Norms in social simulation

Balancing between realism and scalability

Cezara Pastrav[1,2] and Frank Dignum[2]

[1]Matís ohf., Vínlandsleið 12, 113 Reykjavík, Iceland
cezara@matis.is
[2]Utrech Univeristy, Domplein 29, 3512 JE Utrecht, Netherlands
F.P.M.Dignum@uu.nl

Abstract. Agent based modelling (ABM) has been used to study the dynamics of complex systems, including human societies. However, the design of such models often fails to capture one of the key features of human behavior: norms. While norms and normative behavior are extensively studied in artificial intelligence (AI), especially in the context of multi-agent systems (MAS), their approaches are often very complex and formalized, going against the prevailing discourse of ABM, which advocates keeping the models as simple as possible and pruning any unnecessary complexity. Nevertheless, norms are relevant and integral to many social contexts, and capturing their effect and dynamics often requires agents that, while not as complex as those developed for AI, are capable of sophisticated cognition. We present a normative architecture that attempts to capture the ways norms affect cognition and behavior, while at the same time being lightweight enough to be suitable for ABM use in simulations.

Keywords: ABM, norms, normative architecture, policy, social simulation

1 Introduction

In 1990, Norway changed its fishing policy to limit fisheries expansion and relieve the pressure on recently collapsed cod stocks by introducing a quota system. While this new policy had the desired effect overall, it actually had some paradoxical effects on the small-scale fisheries, which ended up expanding their fishing effort, not only in size catches, but also in time spent at sea. Where before fishers were content to fish during the autumn or the spring season, or both, now they fish year round with unknown long term effects on the stock. The previous fishing policy amounted to open access fishing in the case of small scale fisheries and fishers limited their activity based on social and economic constraints, with young fishers with larger debts fishing the most and then decreasing their activity over time as they paid back their debts, preferring to spend their time with their family or engaged in other activities. This is at odds with the assumption that the larger and more technically capable the boat, the larger the catch. It is also at odds with the economically rational assumption that fishers would try to maximize their profits. When the new quota policy was introduced, based on these erroneous assumptions about fishers' behavior, small scale fishers became incentivized to fish more in order to meet their allotted quota or lose access to the fishery. For details on this particular case of fishing policy backfiring in unexpected ways, see [1].

This is just one of many instances where the models used to inform policy making ignore or misrepresent the behavior of the people being managed. In the case of fisheries, the focus is on mathematical stock assessment models and economic models simplified to be mathematically tractable. It is not surprising they leave out the particular complexities present in the behavior of distinct groups such as small scale fishers. But fisheries are complex systems with ecological, economic and social components, and ignoring or oversimplifying any one of these elements is bound to lead to unexpected and, likely, undesirable results. It is in these situations that agent based modeling (ABM), with its ability to handle so much more complexity of behavior and interaction than mathematical models, is suggested as a better alternative for studying complex systems. But would ABM have done much better had it been applied to the aforementioned example, where "to do better" means "to lead to better understanding of the fishers' behavior and motivations, and how policy might interfere with the already established patterns"?

Human behavior is motivated largely by social values and norms. Fishing behavior in the open access version of the Norwegian small scale fisheries was regulated by social values and norms which dictated fishing effort should be measured based on need (debt to be repaid) and not on ability (technical capacity, experience). Spending time on shore with friends and family was more important than fishing as much as possible. Investing in better boats and equipment, too, meant better working conditions rather than the ability to catch more fish. ABM is known for preferring the simplest possible agents that appear to fit the model. While keeping the model as simple as possible works well enough in many cases, it has been criticized as under-exploitation of the possibilities afforded by the ABM approach, especially the generative aspect of it [2], and calls have been made for abandoning simplicity for more descriptive models which would allow the use of richer empirical data, including qualitative data [3]. In this case, using the simplest agents, built on the simplest assumptions, would likely result in rationally bounded utility maximizing agents or simple ad-hoc heuristics driven agents. These agents are driven by rules directly, rather than some higher order internal motivation. The behavior we want to observe is built directly into them from the beginning.

