revision task unable to compute a solution owing to insufficient examples provided, in turn due to the use of partial knowledge. This is a manifestation of the "mostly" issue noted in the introduction to this section and discussed further in section 7.1.

5.2.2 *Experiment 2: Adding a rule.* The aim of this experiment is to demonstrate the ability for the revision mechanism to add a rule that is necessary for the proper functioning of the institution, but which has either been deleted or just not yet written. To do so, we remove the rule that initiates the fluent `in_room` after an agent successfully enters a room, signalled by the occurrence of the arrive event. The `in_room` percept is used by other rules within the institution so its absence from the state will affect the validity of the institutional state.

As before, we investigate whether agents within the MAS are able to detect and report this problem and whether the revision mechanism is able to recommend the addition of a rule to the institution as shown in the code snippet below. We utilise the same experiment set up as with Experiment 1.

```
1 %Old Rule below
2
3 %New Rule below
4 arrive(P,L) initiates in_room(P,L) if revise; %%%%NEW
      RULE ADDED%%%%
```

where line 2 is empty because there was no old rule and line 4 shows the new rule. A total of 8 experiments were conducted, for 5 out of the 8 experiments, we observe that at least one revision task is able to successfully synthesis the norm shown in the code snippet above. Every agent continues to re-encounter this problem until the institution change occurs. Following the institutional change, all the agents are able to fulfil their goals. The remaining 3 experiments had to be manually terminated because some revision tasks did not complete due to insufficient examples which we discuss further in Section 7.1.

5.2.3 *Experiment 3: Adding an exception to a rule.* We investigate a scenario relating to exceeding the capacity of a room where rooms have a specified capacity, but this is not enforced, in that agents can still enter the room after this limit has been reached. In a physical sense allowing agents to enter a room after it has reached its capacity may seem impractical and encroaching on safety regulations. However the COVID-19 pandemic of 2020 and the following years, where capacities within a location vary based on the current restrictions, and which can potentially be ignored in special circumstances, makes this more realistic. We simulate the exceeded room capacity scenario to demonstrate the ability of our revision mechanism to add an exception to an existing rule. The institution currently active within the MAS will enable some of these agents to complete their goals before there is a need to change the institution but this is something that cannot be guaranteed for all agents. It depends entirely on the state of the MAS at the time of an agent's action (how many agents are currently in a room) and as a consequence, agents must hope that their entry into the room is preceded by a state that allows them to complete their goal. However some agents will be unable to complete their goals until the institution changes allowing more agents to be in the room at the same time because of a meeting occurring. The meeting being held in that room acts as the exception to the rule that must be learnt. As before, we investigate whether agents within the MAS are able to detect and report this problem and whether our revision mechanism is able to recommend the addition of this clause to the appropriate rule in the institution as shown in the snippet below. We utilise the same experiment set up as above except for the following change. The maximum capacity of the two rooms are 2 and 1 respectively, the small value ensures that the unwanted situation will arise early within the MAS given that a limited number of agents are used. A total of 8 experiments were conducted and for all 8 experiments, at least one revision task proposed the rule exception as shown below.

```
%Old Rule below
capacityExceededViol(L) when occupancy(L,X), max(L,Y),
    bigger(X,Y), revise;
%New Rule below
capacityExceededViol(L) when occupancy(L,X), max(L,Y),
    bigger(X,Y), revise, not meeting(L);  %%% ADDITION
    %%%
```

As expected, a subset of the agents are able to complete their goals without requiring a change to the institution depending on the room occupancy at the timestep when they attempt to enter. The remaining agents enter the room and receive the feedback, after which they leave the room and request a revision. The agents then await feedback from synthesisers before being able to complete their goals. In each run, there was at least one successful revision task which resulted in the appropriate change to the institution. We note that some revision tasks propose a revision to delete the rule which is rejected by the Oracle, such a proposal is because of insufficient examples provided as input (again, see section 7.1 for further discussion). Agents still attempt their goals again, allowing some to complete their goals while others re-encounter the problematic situation. In the end, all remaining agents are able to successfully complete their goals after the institution changes.

