Social simulation for socio-ecological systems

An agent architecture for
simulations of policy effects

SIKS Dissertation Series No. 2022-32
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems

Cover image generated with Midjourney
DOI: https://doi.org/10.33540/1511

Social simulation for socio-ecological systems

An agent architecture for simulations of policy effects

Sociale simulatie voor sociaal-ecologische systemen
Een agentarchitectuur voor simulaties van beleidseffecten

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit Utrecht
op gezag van de
rector magnificus, prof.dr. H.R.B.M. Kummeling,
ingevolge het besluit van het college voor promoties
in het openbaar te verdedigen op

vrijdag 11 november 2022 des middags te 12.15 uur

door

Cezara Maria Păstrăv

geboren op 18 maart 1984
te Vaslui, Roemenië

Promotor:

Prof. dr. J.J.C. Meyer

Copromotor:

Dr. F.P.M. Dignum

Beoordelingscommissie:

Prof. dr. M.M. Dastani
Prof. dr. P. Yolum Birbil
Prof. dr. P. Ahrweiler
Prof. dr. H. Lindgren
Prof. dr. P. Novais

Dit proefschrift werd mogelijk gemaakt met financiële steun van het project SAF21, "Social Science Aspects of Fisheries for the 21st Century". SAF21 is een project dat wordt gefinancierd in het kader van het EU Horizon 2020 Marie Sk lodowska-Curie (MSC) ITN-ETN-programma (project 642080).

Abstract

Socio-ecological systems (SES) are complex system in which human society is deeply intertwined with the natural world. Fisheries, forestry, water and land use, national parks, all of these and more are examples of context in which human and natural elements are fused together to where they form a new, distinctly recognizable kind of system. Many of our most difficult contemporary problems arise in SES: overfishing, deforestation, damaging tourism, habitat destruction caused by urban and industrial developments, and, of course, climate change. The complexity of SES means that evidence-based policy isn't always the best approach because the issues that arise in SES don't typically have a single universally recognized solution. An alternative policy design process would include the participation of stakeholders and inputs from more scientific disciplines than just the quantitative ones.

Computational models are one of the most useful tools for policy makers, with agent-based modeling (ABM) standing out as particularly well suited to studying policy effects in SES. ABM is at its most useful when analytical solutions are either too complex to be computed or lead to an oversimplification of the system being modelled. It can integrate related, connected or interacting issues that would traditionally be treated separately by different disciplines of science, as well as processes that occur at different scales of time and space. And, especially relevant to the participatory policy design suited to SES governance, it can integrate stakeholder perspectives and integrate *with* stakeholders - to the extent that participatory modelling has become widely accepted in SES modelling.

However, ABM still struggles with a dearth of realistic social and decision models, which is particularly troublesome if ABM is to be included in the policy design process for governing systems in which the human component is crucial to the functioning and behavior of the system. If the models fail to capture it properly, the results will lack informative power or can be downright misleading. In such models, agents need to have a decision process that can operate with social norms and values, at least. Taken together, values and social norms form a particularly stable and consistent framework for the decision making processes of an agent, encompassing both motivations and preferred means of pursuing said motivations. Policy, as another kind of norm, fits in this framework as either supporting/reinforcing (when it promotes the values of an agent or works together with the social norms of an agent) or antagonistic/conflicting (when it goes against the values of an agent or conflicts with and the social norms of an agent).

In this work, we present our agent architecture, built to account for human decision making in contexts where norms meet policy, while also remaining lightweight enough to be usable in ABM. As such, it provides explicit representations of the cognitive elements involved, and realistically replicates the normative deliberation process, while remaining scalable. We also present a modular implementation of the architecture, and a visual model builder that leverages the modularity of the implementation to allow for fast agent and simulation setup. Finally, we demonstrate the use of the architecture by simulating a number of scenarios derived from a real-world instance of fishing policy and its effects. The scenarios cover different assumptions

about the reasoning and motivations of the agents (profit seeking goal-oriented, normative goal-oriented, value driven self-interested, value-driven community-oriented) and their response to the same policy being introduced during times of abundance or scarcity of resource.

Samenvatting

Sociaal-ecologische systemen (SES) zijn complexe systemen waarin de menselijke samenleving diep verweven is met de natuurlijke wereld. Visserij, bosbouw, water- en landgebruik, nationale parken, dit alles en nog veel meer zijn voorbeelden van context waarin menselijke en natuurlijke elementen worden versmolten tot een nieuw, duidelijk herkenbaar soort systeem. Veel van onze moeilijkste hedendaagse problemen doen zich voor in SES: overbevissing, ontbossing, schadelijk toerisme, vernietiging van leefgebieden veroorzaakt door stedelijke en industriële ontwikkelingen, en natuurlijk klimaatverandering. De complexiteit van SES betekent dat evidence-based beleid niet altijd de beste aanpak is, omdat de problemen die zich voordoen bij SES doorgaans niet *één* universeel erkende oplossing hebben. Een alternatief proces voor beleidsontwerp zou de deelname van belanghebbenden en input van meer wetenschappelijke disciplines dan alleen de kwantitatieve disciplines omvatten.

Computationele modellen zijn een van de meest bruikbare instrumenten voor beleidsmakers, waarbij agent-based modeling (ABM) eruit springt als bijzonder geschikt voor het bestuderen van beleidseffecten in SES. ABM is het nuttigst wanneer analytische oplossingen ofwel te complex zijn om te worden berekend, ofwel leiden tot een te grote vereenvoudiging van het systeem dat wordt gemodelleerd. Het kan gerelateerde, samenhangende of op elkaar inwerkende kwesties integreren die traditioneel afzonderlijk door verschillende wetenschappelijke disciplines zouden worden behandeld, evenals processen die plaatsvinden op verschillende schalen van tijd en ruimte. En, vooral relevant voor het participatieve beleidsontwerp dat geschikt is voor SES-governance, kan het de perspectieven van belanghebbenden integreren en *met* belanghebbenden integreren - in de mate dat participatieve modellering algemeen geaccepteerd is geworden in SES-modellering.

ABM worstelt echter nog steeds met een gebrek aan realistische sociale en beslissingsmodellen, wat vooral lastig is als ABM moet worden opgenomen in het beleidsontwerpproces voor bestuurssystemen waarin de menselijke component cruciaal is voor het functioneren en het gedrag van het systeem. Als de modellen het niet goed kunnen vastleggen, missen de resultaten informatieve kracht of kunnen ze ronduit misleidend zijn. In dergelijke modellen moeten agenten een beslissingsproces hebben dat op zijn minst kan werken met sociale normen en waarden. Alles bij elkaar genomen vormen waarden en sociale normen een bijzonder stabiel en consistent kader voor de besluitvormingsprocessen van een agent, die zowel motivaties als voorkeursmiddelen omvat om die motivaties na te streven. Beleid, als een ander soort norm, past in dit kader als ofwel ondersteunend/versterkend (wanneer het de waarden van een agent bevordert of samenwerkt met de sociale normen van een agent) of antagonistisch/conflicterend (wanneer het indruist tegen de waarden van een agent of in strijd is met en de sociale normen van een agent).

In dit werk presenteren we onze agentarchitectuur, gebouwd om rekening te houden met menselijke besluitvorming in contexten waar normen en beleid voldoen, terwijl het ook licht genoeg blijft om bruikbaar te zijn in ABM. Als zodanig biedt het expliciete representaties van de betrokken cognitieve elementen en repliceert het op realistische wijze het normatieve deliberatieproces, terwijl het schaalbaar blijft. We presenteren ook een modulaire implementatie van de architectuur en een visuele modelbouwer die gebruikmaakt van de modulariteit van de implementatie om snelle agent- en simulatie-instellingen mogelijk te maken. Ten slotte demonstreren we het gebruik van de architectuur door een aantal scenario's te simuleren die zijn afgeleid van een praktijkvoorbeeld van visserijbeleid en de effecten ervan. De scenario's hebben betrekking op verschillende veronderstellingen over de redenering en motivaties van de agenten (winstzoekend doelgericht, normatief doelgericht, waardegedreven eigenbelang, waardegedreven gemeenschapsgericht) en hun reactie op hetzelfde beleid dat wordt geïntroduceerd in tijden van overvloed of schaarste aan middelen.

Contents

Part I Socio-ecological systems and governance

Part II Conceptual model

Part III Implementation, models and results

List of Figures

List of Tables

Acronyms

ABM	agent based system
SES	socio-ecologic system
EEZ	exclusive economic zone
TAC	total allowable catch
ITQ	individual transferable quota
MAS	multi-agent system

Acknowledgements

Now that this work is finally in an academically acceptable format, I would like to thank a few people who helped, supported and, occasionally, dragged me kicking and screaming on the path towards graduation. First of all, Frank Dignum, a saint among men, who managed to keep me on the road to finishing this dissertation, despite my every attempt to wriggle free of having to write anything down. May he never have to work this hard to convince someone that, yes, writing stuff down is important, and no, code doesn't count. And may all PhD students have the good fortune of a supervisor as knowledgeable, patient and supportive as he is.

I would also like to thank Jonas Vidarsson, who supervised me while I was part of SAF21 at Matis, in Iceland. He did all he could to get me and my SAF21 fellows to understand the fishing industry in general and its role in Icelandic culture in particular. It was lovely being at Matis, in no small part thanks to him.

Speaking of SAF21, I have to thank Melania Borit for managing an ambitious and challenging project that I enjoyed immensely. I learned and grew so much, both a person and as a researcher and I owe it all to her, and everyone involved who worked so hard to make it such a great experience and opportunity.

I would also like to extend my gratitude to Samaneh Heidari who offered me her support and relentless encouragement, all while working on her own PhD. She persuaded me to believe in myself and my research even when I was at my lowest points and was considering running away to the woods to become a goat herder because I was obviously not cut out for this academia business.

And finally, my most gracious thanks to everyone else I met during this journey who offered me a kind word, a piece of useful advice or a bit of well deserved and well thought out criticism. I may have too deeply consumed by my own struggles to properly communicate it at the time, but I'll remember all of you fondly each time I think back on my years as a PhD.

Chapter 1

Introduction

> I have yet to see any problem, however complicated, which, when looked at in the right way, did not become still more complicated.
>
> Poul Anderson

One night, having fallen down an Internet hole, a pastime that I'm more fond of than I'm comfortable admitting, I ran across this title *Psychics in Portland Are Helping Mainers Find Their Lost Weed* about a company that uses psychics to help people find their lost weed. According to the source [17]:

> The process is simple, if you're over the age of 21 with ID, you visit the website and pick the product you lost along with your address and the psychic will consult with the weed spirits and find and deliver your lost weed to you for a fee!

I couldn't resist digging the hole a little deeper, of course. At the time the article was written, possession of weed for recreational consumption was legal in Portland, but selling it was still limited [87], hence the *finding* and *returning* parts of the business model. A stellar, and hilarious, example of following the letter of the law.

However, given the long and unpleasant history of the war on drugs, this business should have a hard time existing. After all, it wouldn't be very hard to prove no one was out looking for lost possessions, had law enforcement decided to pursue an investigation. Unless American police acquired an unshakeable belief in the existence and power of psychics while no one was looking, something else should be going on. As it turns out, Maine was on its way to fully legalize recreational weed by summer of 2020, but full implementation of the laws was delayed [87], so while legal norms were not yet fully in place, social norms were. This makes an investigation by law enforcement not worth the bother and the "psychics" of Portland can make a lucrative business while waiting for the law to catch up.

I found this to be a fairly straightforward and entertaining example of what can happen when social norms clash with legal norms. People can be quite creative when trying to achieve their goals, and sometimes the law is just another obstacle to overcome. As a policy maker you can expect some people will try to break or bypass the law, and, thus, plan suitable countermeasures, such as additional enforcement and punishment or closing exploitable loopholes. But how can you know what kind of opposition will your laws face? And can you anticipate which strategies people will employ should they decide not to follow the law? Is there a way to anticipate weed-finding psychics?

Drug abuse is part of a class of problems known as wicked problems, where even the definition of the problem itself is difficult to formulate [115]. There are no definitive solutions, only repeated attempts at managing and mitigating the negative impacts. These problems are borne out the complexity of the systems that give rise to them, such as social, behavioral and institutional complexity. There are many reasons people choose to use drugs and it is not fully understood who among them, and under what conditions, will become addicted [57]. Addiction itself can have a range of effects on the individual and their social group, and their severity doesn't always factor into the legal status of the addictive substance. Tobacco, for instance, is perfectly legal, despite causing severe health problems over time. One could argue that nobody becomes severely impaired if they suddenly lose access to cigarettes. However, alcohol addicts do become unable to function if they have to suddenly stop drinking, and alcohol withdrawal can be deadly. Nevertheless, alcohol is legal.

The treatment of drug users and drug providers by law enforcement and society at large can, again, determine the kinds of damage they cause to themselves and to others (see for instance [144]). An overly strict criminalizing of drug use, such as the infamous "war on drugs", leads to disproportionate penalties for drug use, which can then prevent the offender from finding employment, harshly limiting their options for a productive life. It also pushes addicted users away from getting help, even medical help, and, in general, can hinder much needed medical treatment if said treatment involves medical versions of illegal drugs, with people either being turned away under the suspicion of drug-seeking behavior or being made to jump through administrative hoops in order to get their treatment approved. Social stigma associated with drug use cuts addicts away form what should be their social support network and can keep people from accepting medical treatment involving certain drugs in order to avoid being labeled as drug-users. Biases and prejudice in the legal system can focus anti-drug efforts on certain communities, while allowing others to operate with almost no oversight, giving rise to new forms of old problems. Such is the case of the current US opiate crisis, which started with the overprescription of opiates for pain relief under the misleading marketing efforts of pharmaceutical companies ([135], [85]). Lawmakers eventually caught on and put a sudden stop to the flow of legal opiates, but took no steps to aid the massive number of existing addicts, who now faced the choice of going though an extremely difficult withdrawal or get their drugs elsewhere. Many chose to get their drugs on the street, giving rise to an unprecedented number of overdose deaths in communities who had barely had any drug issues before and were, thus, caught unprepared ([109], [145]).