To go beyond the limitations of such simple agents, models must employ more complex cognitive agents that are able to reason about their values, norms and goals, and modify their behavior accordingly. Unsurprisingly, such an approach involves far more time and effort, and therefore normative agent architectures are far more prevalent in fields like multi-agent systems (MAS), where the focus is on agent cognition, than in ABM, where the focus is on the emergent behavior of the system. Nevertheless, agents capable of normative deliberation are still desirable for at least some classes of ABMs. One of them, the interaction between norms and policy, illustrated by the Norwegian small scale fisheries example, is of particular interest. Designing policy that modulates human behavior in some desired way in a population is often complicated by the fact that usually the population is already governed by a system of norms. If the new policy being implemented clashes with well-established norms, people may choose to ignore, skirt, exploit and generally find surprising and creative ways to not comply. When they do comply, the side effects may still be surprising and undesirable, and what's more, can vary wildly from one population to another, as is too often the case in fisheries policy.

In order to be able to implement models with agents capable of normative deliberation and behavior that are scalable to meaningful agent population sizes for simulation, we develop a lightweight architecture that captures the essential aspects of normative behavior and deliberation, while allowing future modelers to add or subtract functionality as needed for their own applications. We are aware that there is little agreement on what constitutes a norm, let alone what is essential to normative cognition, and we explain and argue for each element while highlighting the fact that our approach is by no means meant to be definitive. Because of page limitations, we only describe the individual agent cognition in this paper, leaving the social aspects of the architecture for a future paper.

The paper is structured as follows: Section 2 provides more information on normative agents and norms in ABM and MAS. Section 3 provides a description of the architecture elements and the justifications for the choice to include them over other candidates. Section 4 presents an example of an agent built with this architecture and its capabilities. Section 5 presents a discussion of further requirements for the architecture to become truly social. Section 6 is the conclusion.

2 Literature

In MAS, the focus is on agent cognition. Normative MAS agents are designed to operate in open systems of heterogeneous agents, and norms are seen as a method of helping the agents behave in ways that are individually and collectively beneficial in the absence of a central regulatory authority, much in the way norms help regulate human behavior. For a good description of what is expected from a normative agent in MAS see [4]. A lot of effort is invested in formalizing all aspects of the architecture, but for a conceptual approach, which also describes and explains the role of many social motivators of behavior in the agent's cognition, see [5]. For a good example of a normative framework for use in MAS, see [6]. And finally, for a comparative review of MAS architectures, see [7].

As for ABM, a few illustrative examples of models that include norms include [8], which studies the emergence of norms using a game theoretical approach, and [9], which studies norm diffusion through memetic contagion. The model in [10] focuses on norm adoption based on the cognitive effort it takes to keep conforming to the current norm vs. adopting a new norm, while [11] looks at the role of norms in controlling aggressive behavior in groups. With such a diversity of approaches, it's difficult to produce a classification of approaches to modelling norms in ABM, except maybe in the broadest terms. For such a comparison of normative behavior, in terms of obedience, conformity and compliance, see [12].

3 Architecture

MAS agents are endowed with complex cognitive abilities based on solid theory, and thus safe from the ad-hoc criticism ABM agents tend to attract. This does not mean they are suitable for ABM. In fact, it is their complexity and theoretical formalism that makes them unattractive for social simulation. First of all, the complexity of MAS normative architectures requires significant computational resources to operate making them impossible to scale simulations to sizes demanded by ABM. Second, the ad-hoc quality of ABM agents has persisted for so long for a reason: the systems being modeled are far too complex to ever be properly formalized, as opposed to the virtual environments in which MAS agents operate, which can be designed and formalized to a much higher extent. ABM modelers are forced to choose which of the characteristics of that system are the most relevant for their research questions, and thus for the model and agents they're building. Some components present in MAS architectures may be irrelevant to the model, and including them can generate unwanted complexity which would make the results harder to interpret [13], not to mention the time and effort wasted parameterizing and calibrating said components.

A normative architecture suitable for ABM should, therefore, be light enough to be scalable, and formalized enough to include the essential components and structures of normative behavior, while allowing the modeler to modify them or add new ones as needed. At the very least, such an architecture needs to separate motivation from behavior, and include the possibility to deliberate about motivations and relate them to desired behavior. The go-to architecture for separating motivation from behavior is the BDI architectures, and normative extensions of BDI architectures (aimed at MAS) exist in literature [14] [15]. We extend the BDI template to include norms and normative deliberation, while remaining relevant to ABM.