5.2.4 *Experiment 4: Multiple problems in one revision task.* We demonstrate how the revision mechanism can handle multiple revisions to multiple rules in an institution by showing a rule deletion and the removal of a clause from a valid rule in one revision. We build on the early bird scenario defined earlier and in addition introduce an impossible condition into an existing rule, as we now describe. The institution has a rule which terminates the permissions to enter the room that the agent has successfully entered.

We have added the clause permExit(P,L) to the rule which will never be true when an agent is entering a room and as a consequence the rule will not trigger. Therefore, agents will be in a room and still have the permissions to enter the room which we do not want to happen. As before, we investigate whether agents within the MAS are able to detect and report this problem and whether our revision mechanism is able to recommend the removal of the early bird violation rule and the removal of the unwanted clause, permExit(P,L), from the existing rule of the institution as shown in the snippet below. We utilise the same experiment set up in the other experiments and run the experiment 8 times.

```
%Old Rules below
earlyBirdViol(P,L) when occupancy(L,X), equal(X,1),
     in_room(P,L), revise;
arrive(P,L) terminates perm(enter(P,L)) if permExit(P,L),
      revise;
%New Rules below
%%% DELETED RULE %%%
arrive(P,L) terminates perm(enter(P,L)) if revise; %%%
     DELETION %%%
```

For the purposes of the experiment, to ensure that agents observe both problems together and report them in a single request, we modify our base agents so that only the agents who first enter a room are able to observe the additional permissions. Results show that most agents are able to complete their goals without requiring a change to the institution since only a small subset of agents will recognise the problematic situation. For every revision task in the 8 simulation runs, every revision task resulted in successful revisions. After the implementation of the revisions, the agents with incomplete goals are able to complete their goals.

6 RELATED WORK

Simulation models of norm emergence[27, 32, 35] typically assume the evolution of norms in a MAS. However, although the norms are introduced or learnt via agent interactions, they do not become part of an explicit normative system. As such they are different from the normative system in a normative MAS which is the focus of this paper. Boella et al. [3] mathematically demonstrate how an explicitly defined normative system can change when a new norm is added to the normative system and also explore the revision of norms but it is unclear whether an implementation is available.

Dynamic normative MAS that employ online norm synthesis mechanisms are becoming more common. For example, Mashayekhi et al. [20] utilise a centralised mechanism that determines the appropriate norms for the system over time, although the information they use and the mechanisms are different. Silk, the centralised mechanism of Mashayekhi et al. [20], monitors participating agents' interactions and recommends norms to resolve potential conflicts, and unwanted situations, identified by the system.

Ghorbani and Bravo [10], Ghorbani et al. [11] present a decentralised online norm synthesis mechanism that incorporates into a normative system the most popular strategy among those submitted by individual agents. Similarly, Riveret et al. [31] allows each agent to submit a norm that has been constructed internally from their learning experiences. The most common submission becomes a motion which is then voted upon by all the agents. The decentralised synthesis and voting mechanism is similar to work described in Morris-Martin et al. [26] and Campos et al. [5] however the synthesis in [5, 26] is delegated to special-purpose agents that process individual agents' experiences, rather than expecting each agent to have the capability to synthesize.

Dell'Anna et al. [7, 8] also discuss runtime norm revision, utilising a Bayesian learning network to analyse norm satisfaction/violation and the achievement of system objectives. Note that here, the goal is system optimisation, rather than individual agent satisfaction. The approach in Dell'Anna et al. [7] selects an appropriate norm set, 3 norms, from a collection of norm sets or can put together a new set from among the set of norms when no existing set ensures that norm satisfaction is in line with the achievement of system objectives. Subsequently, Dell'Anna et al. [8] discuss an approach that enforces and revises norms by revising the sanctions associated with norms to ensure system objectives are being met.

Knobbout et al. [15, 16] present a methodology to reason about and represent changes in a normative system by analysing how the introduction of a norm affects the system before and after an action is performed, but synthesis of the norm itself is not the focus here.

Huang et al. [13] present 'dynamic normative systems' where there are several possible norm sets applicable to a MAS and the active set can change over time depending on the context/situation, agents must recognise which norm set is active at any given time. While this clearly permits update of the normative system over time, it appears to depend on selection of and switching between pre-defined norm sets, rather than the synthesis of a revision in response to actions that have occurred.