It’s no wonder, in this snarled landscape of serious problems caused by drugs and overreaction by law enforcement, that the citizens of Portland had to – temporarily, one hopes – get their almost-legal weed from psychics.

Wicked problems aren’t always as obviously rich in human drama as drug abuse, but they can arguably get even more complicated. Fertile ground for wicked problems can reliably be found at the intersection between human activities and ecologic systems. One such problem, which is the case study for this thesis, is the human exploitation of ocean resources, particularly fishing.

1.1 Fisheries are wicked

On the surface, the main problem with fishing is over-exploitation of the resource leading to the depletion of fish stocks. It looks like an easy equation to balance: if there isn’t enough fish to go around, fish less. Put up limits to how much people can fish and you solve the problem. However, fishing, like many other human activities, is deeply intertwined with culture and tradition. These forces exert a substantial pull on human behavior and fishers will try to maintain their way of life despite laws and regulations. For example, limiting the number of days they can be at sea tends to result in "Olympic fishing", with fishers engaging in a race to upgrade their vessels and gear to the point where they can end up fishing more in a limited season than they did in a whole year before the effort limitation rules got put in place [72].

Then we have to consider that fishers aren’t simply profit maximizing entities steering vessels onto the sea, but people embedded into communities, and in many cases vital components of these communities. For instance, Iceland instituted an individual transferable quota policy (ITQ), which solved their overfishing problem, but saw many of their small scale fishers leave the trade, putting some of their coastal communities in real danger of collapse [51]. Some of these fishers left because they couldn’t compete with the large fishing companies, which is to be expected, but the policy also made quota extremely valuable which lead to the younger generation selling their inherited quota in order to get a comfortable start in life in the big city. The same high price for quota also means a very high bar for entrance into the fishery, so even if other members of the community wanted to pick up the fishing mantle, they’d find it exceedingly difficult to do [28].

Iceland is also a good example of how fishing can become an issue in international politics. Fishing is deeply embedded in Icelandic culture, connected with independence, self-reliance and resilience [27]. It is also a considerable part of their economy, given that the island doesn’t offer much else in terms of natural resources [83]. These two factors lead to Iceland getting involved in international conflicts over fishing rights, dubbed the "cod wars" [78], and refusing to become a part of the European Union unless it could maintain full control over its waters [16].

Fisheries are an excellent example of the particular kind of complex system known a social-ecological system (SES). Designing policy is difficult at the best of times, but designing policy for sustainable SES is made even more difficult because of their nature. They are complex systems containing ecologic, social and economic components, which interact with each other in ways that are both complex and, often, irreducible. This compounded complexity, with both people and the resource behaving in complex ways, with an added layer of complexity coming from the interaction between them, is one of the fundamental characteristics of such systems.

In the case of fisheries, we have the fish, who are part of a vast ecosystem and their behavior within it is complex as a result of both other species and environmental conditions. Understanding the dynamics of fish stock is central to the development of fishing policy because any such policy must determine how much fish can be harvested without collapsing their populations, while at the same time ensuring a satisfactory economic return. There is a vast literature detailing many different approaches to estimating how much fish lives within a fishery's area of exploitation and how much can be harvested sustainably (for examples, see [99], [88], [54], [73]). The literature also contains critiques and discussions of the current limitations of such assessment models, ranging from uncertainty due to insufficient data [42] to largely ignoring the human element [56] to how they're used in policy decisions [71].

On top of this ecological intricacy and its accompanying difficulties when it comes to trying to figure out what's really out there in the water, we have the fishers. It's tempting to view them, on one hand, as another, disproportionately successful, predator of the fish and, on the other hand, as economic agents exploiting a valuable resource, but this doesn't capture the whole picture. Depending on the methods and gear, fishing can be an extremely ecologically destructive activity, affecting species of no economic value by catching them alongside the target species [84]. In the worst cases, fishing affects the whole environment through the use of bottom trawlers which scrape the bottom of the ocean destroying everything in the targeted area [96]. This is far outside the capabilities and behavior of any predator in nature. As for being rational economic agents, while many fishers are indeed organized and operate as businesses, many others, as illustrated by our discussion of Icelandic fishing, have much more complex motivations than simple profit maximization (for examples, see [132], [77], [28]). For other fisheries, the motivations behind the behavior of fishers may be less understood [79].

1.2 Human behavior in models

When it comes to modelling people's behavior in policy discourse, "overly simplistic" is the best way to describe the majority of approaches. Despite not being use as widely, there are a range of richer and more nuanced models of human behavior being theorized and developed [123]. *Homo economicus*, who hails from economics, is the quintessential "overly simplistic" model of human behavior: a perfectly rational being, purely self-interested, all-knowing with regards to all relevant information (ranging from other agents to the probability of all possible events) who directs its behavior by solving specific optimization problems under specific constraints. One step close to humanity, are models of "bounded rationality", which allow that people are not optimization engines, their cognitive capacity is limited, and their their thinking is often biased. Another step closer are models which recognize that people are inherently social and their behavior is tightly connected to and influenced by the behavior of others. The final step is recognizing that human behavior is not separated from the environmental context in which it occurs, rather it is fundamentally influenced by it and influences it in return in feedback loops that continuously require adaptation on both sides of the equation.

Using overly simplistic models of human behavior which assume rational and economically motivated agents can lead to a host of unintended effects on the communities being governed. For instance, privatizing fisheries can result in a host of transformations at the social level [97]. Fishing communities are not homogeneous, but comprise fishers who hold different values, beliefs and attitudes, which result in different behaviors, which can deviate in unforeseen ways from the expected economic rationality. Relations of power must also be considered because they can also impact the decisions and behaviors of community members. All these influences can result in detrimental changes in the structure of communities, up to rendering some unviable.

Despite the fact that this economic assumption of rationality is unsound [141] and more complex representations of human behavior are available, there continues to exist a strong preference for models of human decision making that are simple, less nuanced, and arguably less informative ([61], [124]), if the models even include people and their behavior at all. For instance, fisheries science prefers stock models and other resource focused research, and is much less concerned with research about the lives of fishers and fishing communities, a stance which is reflected quite starkly in their measuring of fisheries management by various versions of overfishing vs. maximum yield [73].

1.2.1 Agent based models

When studying SES we have to contend not only with the complexity of the ecological and the social components, but also with the complex interactions between and within them. In order to tackle this challenge, we need to use models to help us untangle these relationships, the effects they have and the ways these effects feedback and reshape both the system components and the relationships between them.

One of the most promising tools for tacking the complexity of SES are agent-based models (ABM) ([23], [48]). Unlike mathematical models, they can capture far more complexity, both in the type of system components and in the types of interactions between them, up to simulating whole societies, albeit simple ones, in silico [49].

However, ABMs are not without drawbacks. Among these, the biggest challenges are a lack of transparency and reusability, difficulties validating models, limited software, insufficient theoretical grounding, and limited models of human decision making [9]. Two of these challenges are of particular interest for this work: transparency and reusability, and models of human decision making.

Despite their potential, many ABMs tend towards simple models of human behavior, often rule- or utility-based with an assumption of rationality. Such ABMs offer limited insight into the behavior of the system they model, and, as any ABM, can even run the risk of being misleading, as some in the field have warned [44]. In many cases, 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 [64].

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, and calls have been made for abandoning simplicity for more descriptive models which would allow the use of richer empirical data, including qualitative data [45]. Rationality assumptions are tempting not only because they simplify the process of modelling of human behavior, but also because they are applicable to a wide range of situations. However, this assumption makes it impossible to build social agents, who are influenced by their social ties, culture or social role [40].

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, goals, identities, roles etc. and modify their behavior accordingly [39]. 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 developing agent cognition geared toward solving specific problems, than in ABM, where the focus is on the emergent behavior of the system and generative power [146]. Nevertheless, agents capable of normative deliberation are still desirable for at least some classes of ABMs. Among them, the interaction between norms and policy is of particular interest. Designing policy that modulates human behavior in some targeted way is often complicated by the fact that, usually, the population is already governed by a system of social norms and has its own culture and values [107]. If the new policy being implemented clashes with well-established norms and/or values, people may choose to ignore, skirt, exploit and find otherwise surprising and creative ways to not comply [100]. When they do comply, the side effects may still be surprising and undesirable [120], and whats more, can vary wildly from one population to another [30], as is too often the case in fisheries policy.

Transparency and reusability are of particular importance to policy modelling for SES because of the characteristics of both ABMs and SES. The ability of ABM to represent a vast array of elements and dynamics can also makes them hellishly complicated, with large numbers of parameters that can be tuned and many different types of agents interacting in numerous ways. This can make it difficult to determine how the outputs of the model are generated, which negates any attempts at using the model to explain the modelled system and the effects of any policy being introduced. Compromises of simplification often also sacrifice the informative power of the models [48].

A lack of transparency also has implications for model replication and reusability. Without the ability to clearly determine the role and function of every piece of the model, attempts to replicate it will likely fail, and determining whether the model, or parts of it, can be used in a different context are far more likely to produce misleading results [46].

The complexity of SES compounds these issues. Not only are such systems composed of many different entities, these entities interact with one another in complex ways. When it comes to designing policy for governing such a system, the process may involve numerous stakeholders, each of them coming to the table with their own agenda and their own perspective of the SES. Furthermore, many of the concepts being considered in SES policy, such as "resilience" or "sustainability", are not very well defined ([34], [29]), and stakeholders may not immediately agree on their meaning in their particular context. Integrating all these perspectives and all the policy options being considered in the same model may not be feasible without resulting in a model so complex that its functioning and results become unintelligible. It may, therefore, be more advisable to build more than one model in order to address such problems [59]. Such models are likely to overlap, and a lack of reusability would not only increase the modelling effort considerably, but also make any comparison between the results of different models questionable.

1.3 The research questions

The goal of this research is to build an agent architecture that can be used in building agents for social simulations meant to study the effects of policies on SES. Such an architecture should account for human decision making in contexts where norms meet policy, while also remaining lightweight enough to be usable in ABMs. As such, it needs to provide explicit representations of the cognitive elements involved, and realistically replicate the normative deliberation process, and still remain scalable.

Building and implementing such an architecture is a challenging undertaking. Underlying concepts such as norms and values remain fuzzy concepts at best, despite – or maybe because – an abundance of exiting interpretations. This means we need to justify not only our choice to build our architecture on norms and values, but also our specific chosen interpretation of norms and values.

Furthermore, given the stated purpose of such an architecture, it is important to make the conceptual representation, as well as its implementation, as transparent as possible, and facilitate replication of all or parts of the models it is included in.

As a result, this research is divided into three main research questions:

RQ1: What are the particular challenges SES pose to governance and what role do norms and values play in them?

This is not a trivial question to answer, but it is fundamental for the development of any model aimed at studying the impact of policy on any complex adaptive system with a strong human component. The main difficulties arise from the fact that often the policy doesn't affect one homogeneous population with one targeted behavior, but rather many diverse groups with their own behavior sets. Furthermore, the same behavior is often connected to different norms and values, or the same norms and values express themselves in a range of different behaviors. We can observe behavior, but we can only learn about the norms and values involved if we talk to people, which is why stakeholder involvement and communication features prominently in answering this question.

As with many other matters relating to SES, answers can only be obtained on a case by case basis. However, guidelines exist about how to include research and discussions of norms and values in matters of policy compliance and design, as well as model design, and provide a good basis for investigation.

RQ2: How to conceptualize norms, values and policy for use in social simulations?

There are many answers to the question of conceptualizing norms as computational structures, and they all depend strongly on the goal they are developed for. These computational structures range from simple rules, to complex formal structures used in equally complex and formal reasoning processes for agents who are part of formal digital institutions or who assist in rule-rich fields such as legal systems [86]. Despite this richness, it is very difficult to find conceptualisations that work well with social simulations, either because they are too complex, not complex enough, or their structure doesn't mesh well with a reasoning process suitable to agents in a social simulation which includes the effects of policy [41].

In order to answer this question, we discuss existing normative architectures and approaches, and then explain and justify the choices we made for our own architecture, and how it is meant to be used.

RQ3: Does our architecture make agents behave in a more realistic manner leading to more realistic simulation results and more useful insights?

What we mean here by "realistic behavior" is agent behavior that is consistent both with theoretical assumptions about human behavior and with observed behavior in the real life context being studied. As an example, *homo economicus* agents behave consistently with theoretical assumptions of rationality, but often not with observed behavior.

This question can be broken up into two parts: "does the architecture generate more realistic results" and "does the architecture let us determine where the results come from". Our interest in the first part of the question is fairly self-evident: we always want our simulations to give us results that are as realistic as possible if we are to use it to learn anything about the system we model. The second part comes from the fact that, given the nature of the real-life systems being modeled, quantitative results may not be enough to understand the effects of a policy. It is equally important to understand how and why the results arise, and this cannot be achieved if the behavior of the agents is overly simplified or unrealistic, or lacks an explicit representation of the decision process and its relevant elements. Under such conditions, we'd be unable to trace back the reasoning behind the decisions of the agents in order to ascertain whether they arise in a plausible manner (from believable motivations, for instance). If we also consider that policy design for SES often includes a participatory stage involving diverse groups of stakeholders, it would be useful to be able to accommodate their their input and feedback into model development.

To decide whether the behavior of the agents is more realistic when we use our architecture, we will base our scenarios on a case study, described in [89]. 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. 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. Fishers would upgrade their vessels, but their size and capabilities had little to no impact on their fishing effort. 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. And many were not happy about it.

We use this case study to build a number of different scenarios, using the architecture to capture different types of agent decision making. We run the scenario described in the case study with with goal-driven agents, both economically rational and norm driven, and with value-driven agents, both community-oriented and self-interested. We introduce this distinction to illustrate the point that while norms might be enough to replicate certain behaviors, they are not enough to explain the full motivation of that behavior. By adding values, the generative power of the model is increased and with it, its explainability and, we hope, usefulness.