3.1 Drivers of behavior

The behavior of an agent can be said to come down to choosing what to achieve and how, or choosing goals and the plans to achieve them. The choice of goals can be influenced both internally, through values, motives, norms, and externally, through limitations imposed by the environment or externally imposed rules, such as policy rules. Internally, values and motives are usually used for selecting goals, while norms can both influence goal selection and act as goals themselves. Values and norms can also influence the type of behavior the agents selects if some (or all) actions are directly related to values, either by promoting or opposing values, or by fulfilling or violating norms.

Since we can set goal priorities directly, without resorting to values or motives, and norms themselves can serve to prioritize goals, we only include goals and norms as minimal requirements for internally driving the behavior of agents in this version of the architecture. Values play an important role in deciding conflicts between norms or whether a norm is internalized or not, but these dynamics are not included for now. We leave it to the modeler to decide whether to add another level of motivation when designing their model.

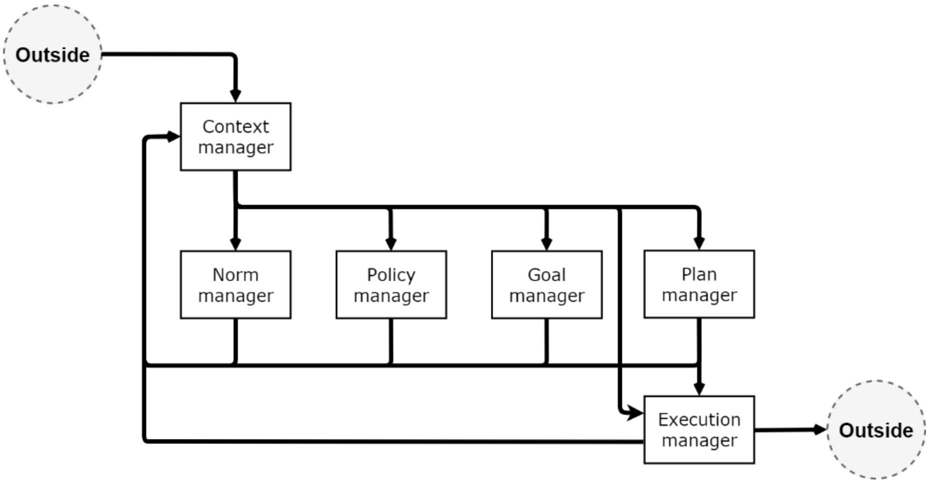

Fig. 1. Conceptual diagram of the architecture showing the more relevant modules and connections between them. The context manager updates the agent's beliefs and motivations, both external (environment states, policies), and internal (norms, goals). The execution manager executes the plans and actions. In between, sit the four main deliberating modules (the norm, policy, goal, and plan managers).

Goals.

Goals are simply states the agent wants to bring about. Some states are preferred over others, whether by definition, or through norm influence, so there exists a preference ordering over all goals. As such, a goal is represented by one or more features of the environment and/or the internal state of the agent together with their desired values, called the condition set of the goal.

In our fishery example, fishers have goals about paying their debt and covering household costs, how much time to spend on shore vs fishing, and how much comfort to work in. A subset of fishers are also driven by the profit motive, as shown by the percentage of fishers who fish considerably more than others.

Norms.

We take norms to be constraints on behavior that are socially agreed upon, rather than imposed by a central authority. Norms affect behavior by requiring certain conditions be met in order for a particular behavior to be acceptable, by dictating the effects a behavior should have or by requiring a specific behavior be performed. As such, norms can act both as conditions as well as goals, and can introduce their own specific actions. Norms are also context specific, becoming active in certain contexts and remaining inactive otherwise.

We define a norm N as a structure [C_a, C_e, G, A, P, C], where:

- C_a is a set of pairs (a, c), where a is an action and c is a set of preconditions that must be fulfilled in order to execute action a without breaking the norm;

- C_e is a set of pairs (a, e), where a is an action and e is a set of postconditions (or effects) for a;
- G is a set of goals;
- A is a set of actions;
- P is a set of punishments for not complying with the norm
- C is the context of the norm, defined as a set of conditions that must be true in order for the norm to apply.