SENSE [24] employs an offline centralised approach that resolves a conflict situation by considering the agent actions that occurred in the time-step before the conflict occurs (cf. the fixed length history of [1]). It then synthesises a norm by prohibiting the action of any one of the participating agents. The apparent weakness of this approach is the short time window and the limited context that feed into the revision. This is similar to earlier online approaches in IRON [21, 23] but which do not have the capacity to reason about the interdependence of norms and fail to synthesise an evolutionary stable normative system [24].

One final point in respect of the works discussed above, is their focus on individual norms. In some cases there is consideration of the impact of a norm on system goals, which implicitly addresses a norm's contribution in conjunction with others, but there is no consideration of the normative specification as a whole, in contrast to what we have presented here. To our knowledge only [6, 18, 19] consider the revision of an holistic normative system that not only contains the deontic notions of norms but also the rules that define state transitions and how the norms change based on agent actions over time. There are however a number of difference between their work and ours. Li et al. [18, 19] identifies how to revise the ASP representation of an institution to resolve conflicts between interacting institutions, but the process is not automatic and does not feed back to the Inst*AL* specification. The goal in Corapi et al. [6] is to revise norms so they better satisfy the provided traces, in order to identify incorrect or absent norms. There is a human-in-the-loop and because the process is initiated from the ASP representation, it lacks the automatic generation of the mode declarations informed by starting from Inst*AL* that is achieved here.

7 DISCUSSION AND CONCLUSIONS

7.1 Incomplete Information

The distributed nature of our framework, with only synthesiser agents being able to access information about the agents they look after, means that for revision tasks, the complete trace or full narrative is not always available. For the narrative to be complete, the acting agents of the timestep that the action causing the need for revision occurred, and the timestep after, must both be supervised by the synthesiser executing the revision for the revision to have a chance of success.

The ILP system's revision can only be as good as the input it receives. Therefore if the action under review has occurred at timestep X, and timestep X+1 is missing from the trace, then it is not possible to observe the effect of the action or generate the appropriate examples that will be needed for the revision task. As a result, the proposed revision may be rejected by the Oracle or on a few rare occasions the revision task will not terminate.

While not ideal, in an online system it is very likely that a similar need for revision will present itself again, and the revision will be successful, as is demonstrated in the experiments in Section 5. In a few rare cases, incomplete information results in a revision task that does not complete and will need to be manually terminated, as we have seen in Experiment 2 in Section 5.2.2.

We must also point out that despite the lack of complete information, it is not possible for a single revision to synthesise incompatible norms. The approach ensures that the norms synthesised do not conflict with each other nor with the existing norms.

7.2 Constraining the learning

Increasing the search space directly impacts the performance of an ILP task, where if the search space is too large, some ILP tasks take very long to compute and utilise significant computational resources, while some tasks become unresponsive and must be manually terminated. We note that this computational problem is reliant on the ability of ILP tools to handle very large search spaces, a problem that is not usually encountered in applications of ILP for simple learning and revisions.

For our purposes, it is necessary to employ methods to reduce the search space which simultaneously has the effect of constraining the learning. In ensuring that the agent-directed norm synthesis framework could be applied to an online MAS, it was non-trivial but necessary to determine how to automate the extraction of the relevant agent experience data for input to the ILP task. For now, the reduction of the search space entails manually annotating the rules to indicate which are revisable, using simulations with a small number of agents and (automatically) generating a custom set of modes in order to limit the mode declarations to only the fluents involved in rules related to the action under review. A more detailed explanation of automated mode generation and other ways of reducing the search space will be the subject of another paper.

7.3 Localised vs. Collective Decision Making

Localised revisions in response to individual experiences are not in general a good way to derive policy as some of these revisions might not be applicable to the entire MAS. However, it is also not advisable to ignore the experiences of individual agents. Instead, following Ostrom's [28] principle of collective decision-making, a mechanism is needed to verify that these localised revisions are applicable for all the agents in the MAS. In our framework, synthesisers ask their peers to check if their responsible agents can continue as normal after the revision. An additional level of oversight is provided by the Oracle. Currently, the behaviour of the Oracle (See Section 3.3) is predicated on a record of the revisions that it will not accept. The authorship of this record could in the future be a more advanced learning agent or another institution with a governance role. Alternatively, we could allow for the implementation of the Oracle as an overarching higher-order institution as proposed by [14] to facilitate collective decision-making.