1.4 Limitations

This thesis deals with fuzzy concepts such as norms, for which don't have a clear definition, but rather many interpretations [31]. It is difficult to provide clear cut formalizations in such cases, so the reader shouldn't expect strict definitions or strong assertions concerning this subject, but rather discussions and interpretations. We do justify our chosen interpretations of the concepts, however these are also not very formal.

Part of this thesis is concerned with the role ABMs play in understanding SES policy and possibly serving in their governance. We are especially interested in their ability to include the perspectives of the different stakeholders and some implementation choices reflect our push towards a participatory modelling approach to developing these ABMs. However, due to time limitations, this aspect is only glanced over in this work, a thorough implementation and test of this approach being relegated to future work.

1.5 Thesis structure

The thesis is split into three major parts, each one seeking to answer one of the research questions.

The first part (chapters 2 and 3) deals with SES and their associated governance challenges and aims to offer a short overview of what makes them such fertile ground for wicked problems, as well as the modeling approaches being used to aid in understanding them. It sets up the context in which the first research question exists and illustrates the way in which norms, values and policy come to interact, as well the strategies used to identify and overcome possible conflicts or undesired side effects. This part is mostly aimed at those who want to model such a system and would like to have a starting point in their investigations.

The breakdown of this part is as follows:

Chapter 2 discusses the particular aspects of SES that make them so difficult to govern or even study, with an emphasis on fisheries, with the goal of showing that human behavior is an integral part of these systems and cannot be studied or modelled separately. The emphasis is on the challenges these systems pose to their governing bodies, especially when it comes to involving diverse stakeholders, as well as some of the strategies used to engage them into the policy design process. We are mainly interested in those strategies meant to identify the norms and values of the stakeholders and the ways policy may interfere with them, and bring them to the forefront of policy design discussions.

Chapter 3 talks about the kinds of models used to study these systems, with a focus on the goals of the research and the degree to which they include social aspects present in the system. Special attention is given to ABM and its advantages in capturing both a complex system and the complex social components (especially norms and values) present in that system. We also discuss participatory approaches to ABM, with considerations to their suitability for participatory approaches to policy design.

Part two of this book makes the transition from real world elements and social concepts such as norms and values to computational structures. It contains only one chapter, Chapter 4, which presents a detailed description of the architecture, together with explanations and justifications for the choices made in our conceptualization of perceived world states, actions, goals, norms and values.

Part three of this thesis (chapters 5 and 6) describes our implementation of the architecture and its use as part of a model sandbox. It sets out to answer the third research question by presenting a number of different model scenarios of the same real life context, and their results. This part is aimed at those who want in depth look how we implemented this an architecture, with considerations for participatory development of models, and/or want examples of how to use it.

The breakdown of this part is as follows:

Chapter 5 shows how we achieved the highly modular implementation of our architecture and how that facilitates the ability to create a visual model builder.

The models in chapter 6 are specifically designed to illustrate how the same architecture can be used to show the differences in results and explainability between models which include norms, values, both or none.

Finally, chapter 7 contains our conclusions and plans for future work.

Part I

Socio-ecological systems and governance

Chapter 2

Humans and nature - system conceptualisations

2.1 Socio-ecological systems

Many of the most difficult issues of the current era occur at the intersection of human activity and the environment. Climate change, obviously, is one of the biggest contemporary concerns, spanning the entire globe and getting more severe with each passing year. At smaller scales, we run into issues like overfishing, deforestation, the destruction of whole ecosystems due to land use, soils being wiped out due to industrial agricultural practices etc. On the other side of the coin, indigenous populations and traditional communities are seeing their ways of life threatened by the degradation of their environment and sometimes even by well-intentioned efforts to restore the same environment.

Efforts to curtail or stop damaging activities often come with significant economic or social costs. For instance, if we want to reduce carbon dioxide emissions, we could close down all coal power plants, which would require large investments into alternative sources of energy. It would also mean that most coal mines would close, which would deal a significant economic blow to mining towns which rely on income from the mines for their continuing prosperity. Without the mines, these towns often fall into poverty, which brings with it a host of social ills such as crime, addiction, homelessness or discrimination, none of which have an easy solution. Another example is conservation efforts seeking to establish protected areas that overlap with territories traditionally belonging to indigenous peoples and traditional communities. Such efforts often end up effectively banning them from continuing to practice their way of life, which can, again, plunge them into poverty, or result in their displacement. Policies aiming to establish sustainable exploitation of certain resources, such as fish stocks, can also disproportionately impact traditional and indigenous communities, even when they are not the ones engaged in over-exploitation. Even those living in urban areas at a remove from the immediate consequences of environmental damage have learned that there is no such thing as a life outside of nature when the COVID-19 pandemic swept the whole world in spite of the measures taken to contain it.

These systems that are characterized by intertwined human and natural elements are called social-ecological systems (SES). This concept arises from the idea that there is no meaningful separation between human society and ecological environment, and the two components cannot be properly studied when isolated from each other. The many elements that comprise these systems interact with one another in complex ways such that the system wide effects of any one of them cannot be reliably predicted, if at all. According to [108], such complex adaptive systems are organized according to six principles, each of which gives us some clue as to the difficulties one might encounter when studying them. These six principles are as follows:

- **relational organisation**: the behavior of the system is more strongly determined by the interactions between its elements, than by the characteristics of the elements themselves. This means that traditional reductionist approaches which aim to break the system into its components, then study each one individually in order to arrive at insights about the whole will not actually provide us with the desired results.
- **adaptive capacity**: the system can change its behavior in response to changing conditions. These changes are path dependent, giving these systems "history" and "memory" which need to be taken into account when attempting to divine possible future trajectories of the system. Methods that attempt to forecast possible futures starting from current conditions only are not as successful.
- **non-linear interactions**: small/localized interactions within the system can can have unexpectedly large results, up to abrupt reorganisation of the system. This is especially important when it comes to intentional efforts to change some parts of the system, as is the case when implementing conservation policies. The best intentions can have unexpected catastrophic effects down the line, and the damage can be difficult and costly to repair.
- **no clear boundaries**: these systems are not, and cannot be, isolated from other systems they interact with or are part of. This makes it difficult to even clearly define what an SES is because it is not immediately obvious whether certain elements are part of the system or the broader environment. Under these conditions, choosing which elements to include in a study will largely depend on its aims, with the knowledge that there is no one true combination of system elements that will result in the best possible outcome for that study.
- **context dependence**: many of these systems have developed strategies and institutions that become active during certain contexts. On the ecological side, such strategies include, for example, biological adaptations for surviving recurring but unpredictable droughts, forest fires, floods or freezes. On the social side, we have institutions such as martial law, which become active in times of crises, or semi-nomadic ways of life in which the community will settle down during predictable weather patterns (which are conducive to practicing agriculture), but will shift to a nomadic lifestyle either by relocating periodically or in times of crises such as crop failures. Being unaware that the system under study functions differently under different contexts can significantly reduce the effectiveness of such efforts.

- **complex causality and emergence**: the relational nature of these systems leads to complex networks of causation, resulting in properties and behaviors that cannot be understood by studying the elements of such systems in isolation. This makes it nearly impossible to guarantee that any given intervention into these systems will have the desired results.

As a result of their interdisciplinary nature, SES are often studied using methods that go beyond the boundaries of any single scientific discipline, either as modified existing methodologies or combinations of existing methodologies. Such studies, especially those aimed at supporting policy design and implementation, also require the cooperation of researchers from multiple disciplines, as well as non-academic stakeholders with expert or local knowledge about various parts of the system.

Due to the plurality of perspectives and theoretical backgrounds of those involved in researching SES, there isn't a single best approach to framing such systems. These frameworks differ in the way they conceptualize the elements and processes of the system, as well as the relationships and interactions between them. Which framework is more appropriate for a study depends on the aims of the research and the scientific background, knowledge and expertise of the participants. As such, the more fitting framework is the one which points out the elements most relevant to the research questions, such as system elements, processes, dynamics, relationships, interactions or variables. For a summary of SES frameworks, see [18]

2.2 SES framework

One framework that is particularly relevant to this work is the social-ecological action-situation framework (SE-AS) [125], which expands Ostrom's SES framwork (2.1) [98] developed for analysing common pool resource management problems, with a focus on institutions and resilience. Ostrom's framework introduces the action situation as characterised by the participating entities and all the elements that define their interaction, including rules, social and spatial elements etc. It then endeavors to identify social action situations within the common pool resource system, and collect and organise relevant variables that could plausibly influence the trajectory of the system, with the aims of guiding data collection and analysis, as well as creating a shared vocabulary that would make it easier to compare different case studies.

The SE-AS framework (2.2) expands on this by introducing socio-ecological and ecological action situations in addition to Ostrom's social action situations. In this way, the framework allows us to capture and describe interactions between only ecological, only social, and between both ecological and social entities of the system. Multiple action-situations can influence one another through their outcomes and thus give rise to the complex interlinked social-ecological phenomena characteristic of SES. This approach emphasises the relational nature of SES by pointing the focus at the connections between different system elements and on the effects

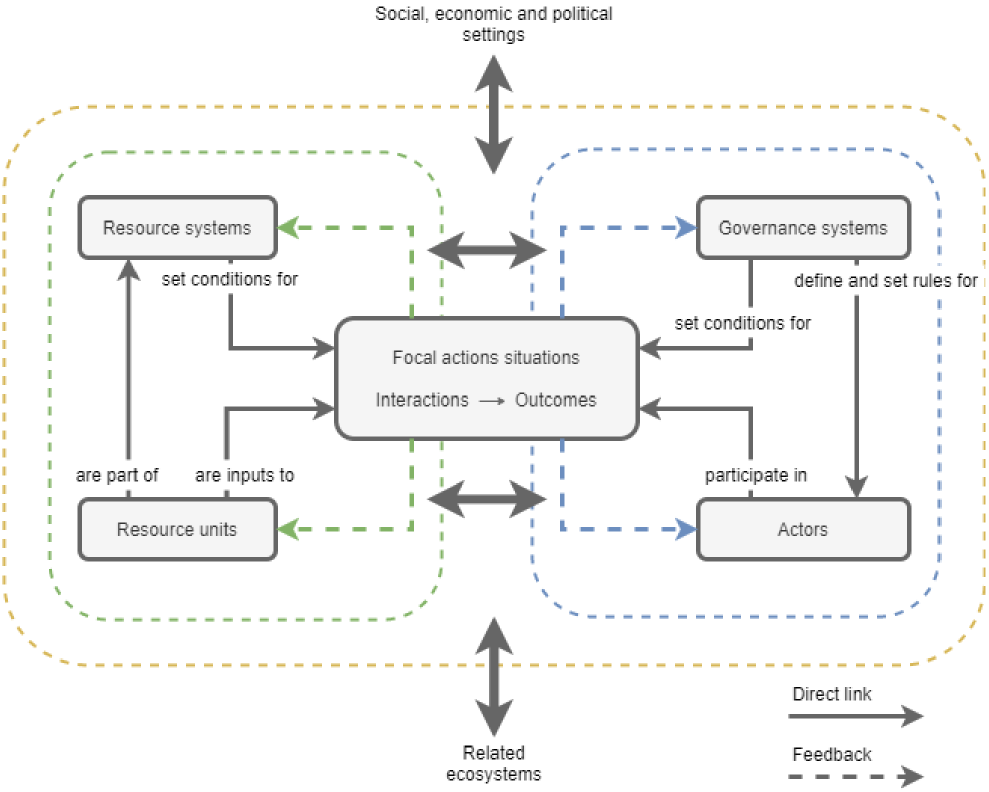


Fig. 2.1: Ostrom's SES framework, from [98]

that travel along these connections, effects which can affect both system elements and system rules.

The emphasis on interactions and rippling effects is especially interesting in the context of introducing policy into a SES. Policies aim to change the way social entities interact with other social entities, or with ecological entities. However, the action situations containing the affected entities are not isolated, but affect other action situations containing other entities, which may have been considered unrelated during the policy design phase. Under this new influence, the entities might change, changing the way they interact. Or vice-versa: the rules change and therefore the entities change. Regardless of the mechanism, the output of these action-situations will travel further into the system, affecting new action situations, changing them in turn.