The conditions follow the same structure as goal conditions. If a normative condition overlaps an existing action condition, it replaces it. Otherwise it is added as a separate condition to the action. The goals in G replace or modify existing goals if their condition sets overlap, or are added to the goal pool as new goals. At least one of C_a, C_e, G, A, C must be non-empty in order for the norm to exist. If C is absent, the norm is considered to be always active.

The norm regulating fishing behavior in the example fisheries dictates fishers should fish enough to cover their needs and no more, and that their fishing effort should be proportional to the amount of debt they have. It only applies to small scale fishers, so the context of the norm includes all fishers who own boats that fall under the category of „small boat“.

Policy.

The structure of policies is similar to that of norms. Policy can condition actions, or the effects of actions, and can impose their own actions and goals. At least one of Ca, Ce, G, A, C must non-empty in order for the policy to exist. If the context C is absent, the policy is considered to apply to everyone at all times. However, policies and norms are deliberated about differently. For details, see section 3.3 and 4.5.

The policy being introduced in the second part of the example, conditions access to the fishery by requiring fishers to own quota. It also limits the amount of fishing to the amount of quota owned by the fisher, who is obliged to fish all of it, or gradually lose it and, with it, his access to the fishery.

Roles.

Roles are used to determine type of agents and group membership with regard to the norms and policy that apply to them, as well as which actions the agents are able to perform. As such, a role R is a structure [A, N, P], where A is the set of actions, N is the set of norms and P is the set of policy rules that apply to the role. Agents can have more than one role and can change roles over time.

3.2 Actions

Actions transform environment states/internal state of the agent into other environment states/internal states. They have a set of preconditions and a set of effects, defined, like goals, by sets of conditions. There are three types of preconditions and effects sets:

physical, normative and policy. We define an action a as a structure [f, C_{ph}, C_n, C_p, E_{ph}, E_n, E_p] where:

- C_{ph}, C_n, C_p are the sets of physical, normative and policy preconditions, respectively;
- E_{ph}, E_n, E_p are the sets of physical, normative and policy effects, respectively;
- f: $S \rightarrow S$, $S = \{s \mid s$ is an environment/internal state$\}$, is the function that transforms one state into another state.

The normative precondition and effects are added to an action once the corresponding norm has been adopted, and are activated depending on context. The policy preconditions and effects are added once a policy comes in effect.
In our example, both normative and policy condition sets cannot be active at the same time. See section 5 for details.

3.3 Deliberation

The deliberation process is split between the different manager components of the architecture (see Figure 1). The managers are called in the order bellow:

The **norm manager** keeps track of which norms are active, decides whether the agent complies with a norm, and when a norm is internalized. An agent will comply with a norm if it doesn't conflict with its goals. In case of conflict, the norm manager weighs the cost of punishment for non-compliance vs. the cost of compliance in order to decide whether to follow the norm or not. An agent always complies with an internalized norm.

The algorithm for deliberating about norms is as follows:

```
foreach(norm in Norms)
  if(norm.isActive)
    if(norm.isInternalized)comply with norm
    else foreach(goal in Goals)
        checkCompatibility(norm, goal)
      if (norm is compatible with all goals)
        comply with norm
      else
        costOfCompliance = calculateCostOfCompliance
        costOfNonCompliance = calculateCostOfNonCompliance
        if(costOfCompliance >=costOfNonCompliance)
          comply with norm
```

Norms can conflict with a goal in three ways: the norm may define an alternative to the goal that is incompatible with the preferred version of the goal, the norm may impose conditions on actions that render plans unfeasible and the goal unreachable, and the norm may impose conditions on actions that alter plans in such a way that the goal is no longer met. For a goal to conflict with a normative goal, there must be partial or total overlap between the condition set of the goal and the condition set of the normative goal. When checking for goal compatibility, the manager checks for all three types of conflicts.

Punishments can conflict with goals in the same three ways, so, when calculating the costs of compliance and non-compliance, the manager checks for all three types of conflicts. As a rule, goals becoming unachievable because plans have become unfeasible carry the highest cost of all three possible outcomes.