7.4 Norm Emergence in Normative MAS

This paper demonstrates that the creation stage, as conceptually proposed by [26], whose realisation we have described here, is viable and shows that norm emergence in normative MAS utilising the experiences of participating agents is possible. With this positive outcome, we can now investigate more intelligent participating agents who are not normative by design but are normative by choice and can learn from their interactions. These type of agents will be able to suggest norms for inclusion into the institution based on their experiences within the MAS. These types of norms could be included as obligations for a specific behaviour under a particular circumstance. This will allow us to properly investigate whether a norm that was suggested by a participating agent will first be accepted as part of the institution and also be adopted by the other agents within the MAS. Additionally, we aim to investigate whether more complex scenarios hold in practice. The completion of this necessary next step in the framework paves the way for the implementation of a norm emergence framework in normative MAS, allowing us to observe the emergence of a norm in the MAS that was introduced based on the experience of a participating agent(s) as envisaged in [26].

7.5 Self Governance

The provision of the agent-directed norm synthesis framework to facilitate dynamic institutions in a MAS at runtime, makes it possible for the introduction of self-governing MAS where the norms in the institution are adapted to meet the needs of the participating agents within the bounds of the MAS objectives. The Oracle enables ultimate oversight of the process with the possibility of still allowing a human-in-the-loop, or another form of oversight, to avoid unacceptable runaway-loop revisions over time. An external entity (human, team or higher-order institution) could be responsible for defining the revision control mechanisms in our framework to ensure that the adaptations to the institutions over time are within the boundaries of the intended purpose of the MAS. Hence, frameworks like the one presented here, can in the future provide adaptable self-governance mechanisms of socio-technical systems through dynamic formally specified institutions.

ACKNOWLEDGMENTS

Andreasa Morris-Martin was supported by the Schlumberger Foundation Faculty for the Future program.

REFERENCES

[1] Natasha Alechina, Nils Bulling, Mehdi Dastani, and Brian Logan. 2015. Practical Run-Time Norm Enforcement with Bounded Lookahead, See [36], 443–451.

[2] Chitta Baral. 2003. Knowledge Representation, Reasoning and Declarative Problem Solving.

[3] Guido Boella, Gabriella Pigozzi, and Leendert W. N. van der Torre. 2009. Normative framework for normative system change. In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1*, Carles Sierra, Cristiano Castelfranchi, Keith S. Decker, and Jaime Simão Sichman (Eds.). IFAAMAS, 169–176. https://dl.acm.org/citation.cfm?id=1558036

[4] Rafael Heitor Bordini, Jomi Fred Hübner, and Michael Wooldridge. 2007. Programming Multi-Agent Systems in AgentSpeak using Jason (Wiley Series in Agent Technology).

[5] Jordi Campos, Maite Lopez-Sanchez, Maria Salamó, Pedro Avila, and Juan A. Rodríguez-Aguilar. 2013. Robust Regulation Adaptation in Multi-Agent Systems. *ACM Transactions on Autonomous and Adaptive Systems* 8, 3 (Sept. 2013), 1–27. https://doi.org/10.1145/2517328

[6] Domenico Corapi, Alessandra Russo, Marina De Vos, Julian Padget, and Ken Satoh. 2011. Normative design using inductive learning. *Theory and Practice of Logic Programming* 11, 4-5 (July 2011), 783–799. https://doi.org/10.1017/S1471068411000305

[7] Davide Dell'Anna, Mehdi Dastani, and Fabiano Dalpiaz. 2018. Runtime Norm Revision Using Bayesian Networks. In *PRIMA 2018: Principles and Practice of Multi-Agent Systems - 21st International Conference, Tokyo, Japan, October 29 - November 2, 2018, Proceedings (Lecture Notes in Computer Science, Vol. 11224)*, Tim Miller, Nir Oren, Yuko Sakurai, Itsuki Noda, Bastin Tony Roy Savarimuthu, and Tran Cao Son (Eds.). Springer, 279–295. https://doi.org/10.1007/978-3-030-03098-8_17

[8] Davide Dell'Anna, Mehdi Dastani, and Fabiano Dalpiaz. 2019. Runtime Revision of Norms and Sanctions based on Agent Preferences. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, 1609–1617. http://dl.acm.org/citation.cfm?id=3331881