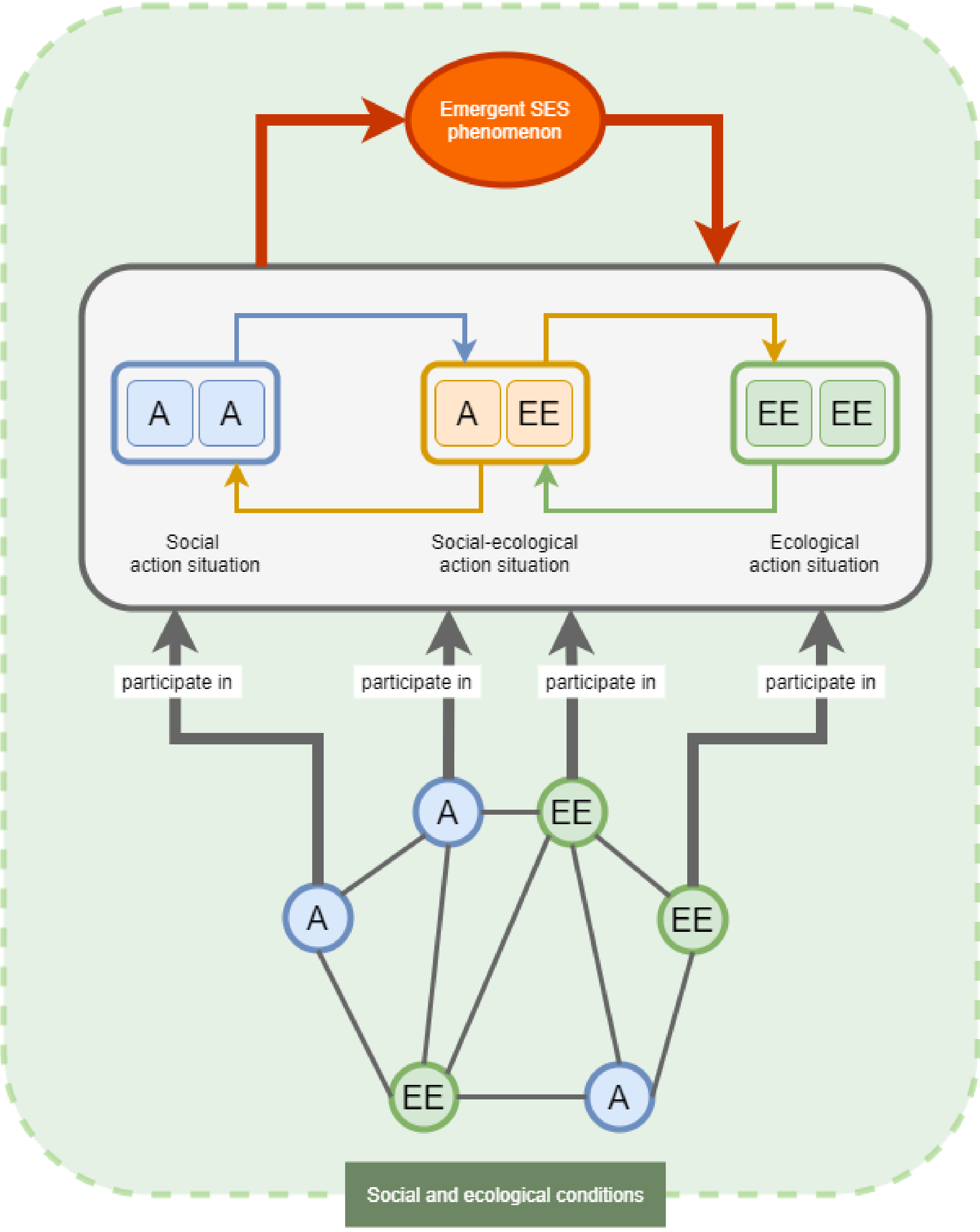


Fig. 2.2: SE-AS framework, from [124]

2.3 Challenges in SES governance

Through the lens of the SE-AS framework, it is easy to see why designing policy for SES can be an undertaking with very uncertain results, especially in cases where the system is only partially understood. Without a full understanding of the system, and the data to describe its current and past states, there's no telling where the ripples from a newly introduced policy might propagate, how far, and whether they'll eventually make their way back to the action situation that was targeted in the first place, continuing to introduce new unforeseen disturbances.

All this to say that governing SES is no mean feat. The fuzzy boundaries and the pervasive uncertainty of intervention outcomes or future system trajectories in general are enough to worry any policy-maker. However, the level and nature of the complexity present in SES and SES research present a particular kind of problem to policy-makers: the problem of too much data. Because such systems cannot be easily simplified and cannot be broken into smaller, easier to study components, they cannot be studied using traditional reductionist approaches. Unlike in the physics model of scientific inquiry (fig 2.3, adapted from [119]), there is no clear path from observation to experiment to theory. The system cannot even be clearly defined due to its fuzzy boundaries and open nature, not to mention that it can also change shape depending on the perspectives of the investigators and the goal of their research.

2.3.1 Too many answers

This creates a problem for policymakers because, as stated in [119], science and policy have fundamentally different goals. Science aims to gain knowledge and insights into the larger world, and goes about it by using the scientific method: observe, hypothesise, experiment, validate, refute, theorize. Achieving scientific consensus on an issue as complex and multifaceted as "how to manage SES" is impossible as long as research efforts barely agree on which methodologies are suitable and how to choose the most relevant ones in any given SES context. Scientists may agree that overfishing needs to stop and sustainable fishing practices introduced into fisheries around the world, but there's far less agreement on which practices to introduce where, and how to introduce them.

On the other hand, policy, at least democratic policy, is about reaching a consensus so that action can be taken. This consensus is achieved not through the scientific method, but through debate over competing interests, goals and values. Scientific findings, data and knowledge are brought in to support the arguments of one position or another, but they are not always enough to sway the final result. This is an especially prevalent issue when it comes to policy that targets SES because, in this case, science cannot give policymakers unassailable answers. SES are too complex, unpredictable, and multifaceted for any one single all-encompassing solution. They can, however, easily accommodate partial solutions whose effects on the larger

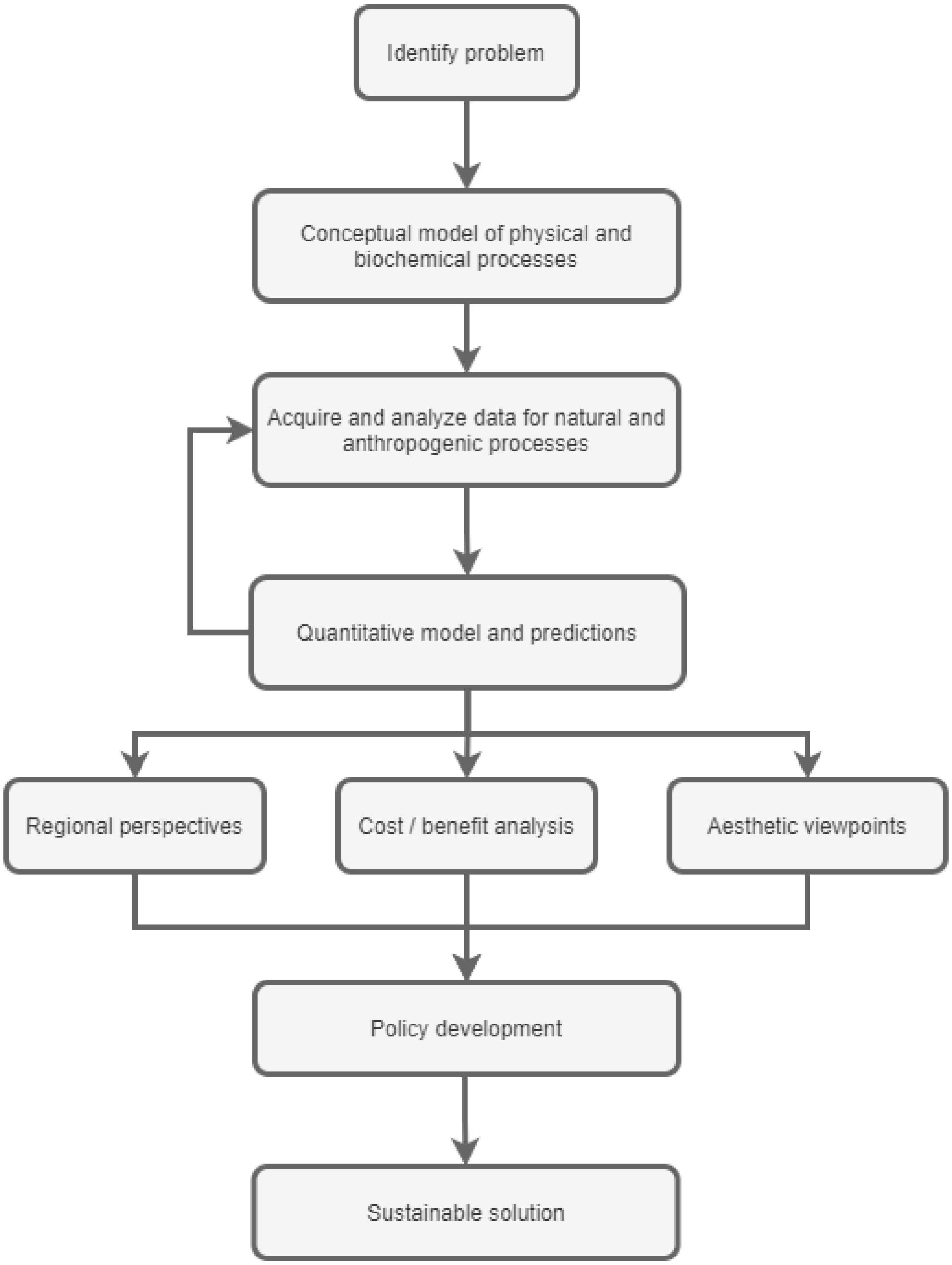

Identify problem
Conceptual model of physical and biochemical processes
Acquire and analyze data for natural and anthropogenic processes
Quantitative model and predictions
Regional perspectives
Cost / benefit analysis
Aesthetic viewpoints
Policy development
Sustainable solution

Fig. 2.3: "Physics" model, from [119]

system are uncertain or outright unknowable. Solutions which aim to redress economic ills may have adverse social or ecological effects. Solutions which would maintain a sustainable level of resource exploitation may carry steep economic costs or affect certain communities and peoples disproportionately. Solutions meant to address social ills such as poverty could result in the over-exploitation of local resources, solving the problem in the sort term, but exacerbating it in the long term.

For example, individual transferable quotas (ITQ) are a contentious subject in a number of fisheries around the world [30]. Iceland started gradually implementing this policy in the 1980s and it became a success story, according to some [10]. The problem of overfishing was declared solved, profits soared, the quality of the catch improved. However, these are only some of the metrics that can be applied in an SES such as Icelandic fisheries, and others argue that ITQs are not without a significant downside ([102], [15]). For Icelanders, fishing isn't just an economic activity, it is embedded in their culture as a symbol of independence and self-reliance. The ability to control their fishing grounds is important enough that Iceland declined to become an EU member in 2015 because, among other reasons, they could not obtain sole exploitation and management rights over their exclusive economic zone (EEZ). Iceland also entered a series of conflicts known as the "cod wars" with the United Kingdom over the size of its EEZ, and won every time. One of these conflicts involved the shelling and subsequent capture of a UK trawler by an Icelandic gunboat. Iceland even threatened to close the NATO base at Keflavik, which would have severely impaired NATO's ability to control the Atlantic Ocean ([16], [78]).

All this to say that Icelanders take fishing very seriously, beyond mere economic concerns. As such, even when ITQs have been successful by economic and sustainability metrics, they have had significant impact on Icelandic fishing communities that have historically relied on fishing as a way of life, without much concern for profits or economic efficiency. ITQs, however, are a market-based approach where fishers thrive when they are able to use economies of scale or, at least, manage fishing activities as a business first and foremost. Because of this, larger fishing companies and business savvy small-scale fishers have outcompeted the more traditionally inclined fishers, eliminating them from the fisheries - a process referred to as quota consolidation ([4], [47]). At the same time, quota is a highly valuable asset that younger generations prefer to sell (upon inheriting it) and use the money to move away from the small coastal communities. This leads to communities losing their fishing rights, and thus their source of income. Fish processing plants which depend on the fish from these communities, have to close, dealing another economic blow. Many Icelanders are not satisfied with this outcome and demand the current ITQ system be replaced with another that would be as sustainable, but more equitable ([28], [11], [81], [52]).

I use this example in part because I am familiar with it, having lived and worked in Iceland for the duration of the SAF21 project which funded this work, and in part to illustrate that even apparent "success stories" may have a less successful side. Iceland's situation regarding its ITQ policy is by no means unique as this type of policy seems to succeed in curbing the over-exploitation of stocks and over-capitalization of fisheries, but often fails with regards to social and governance outcomes such as equitable allocation of fishing rights, preventing power imbalances or quota accumulation, or maintaining social norms and cultural practices [76], [55], [91], [103]).

It is not trivial to balance between desired outcomes across the whole SES, and science can offer reasonable courses of action which can be used to shore up competing political viewpoints. Thus, policy decision won't be reached by pitting science against science, but by pitting values against opposing values.

2.3.2 The wrong answers

Of course, sometimes science is simply misused. In the case of SES, this can arise form the fact that researchers and decision-makers alike are not entirely aware of how the system they're trying to study and govern actually works. This leads to the wrong assumptions being used as the starting point for studies, to data and results being misinterpreted, to using oversimplified models, or even the wrong models.

Acheson [2] synthesises the many ways different governance schemes fail at effectively managing resources and concludes that there exists no universal approach to managing these resources. Instead, institutions need to be designed on a case by case basis, taking into account not only the type of resource, but the rich, complex context surrounding each instance of exploitation, including the behaviour, beliefs and preferences of the people involved. Using science to solve a policy related problem or to reach a policy related goal within an SES that is poorly understood is not only ineffectual, it can cause significant harm ([3]). This can happen either because the scientific methods used are poorly suited to the task (for instance, using a rational economic decision model when studying traditional communities, who are also strongly motivated by social norms or cultural practices), or because the information necessary is scarce, oversimplified or simply subject to a high degree of uncertainty.

Scarce and oversimplified information more often concerns the social side of the system, resulting in policies that may be sound on paper, but conflict with established social institutions, or introduce perverse incentives ([3]) resulting in outcomes that are far from the ones aimed for. For instance, fishery policy must rely on stock assessments, but they are notoriously difficult to measure. There is significant uncertainty when it comes to estimating how various factors, such as human exploitation, water temperature or various kinds of pollution, will affect stock dynamics. Sometimes it's difficult to even reliably distinguish the relevant factors in the first place [1].

Moreover, we shouldn't discount the hold ideology can have on both governing and scientific bodies, or the far reaching effects of policy designed and implemented with ideological blinkers on. For an analysis of some of the more egregious historical examples, read *Seeing like a state* [130]. Grand Utopian visions combined with misplaced confidence in oversimplified models of both human behavior and ecological systems have resulted in everything from failed attempts at farming forests and modernizing tropical agriculture, to failed cities, to immense human tragedy in the wake of forced collectivisation and mass relocation, all in the name of modernizing and improving society.

Such entrenched or unquestioned world views of the various participants in policy design can, sometimes inadvertently, sometimes by design, give disproportionate influence over the process to certain groups, rather than making sure all parties involved contribute according to their strengths. The results are rarely desirable [112]. Furthermore, these world views often shut out knowledge that doesn't comfortably fit [113]. This can be as simple as treating men as the standard human being and ignoring the ways in which women differ, resulting in policies, practices and technology that don't work as well for women as they do for men [105]. When this blind spot results in cell phones that are slightly too big for a woman's hand making them awkward to use, it's mostly an annoyance. When it results in running medical trials only on men or designing safety regulations only around the physiology of men, the consequences can be tragic [140]. Or it can be as significant as considering an economic system based on continuous growth the only one worth perpetuating, even when the goal is preserving finite, possibly non-renewable, resources [70].