The **policy manager** keeps track of which policy is active and applicable to the agent, and decides whether the agent complies with the policy by weighing the cost of punishment for non-compliance against the cost of compliance and probability of getting caught. Unlike norms, policies are not internalized, which means the temptation to not comply is always present. Its algorithm is very similar to the one of the norm manager, with additional checks for policy-norm conflicts, which are decided by comparing the non-compliance costs of the norm and the policy.

The **goal manager** keeps track of the goal queue and goal ranking. It has access to a list of active norms and policies and activates the appropriate normative and policy goals in the goal queue. Then, it selects the next active goal in queue and sends it to the plan manager. When the goal is achieved, it moves the goal to the end of the queue and selects the next one. If the plan manager cannot find a plan to achieve the current goal, the goal manager moves the goal to the end of the queue and selects the next one.

The **plan manager** keeps track of the plan library, and can make new plans if needed. It has access to the list of active norms and policies the agent complies with and activates the conditions and effects sets on the relevant actions, and adds relevant norm- and policy-specific actions to the action pool. It receives a goal from the goal manager and selects appropriate plans from the library, ranks them by score, and submits the one with the highest score to the execution manager. If no plan exists, it attempts to build new plans. If successful, the new plans are scored, ranked, and the highest scoring one is sent to the execution manager. If unsuccessful, it declares the goal not achieved and requests a new goal from the goal manager.

The **execution manager** executes the plan received from the plan manager. It checks the feasibility of each action before executing it. If the action cannot be executed, it requests a new plan from the plan manager.

4 Example

In this section we detail the example we presented in the paper in informal terms so far. We focus on some of the more relevant aspects of the example. The abridged version asks: how do the goals, norms and policies influence an agent's decision about how much fishing effort to exert over a predetermined period?

4.1 The goals

The fishers in the small scale Norwegian fisheries largely fall into two categories: those who fish enough to cover their debts and living costs, and those who fish to maximize their profit. Both these goals refer to the money one expects to make from fishing over a period of time, so we can write the two goals like this:

```
goal1 = {money > min_debt}
goal3 = {money = max_money}
```

Money obained from fishing depends on fishing effort (measured in days at sea), the efficiency of the boat (as a function of size, gear and the experience of the fisher) and the value of the catch. It is calculated as:

```
money = days_at_sea * efficiency * fish_value
```

Min_debt is the minimum costs the agents need to cover over a given time period, and includes both the percentage of their total debt that needs to be repaid and their living costs for the same period. **Max_money** is the maximum amount of money that can be obtained through fishing over the given period. To maximize their income, fishers need to spend the maximum amount of time at sea. The income can also be increased by acquiring more efficient boats, or by fishing more valuable fish, but for the purpose of this paper, maximizing the time spent at sea is sufficient.

4.2 The norm

The norm regulating this fishing behavior dictates fishers should fish enough to cover their needs and no more, and that their fishing effort should be proportional to the amount of debt they have:

```
N_fishing = [G = {money = current_debt * repayment_rate},
             C = {boat = small_boat}]
```

All other elements of the norm are empty.

4.3 The policy

In our case, policy dictates an agent must have quota in order to fish and must fish its quota or lose access to the fishery. We can write the policy rules like this:

```
P_fishing =
[Ca = {(fishing_action, {remaining_quota > 0)},
Ce = {(fishing_action, remaining_quota -= fished_quota)},
G = {money = quota_depletion_rate * initial_quota *
fish_value},
P = {fishing_action, {remaining_quota = 0}],
```

All other elements of the policy are empty.

4.4 Actions and plans

In our example, the condition sets of the fishing action can be represented as:

```
fishing_action =
[Cph = {boat != null, days_at_sea > 0},
 Eph = {money = days_at_sea * efficiency * catch_value}]
```

where **days_at_sea** is time spent fishing measured in days and efficiency is determined by the type of boat and gear the agent has, as well as its experience in fishing. If the agent follows the norm that dictates time spent fishing is proportional to the debt, we add the corresponding normative precondition:

```
fishing_action =
 [Cph = {boat != null, days_at_sea > 0},
  G = {money = current_debt * repayment_rate}]
```

If the quota policy comes into effect, the action becomes:

```
fishing_action =
 [Cph = {boat != null, days_at_sea > 0},
  G = {money = quota_depletion_rate * initial_quota *
fish_value},
  Cp = {remaining_quota > 0},
  Ep = {remaining_quota -= fished_quota}]
```

4.5 Deliberation

During the deliberation phase, the agents decide whether to comply, first with the norm, then with the policy. During deliberation three conflicts arise and are resolved: between the norm and the profit maximization goal, between the policy and the norm, and between the policy and the profit maximization goal.