[9] Christopher Frantz and Gabriella Pigozzi. 2018. Modeling Norm Dynamics in Multiagent Systems. *FLAP* 5, 2 (2018), 491–564. https://www.collegepublications.co.uk/downloads/ifcolog00022.pdf

[10] Amineh Ghorbani and Giangiacomo Bravo. 2016. Managing the commons: a simple model of the emergence of institutions through collective action. *International Journal of the Commons* 10, 1 (Feb. 2016), 200–219. https://doi.org/10.18352/ijc.606

[11] Amineh Ghorbani, Giangiacomo Bravo, Ulrich Frey, and Insa Theesfeld. 2017. Self-organization in the commons: An empirically-tested model. *Environmental Modelling & Software* 96 (Oct. 2017), 30–45. https://doi.org/10.1016/j.envsoft.2017.06.039

[12] Christopher D. Hollander and Annie S. Wu. 2011. The Current State of Normative Agent-Based Systems. *J. Artificial Societies and Social Simulation* 14, 2 (2011). https://doi.org/10.18564/jasss.1750

[13] Xiaowei Huang, Ji Ruan, Qingliang Chen, and Kaile Su. 2016. Normative Multiagent Systems: The Dynamic Generalization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, Subbarao Kambhampati (Ed.). IJCAI/AAAI Press, 1123–1129. http://www.ijcai.org/Abstract/16/163

[14] Thomas C. King, Marina De Vos, Virginia Dignum, Catholijn M. Jonker, Tingting Li, Julian Padget, and M. Birna van Riemsdijk. 2017. Automated multi-level governance compliance checking. *Autonomous Agents and Multi-Agent Systems* 31, 6 (Nov. 2017), 1283–1343. https://doi.org/10.1007/s10458-017-9363-y

[15] Max Knobbout, Mehdi Dastani, and John-Jules Ch. Meyer. 2014. Reasoning about Dynamic Normative Systems. In *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings (Lecture Notes in Computer Science, Vol. 8761)*, Eduardo Fermé and João Leite (Eds.). Springer, 628–636. https://doi.org/10.1007/978-3-319-11558-0_46

[16] Max Knobbout, Mehdi Dastani, and John-Jules Ch. Meyer. 2016. A Dynamic Logic of Norm Change. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016) (Frontiers in Artificial Intelligence and Applications, Vol. 285)*, Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen (Eds.). IOS Press, 886–894. https://doi.org/10.3233/978-1-61499-672-9-886

[17] Robert Kowalski and Marek Sergot. 1986. A logic-based calculus of events. *New Generation Computing* 4, 1 (March 1986), 67–95. https://doi.org/10.1007/BF03037383

[18] Tingting Li, Tina Balke, Marina De Vos, Julian A. Padget, and Ken Satoh. 2013. A model-based approach to the automatic revision of secondary legislation. In *International Conference on Artificial Intelligence and Law*, Enrico Francesconi and Bart Verheij (Eds.). ACM, 202–206.

[19] Tingting Li, Tina Balke, Marina De Vos, Julian A. Padget, and Ken Satoh. 2013. Legal Conflict Detection in Interacting Legal Systems. In *Legal Knowledge and Information Systems - JURIX 2013: The Twenty-Sixth Annual Conference, December 11-13, 2013, University of Bologna, Italy (Frontiers in Artificial Intelligence and Applications, Vol. 259)*, Kevin D. Ashley (Ed.). IOS Press, 107–116. https://doi.org/10.3233/978-1-61499-359-9-107

[20] Mehdi Mashayekhi, Hongying Du, George F. List, and Munindar P. Singh. 2016. Silk: A Simulation Study of Regulating Open Normative Multiagent Systems. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 373–379.

[21] Javier Morales, Maite Lopez-Sanchez, Juan A Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. 2013. Automated synthesis of normative systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 483–490.

[22] Javier Morales, Maite López-Sánchez, Juan Antonio Rodríguez-Aguilar, Michael J. Wooldridge, and Wamberto Weber Vasconcelos. 2015. Synthesising Liberal Normative Systems, See [36], 433–441.