2.3.3 Noncompliance and unintended effects

Yet another way policy can fail is through noncompliance. It is one of the more difficult elements to measure, because people are not usually eager to admit to breaking the law. Noncompliance is an issue for policy in any context, but it can be particularly difficult to understand and correct in the case of SES because their complexity make it challenging to track the problem to its roots. The applies to unintended effects, as well.

The problem of non-compliance has been widely studied and a number of frameworks exist to facilitate research and analysis [100]. These can be divided into actor-based, which emphasise people's motivation to not comply, and opportunity-based, which emphasise the role of the environment in people's decision to not comply.

Since we are aiming to develop agent-based models to aid in the policy design process for SES, we are more strongly interested in frameworks that center individual motivation as the source of non-compliance. These are the deterrence model [13], the compliance framework [111], and the theory of planned behavior [5]. The deterrence model assumes individuals are rational actors who will compare the benefits of breaking the law against the probability of getting caught and the severity of the punishment. The compliance framework assumes people comply more readily if the law fits in with their already existing norms and if they consider the legislative authority to be legitimate. The theory of planned behavior uses attitudes towards the non-compliant behavior, social norms associated with the same, and the difficulty of performing the behavior to predict whether someone is likely to be non-compliant.

When it comes to SES, one of the recurrent sources of non-compliance is the perceived mismatch between the motivations and methods of the governing authority and the motivations and methods of the governed communities. A failure to understand how and why people live a certain way can result in long strings of failures to implement successful policy ([67] [92]). These issues can be magnified if the people being governed are excluded from the policy design process [120]. Perhaps not surprising, people are more likely to comply when they feel like they can influence the policy design process and tailor regulations close to their socio-cultural preferences [65], so a participatory and empowering management regime can prove to be a more effective alternative to centralized authority [7].

However, opportunity-based frameworks of non-compliance are also highly relevant to SES, even if they fall outside the scope of our immediate interest. These frameworks represent non-compliant behavior as highly context-dependent [22], and in SES contexts, both environmental and social, can change in ways that are unexpected and unaccounted for.

2.3.4 Is evidence enough?

Evidence-based policy would appear to be the most rational way to govern, but the approach has found itself the target of criticism, especially when it comes to managing complex system like SES. The two main issues with applying evidence policy to SES problems are twofold. The first is that evidence-based policy works best when the evidence points to a clear solution (as is the case in many issues related to health care, for instance, although medicine is known for harboring some very thorny ethical issues of its own), and that is not necessarily the case when it comes to SES issues [66]. In a SES, any solution is always a solution "for some" [53], and any hopes that more data and more technology would somehow result in a solution "for all" are more likely than not to be misguided at best [95]. To further complicate matters, SES are large enough and complex enough that different sides of an issue can find evidence to shore up their own case (sometimes opposing sides can even use the same evidence [75]) and contest opposing cases [142]. So if scientific evidence cannot be the final arbiter, who can?

The second criticism is that scientific inquiry, especially scientific inquiry initiated at the request of policy makers, is not devoid of bias [136], which often manifests in oversimplification of key dynamics in the system (*homo economicus* springs to mind) or in the exclusion of other relevant perspectives [118], which are often dismissed as "uncomfortable knowledge" when they can’t easily be made to fit into the dominating paradigm [113]. This is an especially relevant criticism given that many SES problems and their proposed solutions have the potential (accidentally or by design) to cause deep, far-reaching and long-lasting changes in the properties and functioning of the system. These changes can easily cause imbalances of power, injustice and unethical practices, which either undermine the goal of the policies or lead to new complex problems [20].

To overcome these limitations, the policy design process should become more open and include the input and participation of more disciplines of science, rely on more than quantitative methods, and include more nuanced models of people, not simply as rational actors, but as beings with social norms and values [32, 118]. At the same time, the process should be aware of the scientific biases that can hide in different disciplines and make space for effective knowledge co-production strategies [147]. Furthermore, it should involve and empower a larger diversity of stakeholders to bring their own perspectives and participate effectively in the policy design process [104, 118]. This would shift the process from the physics inspired on from figure 2.3 to one more suited to the management of complex systems such as SES in figure 2.4.

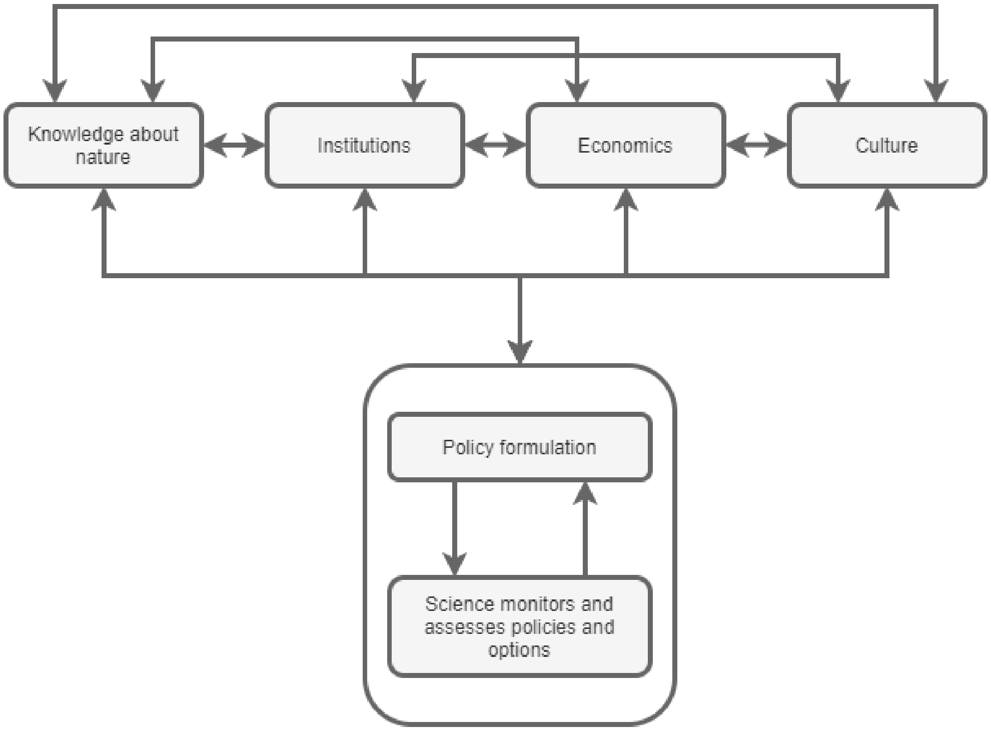


Fig. 2.4: "Geologic" model of policy, adapted from [119]

2.4 Conclusion

In this chapter we touched on what SES are and why they are difficult to govern. SES are complex system in which human society is deeply intertwined with the natural world. Fisheries, forestry, water and land use, national parks, all of these and more are examples of contexts in which human and natural elements are fused together to form a new, distinctly recognizable kind of system. They are not merely complicated, but irreducible. We cannot separate the human component from the ecological component to study them separately and expect to understand all the dynamics that might be relevant to whatever question we're asking or problem we're attempting to solve. Not only that, but they are also vast, open and their boundaries are fuzzy. The many elements that make up an SES are all connected in a web on interactions, cause and effect, and feedbacks. Where one web ends and another web begins is not really a thing anyone can determine with any certainty.

All this makes governing SES a challenging problem, that is nevertheless of utmost importance. Many of our most difficult contemporary problems arise in SES: overfishing, deforestation, damaging tourism, habitat destruction caused by urban and industrial developments, and, of course, climate change. We sometimes tend to focus on the ecological damage inflicted on these systems, but, in SES, ecological deterioration always entails social deterioration. Damaged ecologies harm the communities that depend on them for continued survival. At the same time, ill-advised policy can bring about social end economic changes that damage the environment, or can harm communities in an attempt to preserve the environment.

The complexity of SES means that evidence-based policy isn't always the best approach, for two main reasons. First, evidence-based policy works best when the problem it's trying to solve has a clear solution, which is rarely the case in SES. Second, scientific inquiry, especially when conducted at the behest of policy-makers, is not without bias. This means that, when it comes to devising policy for SES, it's more likely than not that policy makers will have to choose from multiple solutions proposed and backed by different stakeholders, each with their own agenda and priorities, and none of these solutions will lack support from scientific data and arguments.

Therefore, if evidence alone can't point to a winning solution, policy makers must turn to values, theirs and their stakeholders'. An alternative policy design process would include the participation of stakeholders and inputs from more scientific disciplines and rely on more methods than strongly quantitative ones. Such a process could ensure that different scientific disciplines cover one another's blind spots and correct one another's biases (most notably, overly simplified models of human behavior), while multiple stakeholder perspectives would ensure relevant local knowledge is brought to the table for consideration, and no single party gets free reign to impose their values, perspective and preferred approach on everyone else unchallenged.

Computational models are one of the most useful tools for policy makers, so they would undoubtedly be a part of this participatory policy design process as well. In the next chapter, we'll have a look at what kinds of models can capture the complexity of SES without resorting to overly simplistic conceptions of human behavior, and can also be a good match for a participatory approach.

Chapter 3

SES and agent based models

In the previous chapter we discussed the characteristics of SES and why they make policy design so challenging. One of the methods used to mitigate complexity that outstrips the unaided human ability to understand the dynamics of a system, to anticipate the effects of an intervention or to ascertain risks and uncertainties is modelling. Computational models have been around long enough that they now permeate any number of scientific disciplines, from physics to biology to economy to sociology, and display much variety of methodology, applications and purposes [24]. Computational models are used for prediction or forecasting, explanations and exploration of scenarios, understanding theory, illustration and visualisation and as analogies.

3.1 Agent based models in socio-ecological systems

Given the complexity of SES nothing short of multi- and trans-disciplinary research approaches can serve in their study. As such, there is a plethora of modelling options available to researchers, each with their usage context, and their own strengths and weaknesses. Among these, ABM has established itself as one of the more powerful tools available to researchers, especially in participatory settings.

ABM is at its most useful when analytical solutions are either too complex to be computed or lead to an oversimplification of the system being modelled. We want to use agents when the system displays a high degree of localization and distribution, it's important to capture the heterogeneity of individuals in a system and make the interactions between them visible [137]. Since SES often contain distinct groups of people, with their own interests, tightly interconnected among themselves and with an ecosystem, which itself operates at different spatial and temporal scales, it is not difficult to see why ABM would prove itself suitable.

To go into more detail, ABM has the capacity to integrate multiple dimensions involved in the policy design process [80]. It can integrate related, connected or interacting elements that would traditionally be treated separately by separate disciplines such as economy, sociology, ethnography or ecology, and in doing so it can integrate insights from all these disciplines, as well as connect models that address different process of interest from these disciplines, such as networks of kinship and influence, markets and ecosystem dynamics. It can also integrate processes that occur at different scales of time and space, which is important because the issues of interest, as well as the underlying dynamics and process that give rise to them, are guaranteed to operate on multiple scales in systems as vast as SES. And finally, but of no less importance, it can integrate stakeholder perspectives and integrate *with* stakeholders - to the point where participatory modelling has become widely accepted in SES modelling [134]. All of this makes ABM particularly well-suited for use as support in participatory policy processes such as those we touched on in the previous chapter and are described in figure 3.1.

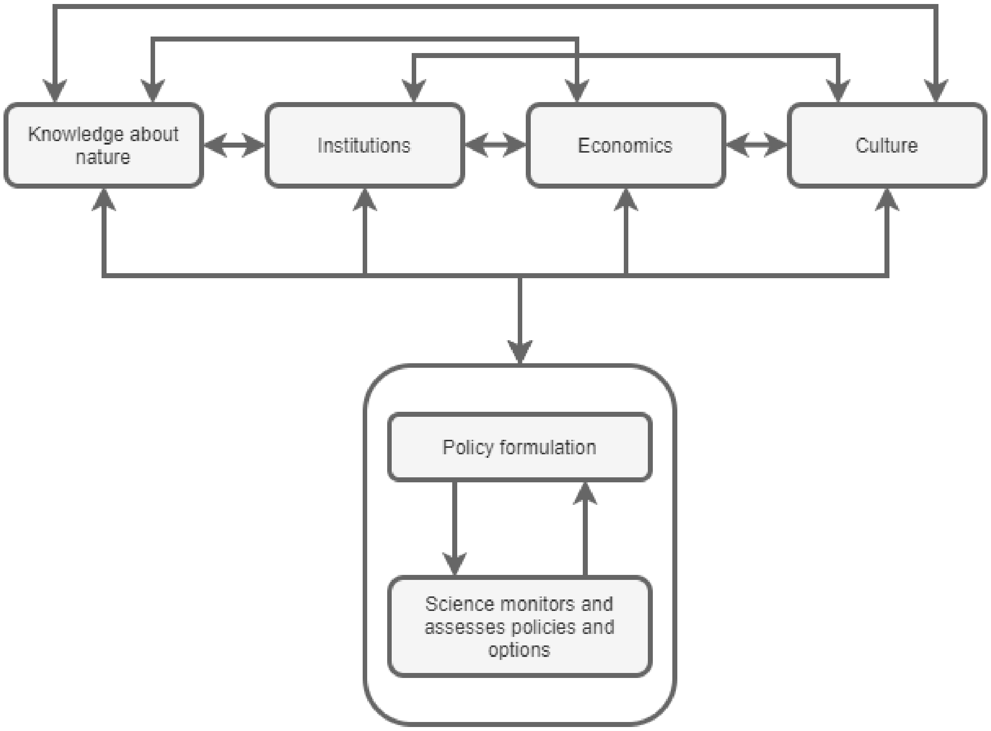


Fig. 3.1: Participatory model of policy reprised

The last part of the previous chapter briefly touched on the subject of participatory policy design, a process that includes stakeholders in the design and decision making process in order to bring together multiple points of view and a larger breadth of local knowledge with the hopes that the resulting policy would do a better job in serving more values than just those of the governing authorities. This doesn't mean science and its methods have no place at the table, just that it needs to make space for other perspectives (and we mustn't forget that science is not unbiased in the first place [66]). This opens the door for modelling approaches that also allow stakeholder participation and ABM has been quite successful in this regard [134].