About the norm.
The norm conflicts with the profit goal because both the goal and the norm attempt to attach incompatible values to the money obtained through fishing. The norm manager identifies the conflict when checking whether the norm interferes with the goal by assessing the condition sets of the goal and the normative goal, and comparing the results. The norm requires a **money** value that is less than the one required by the profit goal. Since there is no punishment attached to violating this norm, agents with profit goals choose to not comply with the norm.

The norm does not conflict with the debt repayment goal. The goal allows for any value for the **money** attribute, as long as the value covers the minimum amount required to repay the debt, and the norm ensures debt repayment by definition. Since there is no conflict between their goal and the norm, agents with the debt repayment goal decide to comply with the norm, and the debt repayment goal take the form of the normative goal.

About the policy.

In the case of agents with the debt repayment goal, the policy conflicts with the norm because, again, they both attempt to attach incompatible values to the `money` attribute of the goal. The real life policy assigned fishers amounts of quota that required higher fishing effort to fish in their entirety, but even if that weren‘t the case, the fact that debt declines over time, while quota does not, forces the conflict between the norm and policy eventually. The conflict is resolved by calculating non-compliance costs for the norm and the policy and complying with the one with the higher non-compliance cost, in this case, the policy. Interestingly, once the norm is rejected, the goal reverts to its original form, which only requires that fishing money covers the minimal debt repayment rate, without an upper limit. As such, the policy does not conflict with the original goal and is complied with. The goal takes the form of the policy goal.

In the case of agents with the profit goal there is no norm-policy conflict since the agents do not comply with the norm. However, the policy conflicts with the profit maximization goal since the amounts of quota allocated are calculated to fall under the maximum capacity of the boats. The conflict is resolved by calculating and comparing the cost of compliance and non-compliance. Since the punishment for non-compliance is loss of access to the fishery, fishing becomes impossible and the goal becomes unachievable. As mentioned in the previous section, this carries a far greater cost than mere loss of profit. The policy is complied with and the goal takes the form of the policy goal.

About the plan.

Once the deliberation about the norm and policy is complete and the goal is set to its final form, the plan manager takes over. For simplicity‘s sake, we limited the plan to the necessary minimum: one action – fish. It‘s the plan manager‘s task to parameterize this one action in order to fulfill the goal. As such, it is the plan manager that, finally, gives an answer to how much effort do fishers need to exert in order to meet their goals.

In the absence of the quota policy, agents with the debt repayment goal have lower effort requirements since their goal requires less money be made from fishing, and, since the comply with the norm, their effort decreases over time. Agents with the profit maximization goal have the highest effort of all, since they refuse to comply with the norm and the quota policy is not in place to limit the amount of money they can aim to make from fishing.

When the quota policy is introduced, the agents with the debt repayment goal are forced to discard the norm and comply with the policy insted, which raises their fishing effort and maintains it at a constant level over time. Agents with the profit maximization goal are also forced to comply with the policy, and their effort level is lowered.

If the overarching goal of the policy is to lower total effort in the small scale fisheries, then it comes down to whether the total reduction in the effort of the profit seeking agents is greater than the total increase in the effort of agents that follow the norm.

In the real life fishery this was not the case, but it would be interesting to calculate whether there exists a quota allocation rule that would prove successful.

5 Discussion and future work

The architecture described in this paper aims to allow agents to reason about social norms, while remaining light enough to be scalable for social simulations. Due to space limitations, we chose to present the individual cognition aspect of the architecture, but a proper normative architecture must include social cognition as well. We will briefly touch on this aspect here.

First, the social part of norms. Norms are generally enforced socially, by other agents. This requires agents to be able to recognize normative behavior in other agents, specifically whether another agent is not complying with a norm. Once such behavior can be recognized, the agent must decide whether to punish the offending agent, which required additional deliberation capabilities.