[23] Javier Morales, Maite López-sánchez, Juan A. Rodriguez-Aguilar, Wamberto Vasconcelos, and Michael Wooldridge. 2015. Online Automated Synthesis of Compact Normative Systems. *ACM Transactions on Autonomous and Adaptive Systems* 10, 1 (March 2015), 1–33. https://doi.org/10.1145/2720024

[24] Javier Morales, Michael Wooldridge, Juan A. Rodríguez-Aguilar, and Maite López-Sánchez. 2018. Off-line synthesis of evolutionarily stable normative systems. *Autonomous Agents and Multi-Agent Systems* (June 2018). https://doi.org/10.1007/s10458-018-9390-3

[25] Andreasa Morris-Martin, Marina De Vos, and Julian Padget. 2019. Norm emergence in multiagent systems: a viewpoint paper. *Autonomous Agents and Multi-Agent Systems* 33, 6 (Nov. 2019), 706–749. https://doi.org/10.1007/s10458-019-09422-0

[26] Andreasa Morris-Martin, Marina De Vos, and Julian A. Padget. 2020. A Norm Emergence Framework for Normative MAS - Position Paper. In *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII - International Workshops COIN 2017 and COINE 2020, Sao Paulo, Brazil, May 8-9, 2017 and Virtual Event, May 9, 2020, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 12298)*, Andrea Aler Tubella, Stephen Cranefield, Christopher Frantz, Felipe Meneguzzi, and Wamberto Vasconcelos (Eds.). Springer, 156–174. https://doi.org/10.1007/978-3-030-72376-7_9

[27] Partha Mukherjee, Sandip Sen, and Stephane Airiau. 2007. Emergence of norms with biased interactions in heterogeneous agent societies. In *Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on*. IEEE, 512–515. https://doi.org/10.1109/WI-IATW.2007.115

[28] Elinor Ostrom. 1990. *Governing the Commons. The Evolution of Institutions for Collective Action*. CUP.

[29] J. Padget, Emad Eldeen Elakehal, Tingting Li, and Marina De Vos. 2016. InstAL: An Institutional Action Language.

[30] Oliver Ray. 2009. Nonmonotonic abductive inductive learning. *Journal of Applied Logic* 7, 3 (2009), 329–340. https://doi.org/10.1016/j.jal.2008.10.007

[31] Régis Riveret, Alexander Artikis, Dídac Busquets, and Jeremy Pitt. 2014. Self-governance by Transfiguration: From Learning to Prescriptions. In *Deontic Logic and Normative Systems - 12th International Conference, DEON 2014, Ghent, Belgium, July 12-15, 2014. Proceedings (Lecture Notes in Computer Science, Vol. 8554)*, Fabrizio Cariani, Davide Grossi, Joke Meheus, and Xavier Parent (Eds.). Springer, 177–191. https://doi.org/10.1007/978-3-319-08615-6_14

[32] Bastin Tony Roy Savarimuthu, Stephen Cranefield, Martin K. Purvis, and Maryam A. Purvis. 2009. Norm emergence in agent societies formed by dynamically changing networks. *Web Intelligence and Agent Systems: An International Journal* 7, 3 (2009), 223–232. https://doi.org/10.3233/WIA-2009-0164

[33] Marek J. Sergot. 2007. Action and Agency in Norm-Governed Multi-agent Systems. In *Engineering Societies in the Agents World VIII, 8th International Workshop, ESAW 2007, Athens, Greece, October 22-24, 2007, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 4995)*, Alexander Artikis, Gregory M. P. O'Hare, Kostas Stathis, and George A. Vouros (Eds.). Springer, 1–54. https://doi.org/10.1007/978-3-540-87654-0_1

[34] Marc Serramia, Maite López-Sánchez, Juan A. Rodríguez-Aguilar, Manel Rodríguez, Michael J. Wooldridge, Javier Morales, and Carlos Ansótegui. 2018. Moral Values in Norm Decision Making. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (Eds.). International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 1294–1302.

[35] Daniel Villatoro, Sandip Sen, and Jordi Sabater-Mir. 2009. Topology and Memory Effect on Convention Emergence. IEEE, 233–240. https://doi.org/10.1109/WI-IAT.2009.155

[36] Gerhard Weiss, Pinar Yolum, Rafael H. Bordini, and Edith Elkind (Eds.). 2015. *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*. ACM.