3.1.1 The drawbacks of ABM

So far we've mentioned some of the strengths of ABM that are relevant in the context of modelling policy for SES. However, ABM is not without its drawbacks. The modeller's dilemma - deciding what aspects of the system to include in a model and at what level of simplification in order to make the model informative, while ensuring it doesn't become so complex that it is no longer understandable - can be an even more difficult proposition when it comes to modelling policy for SES [126]. These systems are complex enough and ABM is flexible and encompassing enough that it is very easy to either oversimplify or overcomplicate a model, rendering it uninformative either because it leaves out key elements and dynamics, or because it's too complicated to confidently interpret the results. On top of this, there is always a risk that the purpose of the model will be misunderstood and the model used for purposes it was never intended for [6, 43]. In the context of policy for SES, this distinct pitfall can be particularly easy to fall into since policy makers tend to expect predictive models [6], while most ABM for SES are built for the primary purpose of understanding the system [134].

Another issue of ABM that is of particular importance when it comes to supporting the policy process is that the field suffers from a not inconsiderable replication problem [19], which makes it difficult to reuse and compare models. This is partly due to the fact that modelling teams do not often stay together for more than one project and, thus, modellers have little incentive to develop reusable implementations of their models. It is far more convenient to build one-use models, publish, and move on to the next project. This leads to a proliferation of models that address very specific contexts, but have little to contribute to the larger issues when taken together [101, 116].

Compounding this issue is the fact that ABM documentation is not as ingrained a practice as it should be, nor does it guarantee the model will be properly replicated, despite the existence of a shared documentation protocol [62] that has been repeatedly updated over the years in order to cover more and more model elements and dynamics. Even small differences in how a model is built and implemented can change the results to a significant degree [93], and hidden assumptions that are not made explicit in the documentation can undermine the whole effort [118].

3.1.2 The (underutilized) social elements in ABM

For all its power, the ABM approach in SES has been criticised for its underdeveloped social element [134, 123] (although, to be fair, the same criticism is sometimes aimed at SES research in general [117]). This might come as a surprise to people familiar with the considerable diversity of human decision making processes represented in ABM used in SES and other conceptualizations of coupled human and natural systems [8]. However, this criticism does not concern a lack of variety, but rather one of complexity and nuance. In particular, two social elements stand out as promoters of social behaviors: norms and values [41]. They are considered of interest, among others, for postnormal conservation science [32], for assessing the popularity of proposed policy ([36, 138, 121], for example), for overcoming some of the limitations of evidence-based policy [118], or for framing and conceptualizing stakeholders' perspectives and lived experiences [133, 92, 139, 60].

3.1.2.1 Values

Values are commonly understood as principles or ideas that are important to people and that are stable over most of the course of someone's life [129], which is exactly as fuzzy a definition as it sounds. As a consequence, there are a number of conceptualizations of values that can be used in SES research and modelling [139]. Among these, we have:

- Universal values [128]: this research showed that all human values across cultures can be classified under these categories: power, achievement, hedonism, stimulation, self-direction, universalism, benevolence, tradition, conformity, security. Furthermore, these categories are related to one another, represented by placing the values in a circle (see figure 3.2). Neighboring values reinforce each other (promoting one promotes the other), while opposing values attack each other (promoting one weakens the other),
- Intrinsic values [90], which include life, health, truth, pleasure, beauty, freedom, power etc.
- Lived values [60], which include health, safety, belongingness, esteem and self-actualization
- Noneconomic/invisible losses [131], which include invisible losses (identity, life style, culture, knowledge etc.), intrinsic values (biodiversity, health, dignity etc.) and instrumental values (productive land, education, tradition, social bonds, sense of place etc.)

When it comes to the natural world, the relationship between people and nature can be classified based on the values associated with it [94], which in turn determine the goals and practices connecting the human and ecological elements:

- Detachment: society separate from nature, preference for urban spaces
- Dominion: society separate from nature, preference for human control
- Devotion: society not separate from nature, nature is perceived as sacred

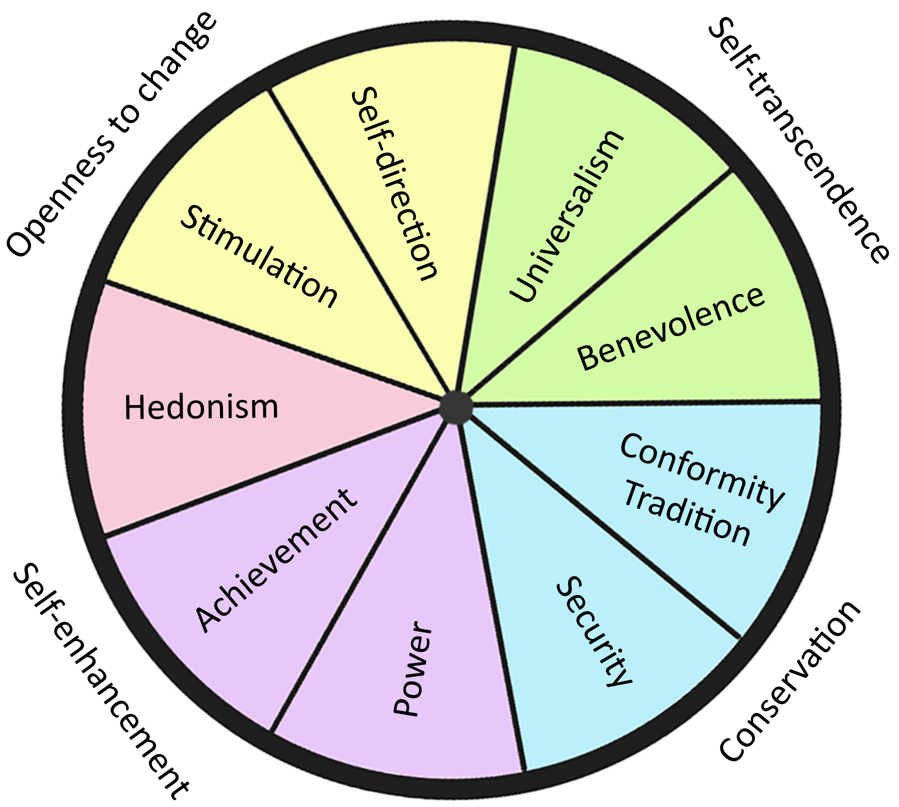


Fig. 3.2: Schwartz value wheel. Neighboring values reinforce each other (promoting one promotes the other), while opposing values attack each other (promoting one weakens the other)

- Stewardship: society not separate from nature, preference for human restraint
- Wardship: society separate from nature, nature to be protected
- Ritualized exchange: society not separate from nature, nature as interactive agent
- Utilization: society separate from nature, nature as source of goods and services

Using just this classification as an example, we can already see that the dynamics arising at the intersection of the natural and the human are diverse and span the range from mutually supportive to antagonistic. Models that include ways of representing just this aspect of human values would already gain considerably in expressivity and explanatory power.

3.1.2.2 Social norms

Social norms are another social concept that shows up often in SES and policy literature, and would allow models to capture very relevant human behaviors and decision processes. It is, unfortunately, just as fuzzy a concept as values, with multiple coexisting interpretations of what they are, what they do and what purpose they serve [31]. However, they all share a common thread, which is that social norms are collectively agreed upon behaviors and goals that are considered proper and/or expected in certain contexts. We are also interested in social norm conceptualizations that connect them explicitly to values, in particular Schwartz's interpretation in which norms promote values in specific contexts [127].

Taken together, values and social norms form a particularly stable and consistent framework for the decision making processes of an agent, encompassing both motivations and preferred means of pursuing said motivations. Stability over time is essential because much of SES management deals with time intervals of years and decades. The less the decision machinery of any one agent changes during that window, the more confidence we can have in the model. In particular, Schwartz values [128] and their associated norms [127] can span the gap from abstract to concrete by definition, allowing modelers to strike the right balance, on a case by case basis, between too general and too specific without getting lost in details or losing model specificity. Policy, as another kind of norm, fits in this framework as either supporting/reinforcing (when it promotes the values of an agent or works together with the social norms of an agent) or antagonistic/conflicting (when it goes against the values of an agent or conflicts with and the social norms of an agent).

3.1.3 Norms and values in agent models

Since they are wonderfully useful concepts, norms and values have not been ignored by the modelling community. Norms in particular have been included in quite a few agent architectures, sometimes together with policy, with different representations, effects on behavior and circumstances under which an agent can choose to break a norm [86]. Among these, three architectures stand out in particular [12]:

- Deliberative Normative Agents [26, 37]: norms are too complex to be represented as implicit constraints on behavior and instead need an explicit representation for agents to reason with, and be able to break as desired.
- The EMIL-A architecture [25]: it allows agents to learn about norms in their society, internalize norms and use them in their decision processes. In addition to norms, agents can have normative knowledge, normative goals and normative intentions

- The NoA agent architecture [82]: the definition of norms is broader than in the other architectures, allowing for obligations, prohibitions, legal powers, immunities etc. Agents have normative states that they use to determine their behavior. Norms are activated in context

Despite the numerous kinds of normative agent architecture available for agent models, many of them are rather abstract and more focused on their chosen conceptualization of norms than on agent behavior overall. This is not necessarily surprising, since most of them come from MAS, where their main concern is using agents to solve specific problems rather than simulating societies of agents under environmental, social, economic or political influences [146]. Unfortunately, this means that they are not necessarily well suited to social simulation.

Agents with values are more equitably represented in both MAS and ABM, but value architectures are not as numerous as normative architectures. Notably, values are used to determine an agent's preference over world states, which in turn determines its behavior [143], or values are used to directly determine an agent's preference over behavior [68]. Both of these use Schwartz values [128] for their solid empirical foundation and underlying circular structure (see figure 3.2). However, they are used for entirely different purposes. The value architecture in [143] is used to construct arguments for why a course of action is preferable to another, while the agent architecture in [68] is used to construct agents for social simulations of fishery management. The later is also expanded to include norms and is used to look at their life cycle under the influence of values [69].

3.2 Conclusion

The previous chapter explained why studying and governing SES is challenging, if not outright daunting. The current chapter argued why ABM are uniquely well suited to help. ABM is at its most useful when analytical solutions are either too complex to be computed or lead to an oversimplification of the system being modelled. It can integrate related, connected or interacting elements that would traditionally be treated separately by different disciplines of science, as well as processes that occur at different scales of time and space. And, especially relevant to the participatory policy design suited to SES governance, it can integrate stakeholder perspectives and integrate *with* stakeholders - to the extent that participatory modelling has become widely accepted in SES modelling.

Despite all these advantage, the field still struggles with some appreciable deficits, such as issues reproducing, comparing and reusing models or parts of models, and while there is progress being made with regards to the inclusion of nuanced social elements and decision processes, there is certainly room for improvement. The dearth of realistic social and decision models is particularly troublesome if ABM is to be included in the policy design process for governing systems in which the human component is crucial to the functioning and behavior of the system. If the models fail to capture it properly, the results will lack informative power or can be downright misleading.

In matters of experimenting with policy alternatives in particular, agents need to have a decision process that can operate with social norms and values, at least. Values describe what kind of world states the agents prefer, and are extremely stable over time. Social norms are collectively agreed upon behaviors and goals that are considered proper and/or expected in certain contexts. Taken together, values and social norms form a particularly stable and consistent framework for the decision making processes of an agent, encompassing both motivations and preferred means of pursuing said motivations. Policy, as another kind of norm, fits in this framework as either supporting/reinforcing (when it promotes the values of an agent or works together with the social norms of an agent) or antagonistic/conflicting (when it goes against the values of an agent or conflicts with and the social norms of an agent).

In the next three chapters, we'll describe the conceptualization and implementation of our own normative value-driven agent architecture for use in social simulations of policy effects in SES. Our goal is for our architecture to exhibit a few desirable traits, which would not, by themselves, solve all the issues mentioned, but aim to support more realistic models, which would, hopefully, be more informative to stakeholders and policy makers alike. These traits are as follows:

- Explicit and flexible representation of norms, values, and any other cognitive or social component of the agent's decision making processes: this maintains model transparency and explainability, makes it possible to easily track the decision process from motivation to behavior, keeps the implementation modular, and therefore more scalable in terms of complexity
- Sandbox approach as support for multiple related models as part of the same project: since we aim for significant modularity of the model and agent architecture, cognitive elements such as norms, values, goals or actions, and their associated decision processes can be reused in multiple models with minimal additional effort
- Support for reusability, comparison and reproducibility of models and model elements: this should already be accounted for within a project sandbox by virtue of the intended modularity of the model and agent architecture. In order to maintain these properties outside the boundaries of one project, the models and their elements needs to be easy to save and disseminate.
- Low barrier to participation for stakeholders: since participatory modelling is becoming such an important part of modelling and policy design for SES, we should make it easy for stakeholders to build, modify and interact with the models, which can be achieved by harnessing the above mentioned modularity and building a suitable graphic interface.

Part II

Conceptual model

Chapter 4

The Agent Architecture

In previous chapters we've had a look at how SES are conceptualized, what kind of models are used to study them, as well as how stakeholders are involved in the research and modeling process. Given the complexity and diversity of SES, it's not a surprise that there's room for many different approaches at this research table, and that is before we factor in that a diversity of stakeholders come to the table with a diversity of perspectives. Accounting for, and integrating, all these elements is a herculean undertaking, which is why most research carves away a slice of one SES (or one kind of SES) to focus on. However, we also talked about ABM and its ability to deal with more complexity, especially when it comes to systems in which the heterogeneity of the agents is paramount to understanding their emerging dynamics and overall behavior. In this context, ABMs might be less accurate when it comes to prediction, but they are unsurpassed when it comes to experiments and investigation.

Unfortunately, it is precisely at this point that ABMs run into a bottleneck: the speed and scope of experiment iteration. It is fairly painless to run any number of experiments if all that changes from one to the next are parameter values. However, if what we mean to change are the rules of the agents' behavior, or worse, their cognition, the experimental process slows down considerably.