Second, agents need to be able to communicate about norms, at least. This is especially relevant when not all agents are aware of all existing norms and need to be informed about them.

Third, a social structure connecting agents to one another needs to be in place over the agent population in order to be able to apply social pressure with regards to norms. We prefer a social network because it makes it easy to define norm contexts in terms of who is responsible for monitor

\Seing and punishing whom, as well as for delineating normative contexts (work related norms apply on the work network among agents connected by work relations, for instance).

6 Conclusion

We have described a normative architecture that attempts to strike a balance between the formal and rigorous normative architectures of MAS agents and the overly simplified ad-hoc, but flexible, architectures of ABM agents. The goal was to build an architecture that is capable of deliberating over norms, while remaining light enough to be scalable, and flexible enough that modelers can adapt it to the requirements of their own models. We also presented a real life example of a situation where norms play an important part in the dynamics of the system (small scale fisheries in Norway), highlighting the desirability of such architectures in ABM, and showed how it can be implemented using our proposed approach.

Acknowledgements: This work was funded as part of the SAF21 project, financed under the EU Horizon 2020 Marie Skłodowska-Curie MSCA-ETN program (project 642080).

References

1. A. Maurstad, "To fish or not to fish: small-scale fishing and changing regulations of the cod fishery in northern Norway," *Human organization,* vol. 59, pp. 37--47, 2000.
2. R. Conte and M. Paolucci, "On agent-based modeling and computational social science," *Frontiers in Psychology,* vol. 5, p. 668, 2014.
3. B. Edmonds and S. Moss, "From KISS to KIDS--an 'anti-simplistic'modelling approach," in *International workshop on multi-agent systems and agent-based simulation*, 2004.
4. M. Luck, S. Mahmoud, F. Meneguzzi, M. Kollingbaum, T. J. Norman, N. Criado and M. S. Fagundes, "Normative agents," in *Agreement Technologies*, Springer Netherlands, 2013, pp. 209-220.
5. F. Dignum, V. Dignum, R. Prada and C. M. Jonker, "A conceptual architecture for social deliberation in multi-agent organizations," *Multiagent and Grid Systems,* vol. 11, no. 3, pp. 147-166, 2015.
6. F. L. y López, M. Luck and M. d'Inverno, "A normative framework for agent-based systems," *Computational \& Mathematical Organization Theory,* vol. 12, pp. 227-250, 2006.
7. M. Neumann, "The cognitive legacy of norm simulation," *Artificial Intelligence and Law,* vol. 20, no. 4, pp. 339-357, 2012.
8. R. Axelrod, "An evolutionary approach to norms," *American political science review,* vol. 80, no. 4, pp. 1095-1111, 1986.
9. D. Gatherer, "Identifying cases of social contagion using memetic isolation: Comparison of the dynamics of a multisociety simulation with an ethnographic data set," *Journal of Artificial Societies & Social Simulation,* vol. 5, no. 4, 2002.
10. J. M. Epstein, "Learning to be thoughtless: Social norms and individual computation," *Computational economics,* vol. 18, no. 1, pp. 9-24, 2001.
11. R. Conte and C. Castelfranchi, "Understanding the functions of norms in social groups through simulation," *Artificial societies: the computer simulation of social life,* vol. 1, pp. 252-267, 1995.
12. C. Elsenbroich and M. Xenitidou, "Three kinds of normative behaviour: minimal requirements for feedback models," *Computational and Mathematical Organization Theory,* vol. 18, no. 1, pp. 113-127, 2012.
13. F. Squazzoni, "The "Reign of Mystery": Have We Missed Something Crucial in Our Experimental and Computational Work on Social Norms?," in *The Complexity of Social Norms*, Springer, 2014, pp. 177-188.
14. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang and L. van der Torre, "The BOID architecture: conflicts between beliefs, obligations, intentions and desires," in *Proceedings of the fifth international conference on Autonomous agents*, 2001.
15. F. Dignum, D. Kinny and L. Sonenberg, "From desires, obligations and norms to goals," *Cognitive science quarterly,* vol. 2, no. 3-4, pp. 407-430, 2002.