One of the reasons, the one we'll be focusing on going forward, is that many of the behavioral or cognitive elements involved tend to only be explicitly described in the most theoretical sense, and the closer the model gets to its practical implementation, the less explicit these elements become. Instead, they morph into a series of rules and equations that describe a very concrete scenario, losing their general form and, thus, their applicability outside the narrow scope for which they were made concrete in the first place. What this means for the modeling process is that all parties involved must sit down and agree on this initial scope and any further refinements of the model must remain within its boundaries if the process is to continue apace.

Let's return to our fishing case study to illustrate our meaning. We are aware of two main kinds of fishers in the system, some who look at fishing as a business to be run efficiently for profit, and some who look at it through a more nuanced cultural way, a way of life with its own rules divorced from considerations of maximizing profits or efficiencies. In this system we are considering one particular policy transition, from licence-based fishing or open-access fishing to a quota-based system. Since this already happened, we can use existing data to infer the reaction of each group to the new policy so we can easily implement a rules-based model that would reflect that behavior. We wouldn't learn much from it, since we're putting in the rules we already observed so that the model can reflect them back at us. We could use normative agents to simulate the transition, which would let us ascertain whether the observed behavior was due to the norms the fishers adhere to. We could push it further and use value-based reasoning, which would let us look at different combinations of value priorities and value concretizations and see which of these are most likely to be at play in the communities we're studying.

But if we first work out these behaviors on paper in order to reduce them to a series of "if-then" statements liberally peppered with equations, we're making it very difficult for ourselves to move beyond these fisher groups and this policy scenario. If we put the model out and then we get asked "what if we don't do quota, what if we instead try territorial rights?", this new policy comes with a whole lot of new possible behaviors arising from different motivations. Does the notion of "territory" map onto values the same way "quota" does? Did we ever fit the notion of "sharing" and "accountability" in the normative space of these groups? Is it possible that some fishers who operate as self-interested business owners when fishing is framed as an individualistic activity would find themselves operating in a richer cultural decision space once the activity takes on explicit communal elements?

Fitting all of this into the existing "if-then" decision trees of our model would be a cumbersome endeavor so we'd probably end up building a whole new model, which would require us to figure out the concrete rules of the agents' behavior again, implement them again, test the model again.

And then: "what if we consider some market policies too?"

You get where this is going. Modeling scenarios where we watch behavior change because the *rules of behavior* change takes a lot more effort than modeling scenarios where we watch behavior change because the underlying parameters change. It's not adjusting a few sliders, it's building whole new models.

4.1 Agents

ABM is one of the more suitable tools for investigating policy effects *in silico*, but not all agents are created equal. In its most abstract form, a deliberative agent is an entity that can perceive both its internal state and parts of the world it inhabits, has preferences over both, can perform operations to alter both, and can reason about which of the operations it can perform best suit its preferences. Every part of this description can be, and usually is, subject to fairly severe constraints, either by design, by technological limitations, or both.

When it comes to ABM, these constraints are severe mostly by design as ABM is known for preferring the simplest possible agents that appear to fit the model. The world is reduced to whatever features and entities are deemed most relevant, and the agent can act upon it in very narrow, rigid ways. Their behavior, even in the more sophisticated approaches, is mostly inflexible, predefined. This is not usually an issue, since the whole point of those models is to study the effects of the predefined behavior. In many of these cases, using the simplest agents, built on the simplest assumptions, means resorting to 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(fig. 4.1).

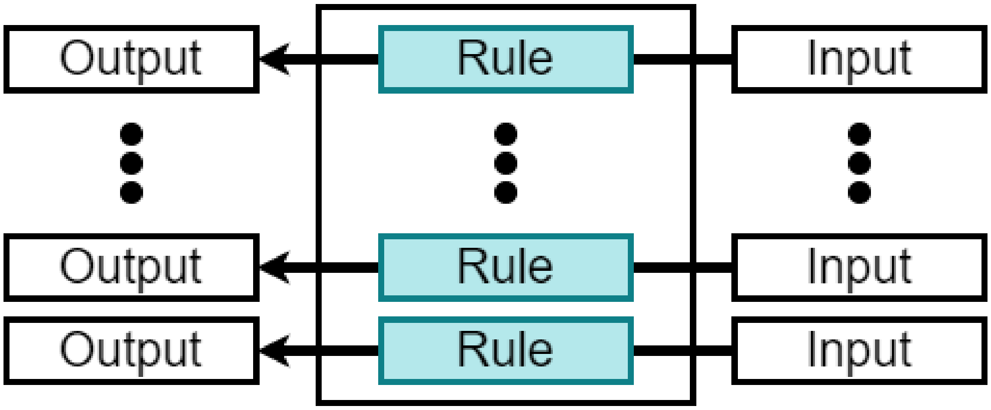


Fig. 4.1: Simple diagram of a rule-based architecture

This approach to building ABMs is not without criticism, especially when it comes to models that include influential social components. In these cases, the preference for simplicity has been called out as under-exploitation of the possibilities afforded by the ABM approach, especially the generative aspect of it, and calls have been made for abandoning simplicity for more descriptive models, which would allow the use of richer empirical data, including qualitative data [33].

When it comes to modelling policy effects, not only are we modelling a system in which social components have a significant influence on the results, we are also introducing a type of perturbation that changes the very rules by which the agents operate without necessarily knowing in advance what form these changes may take. Indeed, discovering these changes may well be the point. In this context, the glaring issue with ad-hoc rules is that, lacking higher order motivations, they offer no basis from which to infer future behaviour. The best we are left with are assumptions, that are sometimes hidden, and sometimes are based on theory whose applicability is not fully justifiable, such as the ever popular *homo economicus*.

To look at a concrete example, let's go back to our case study again. If we start from the assumption that fishers are rational profit maximizers, the increase in fishing effort for small-scale fishers is not really surprising since rational agents will try to extract as much profit as they can within the limits of the law. What's surprising is that the fishers were not pleased about it and that they weren't fishing way more under the previous policy, which put basically no restrictions on how large their catches could be. In reality, fishing behavior in the almost 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.

In order to build a model that explains this behavior, one that goes beyond assumptions that people involved in economic activities are either profit maximizers or automatons quietly following a set of predefined rules, we need to be able to build agents that can hold and reason about complex motivations (fig. 4.2). In order to study the effects of policy on such agents, policy cannot simply be one more set of rules, but rather one more motivation ([38]).

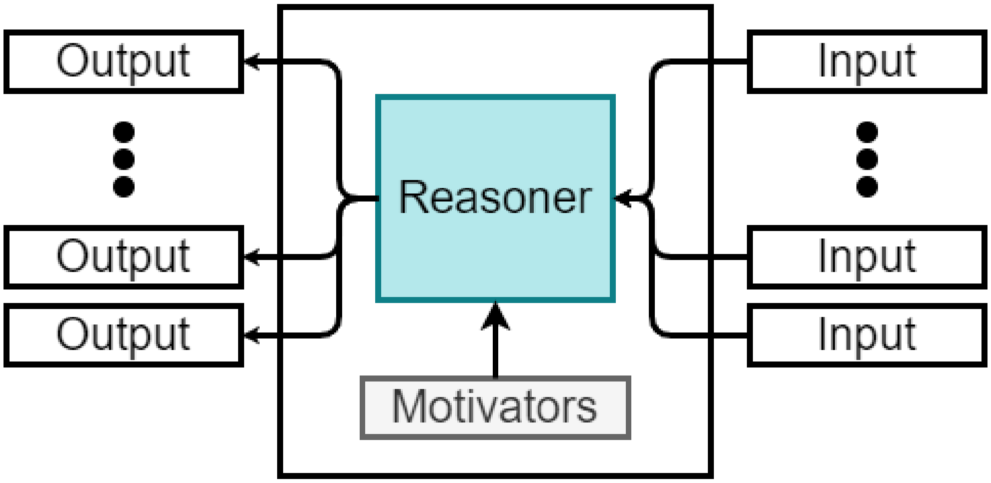


Fig. 4.2: Simple diagram of a deliberative architecture.

Since complex cognitive agents already exist (see 3), it is tempting to think that it would take fairly little effort to adapt them for social simulation. However, although 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, it 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 it 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 a real world system are the most relevant for their research questions, and thus for the model and agents theyre 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, 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, without unnecessarily sacrificing the sophistication of normative deliberation.

4.2 A sandbox approach

ABMs are often described as "able to handle far more complexity than mere mathematical models" and this is true, but what's not often discussed is how far more and what that means in practice. Agents can be very simple, but numerous, and together give rise to emergent phenomena. Schelling's model of segregation ([122]), one of the first agent models, is an excellent illustration and it's probably why no ABM course or tutorial can help but refer to it. Or, agents can be complex enough to run a robot body over rough terrain and adapt to circumstances in order to complete a task, as is the case with the fairly well known Boston Dynamics robots ([110, 58]).

Despite this breadth of complexity, ABMs often fall towards the low-complexity end of the spectrum. This is mostly due to the goal they are developed for. For instance, many ABMs are built in order to investigate the mechanisms which give rise to a particular observed phenomenon, especially in the sphere of ecological and natural sciences. Reynold's flocking model ([114]) is practically ancient in computer science years, but remains one of the best examples of how a few very simple rules are enough to explain complex behaviors. The strategy of pattern-oriented modelling ([63]) arose from ecological ABMs and describes the practice of running models with different underlying rules to see which of the proposed mechanisms give rise to the desired system dynamics.

Pattern-oriented modelling came about partly because we cannot ask animals why they behave the way they do. Their internal motivators and reasoning is hidden. When it comes to building models of human behavior, the same limitation often applies, not because we cannot ask the people involved what their reasoning and motivations are, but because doing so takes considerable time, effort and money. However, if we extend this pattern-oriented modelling approach to the social behavior of agents with complex motivations and complex reasoning processes, we can build a series of models that test a range of assumptions about what drives the people in the real life context we are investigating. In order to do this effectively, we would need to build agents that can be customized not only with different values or different norms, but agents that can function without values or without norms. After all, different decision contexts include different motivators and different reasoning processes, from habits to social practices to norm and value reasoning, and we don't necessarily know beforehand which of them applies to our modelling context.

As such, what we want is a modeling environment in which we can easily build many agent types with different combinations of motivations and reasoning processes, which we can then put into a simulated world and subject them to different policies and policy changes in order to observe and analyse their behavior over time. This is what we call a modeling sandbox, and both our conceptual architecture and its implementation have been designed with this goal in mind.

4.2.1 Hard vs. soft coding

Hard coding refers to the practice of embedding data directly into the source code of a program and requires changes to the code when the aforementioned data changes. Soft coding, on the other hand, is the practice of embedding arbitrary data in the code, which is then made specific at run time through user input, loading from files, databases etc. Agent modelling tools like Netlogo or Repast already allow the soft coding of certain types of data, mainly parameter values. In the code, they are represented as variables and get their values at runtime from the user interface or customization files. This is often sufficient in the case of agents with a fixed set of behavior rules. If the rule set doesn't change from one scenario to another, hard coding the rule set is a perfectly serviceable practice.

However, when dealing with normative and value agents in scenarios with changing policy, the immutability of the rule set is not a given. Different policies often introduce new rules and concepts, rather than just new parameter values, and different agents may have different motivators and behavioral capabilities. To complicate matters, some scenarios can require changing the policy mid-simulation, either from a "no policy" condition or from a different starting policy. In these models, hard coding the rule set is possible, but it virtually guarantees the codebase can only handle a small number of scenarios before becoming difficult to maintain or expand. Every new rule, even if it's not active in all scenarios, must be included in the branching logic of the decision algorithm. The more branches to keep track of, the more opportunity for mistakes or oversights, which can then result in unsound agent behavior. Simply coding new agent types for every rule set included in the simulated scenarios is also not ideal, since it's time consuming and can also spiral out of manageable bounds if there are many agent types and many policies to experiment with. This can be especially bothersome if the modeling process is an open-ended participatory one, where each simulation experiment elicits ideas for new experiments.

To illustrate, let’s consider a simulation experiment which includes profit-motivated agents and two types of normative agents operating under an effort quota policy. This policy can lead to "Olympic fishing", where fishers increase their fishing capacity and efficiency as much as possible to get the most catch during the limited number of fishing days. Since this situation is often characterized by overfishing, two alternative policies are proposed for experimentation: a transferable catch-quota policy and a territorial use rights policy. These new policies are fundamentally different from each other and from the old policy. It’s not just a matter of changing the number or distribution of fishing days, or limiting the size and gear of the ships. They introduce new concepts such as *catch-quota* and *territory*, new actions such as *buy quota* and *sell quota* or *monitor other fishers*, and new rules such as *sell quota if* ***end of year*** *and* ***quota remaining is more than zero*** or *administer punishment if* ***observed infraction***. All of these will need to be coded into all agent types, even though, only two policies will be active during the simulation. And, at any time, a request may be made for yet another policy to be introduced for experimentation, for instance a rights auction policy, which in turn introduces its own concepts such as *auction*, actions such as *bid* and rules such as *bid if* ***current cost*** *is lower than* ***available funds*** or *bid if* ***expected value of catch*** *is larger than* ***cost***.

What this example shows is that representing features of interest by using variables is not going to offer any easy gains for behavioral flexibility. What is needed is a different, more general purpose representation. Much of the development effort that goes into coding a suitable model for the aforementioned scenarios can be simplified if the cognitive elements such as actions and norms could be partially or fully soft coded. In order to achieve this, the code must extend arbitrary data from simple parameter values to more complex structures encompassing both variables and rules.

4.3 Architecture requirements

The type of architecture we are interested in must include behaviors, motivators of behavior, as well as the reasoning necessary for deriving behavior from motivations. These components must have explicit representations and can be hard-coded only to a limited extent. In making the motivational and behavioral elements **explicit**, we are then able to separate them from the reasoning process of the agent. The entire point of this architecture is to allow the agent to reason about its motivations and this is more readily achieved when the motivators are explicit and are seen as elements the reasoning processes **operates with**, rather than **parts of** the reasoning process.

To be more accurate, we impose the following requirements:

- agents must have an internal representation of full or partial world states
- agents must have an explicit internal representation of the behaviors they are able to perform, which we will call actions
- agents must have an explicit internal representation of their motivators, in our case goals, norms, policy and values
- agents must be able to map their motivations and behavior onto their internal and external environment
- agents must be able to prioritize their motivations, assess their current ability to perform behaviors and determine which of their available behaviors best suits their current motivations

Since we want our agents to be able to act even in the absence of some types of motivators, we impose one more requirement on the architecture:

- the agent must be able to operate with any plausible combination of motivational elements

This requirement is explored further in this chapter in section 4.4.6, and is illustrated in chapter 6 through models containing different types of agents, including agents with only goals, agents with norms and goals, and agents with norms and values.

4.4 Architecture

Our proposed architecture, pictured in fig. 4.4, is based on the principle of composition (see fig. 4.3). Rather than attempt to build one rigid structure that would conceivably cover as many types of simulation scenarios as we could envision at the time of development, we chose to focus on identifying, defining and building the parts we'd need in order to assemble a variety of agents on the fly. These parts, and the rules for putting them together, are described briefly bellow and in more detail the following sections.

We define a world state as a set of entities, which in turn are sets of attribute-value pairs [1]. The structure of any entity is not fixed, but can be altered at any time by adding or removing components from its component set.

An agent contains beliefs about current and possible future world states, a set of behaviors (actions), multiple sets of motivators (goals, norms, values and/or policies), and a reasoner which can operate with the agent's motivator types. The structure of the agent is also not fixed. It can be altered by altering its behavior set or any of its motivator sets (including removing or

[1] These are referred to in chapter 5 as component-value pairs, because we borrow from the entity-component-system architectural pattern. Conceptually "attribute" has the stronger connotations, but computationally, "component" is the term that best fits with the approach, hence the differing terminology

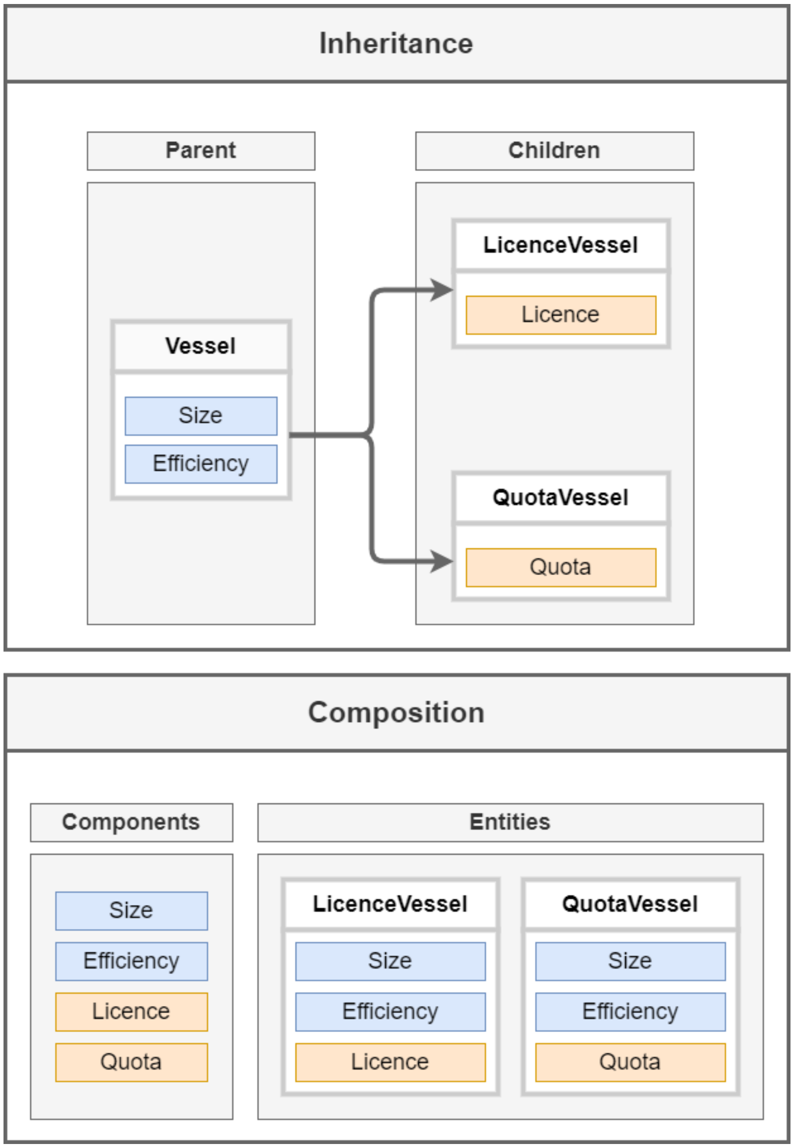


Fig. 4.3: Inheritance vs. composition

adding whole sets of motivators), and by swapping out the old reasoner with a new one that can operate on the types of motivators the agent currently has.

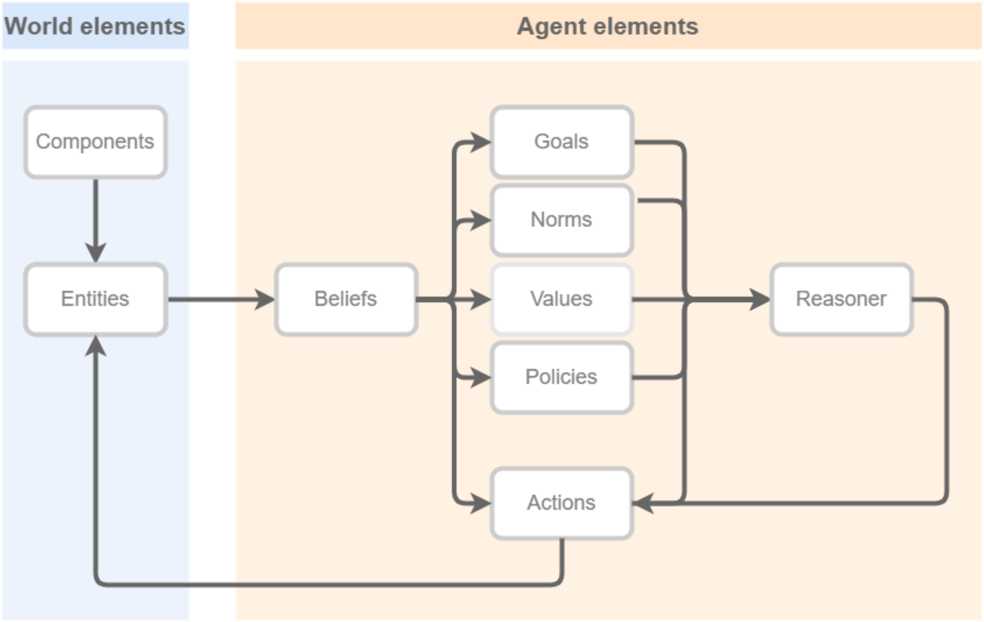


Fig. 4.4: Diagram of proposed architecture

4.4.1 World states

The first step in building the agent architecture is giving structure to the world. We use attributes gathered into meaningful groups – called entities – to describe the elements of the world. These attributes are not necessarily unique to one entity and can be combined and re-combined as needed to create all the entities that are part of the world. If we describe the world as a collection of world states $W = \{w_1, w_2...w_n\}$, then an attribute c takes the value $c(w)$ in world state $w \in W$. Then we can say entity $e(w) = \{c_1(w), c_2(w)...c_n(w)\}$. The collection of attributes $e = \{c_1, c_2...c_n\}$ is called the entity's signature (or archetype).

Using an example to illustrate, let's consider the concepts of $catch$ and $vessel$. we can describe

$catch(w) = \{weight(w), species(w), price(w)\}$ and

$vessel(w) = \{weight(w), length(w), capacity(w), efficiency(w), price(w)\}$.

The attributes $weight$ and $price$ repeat in both concepts and share the same meaning. Ideally, any other entity that shares these attributes will also preserve the meaning, not only for reasons of model legibility, but also because we use attributes and entities as "hooks" for actions and motivators, as we'll explain further in the chapter.

4.4.2 Beliefs

Let $\mathcal{A}$ be an agent. The belief set $B(w)$ of agent $\mathcal{A}$ for any given world state w, is defined as a collection of entities, denoting both the internal and external perceived environments of the agent: $B(w) = \{e_1(w), e_2(w)...e_n(w)\}$.

The agent's beliefs can be distorted by its perception. Let $\mathcal{A}_P : W \longrightarrow B$ be the perception function of agent $\mathcal{A}$. This distortion can be more than just the values of certain attributes, it can extend to the structure of entities themselves. Going back to our example $vessel(w)$ entity, a fisher agent might perceive it in its entirety so its belief would look identical to the world version of the entity, namely:

$vessel(w) = \{weight(w), length(w), capacity(w), efficiency(w), price(w)\}$.

However, an agent who is not a fisher, but is responsible for assigning a suitable berth in harbor, might perceive the entity as:

$vessel(w) = \{weight(w), length(w), capacity(w)\}$.

The $efficiency$ and $price$ attributes are missing from its beliefs about the $vessel$ entity because they play no role in this agent's reasoning processes. It only needs to know how big the vessel is in order to fit it in the harbor. It is the same way most of us think about, for instance, our cars. We know enough to be able to use them to suit our needs. A mechanic, however, will hold much broader beliefs about what a car is, as required by their job. This implies that agents can modify their perception in order to create beliefs that more closely reflect the parts of the world they are interested in, perhaps through some learning process, either self-motivated or socially-driven. While the architecture allows this, we do not explore this possibility in this work.

The agent also has a set $B_H = \{e_1(w), e_2(w)...e_n(w)\}$ of hypothetical beliefs, a copy of its beliefs on which it can test the outcomes of various actions to check whether, or to what extent, they would satisfy its motivations. The outcomes of an action on the agent's hypothetical beliefs can include a measure of uncertainty and need not match exactly what the actual outcomes of an action will be (see section 4.4.3).

4.4.3 Actions

We define the basic action as a structure consisting of:

- a function that transforms one world state into another world state

  $A_t : W \longrightarrow W$

- a precondition, which is a function determining whether the action can be performed by the agent given its current beliefs

  $A_{pc} : B \longrightarrow [true, false]$

- an effect, which is a function that operates on the hypothetical beliefs of an agent and estimates the resulting world state should the action be performed

  $A_e : B_H \longrightarrow B_H$

Let's consider the action of fishing as an example. In order to go fishing, an agent needs to own a vessel. In this case, the precondition function checks the beliefs of the agent for a vessel entity. If found, the precondition returns true, signaling the fishing action is feasible and the reasoner can consider it for next action to be performed. Otherwise, the action is not feasible and is discarded from consideration by the reasoner. The form of the vessel entity depends heavily on the kind of model being built. If the only vessels in the model are fishing vessels, then the precondition needs to look for an entity $vessel$ within the agent's beliefs. However, if there is more than one kind of vessel, we need additional attributes to distinguish between the ones which can be used for fishing and the ones which cannot. For instance, we can add an attribute $fishingGear$ to fishing vessels, which is absent from vessels used for other purposes. In this case, the precondition returns true for entities whose signature contains an attribute named $fishingGear$. The precondition doesn't need to know about $fishingGear(w)$ because agents can use any fishing gear to fish, although efficiencies and bycatch may vary. If we also want to consider that vessels will become unusable unless regularly maintained, we can add the attribute $isSeaworthy$. Now, the precondition looks for entities $vessel(w)$ which contain the attributes $fishingGear(w)$ and $isSeaworthy(w)$. If $isSeaworthy(w) = false$, the vessel cannot be used to fish and the precondition function must return false.

While the transformation function is the only one necessary for the agent to act upon the world, the preconditions and effects are used by the reasoning process to determine whether the action is possible and suitable. Note that the effect is not a retrospective assessment of the world state after the action has been performed, but a hypothetical estimation of the future world state should the action be performed. This means that the result of the function are expressed in the agent's hypothetical beliefs, and are subject to distortion due to the agent's biases, lack of information or misinformation (coming from beliefs that are not reflective of the real-world), or just the randomness inherent in certain world systems. For example, paying one's debts depends only on how much money the agent has and how much it needs to pay at any given simulation step. It can easily and precisely estimate the effects this action will have on its reserve of money and the size of the remaining debt. In this case, the effects function is the same as the transformation function, except it acts on the hypothetical beliefs of the agent and not on the world. In contrast, fishing involves some degree of randomness. The odds of making a good catch can be increased by using better gear, a better vessel and gaining more experience, but luck still plays a part. As such, when estimating the effects of a fishing action, the agent can factor in its gear, vessel and experience, plus its own history of catches, and get a fairly accurate result most of the time, but not every time. How much the result deviates from beliefs depends on the effect function and the dynamics of the actual resource the agent is exploiting. The influence of these discrepancies between expectation and reality on the agent's behavior need to be determined by the modeler and can range from no influence, to decreased motivation to perform the action, to increased motivation to learn or improve in ways that will close the gap between what the agent expects to happen and what actually happens. In the models presented in this work, these discrepancies have no influence on behavior because we are more focused on setting up the architecture than exploring possible uses.

4.4.4 Drivers of behavior

For this version of the architecture, we chose goals, social norms, policy and values as the drivers of agent behavior.

We include goals because they are one of the more basic drivers of behavior and can serve to motivate an agent even in the absence of other drivers such as norms or values. This is important when comparing simulation outcomes between populations assumed to be driven by norms and those assumed to be driven by personal interests or preferences.

We include norms because they are recognised as fundamental drivers of human behavior, and they share a similar function and structure with policy (indeed, policy is just a different type of norm). This serves not only in building agents with more realistic decision making processes, which leads to more realistic simulation results when studying policy impacts, but also serves to highlight and make explicit the kinds of conflicts that arise when a new policy is introduced to a population already governed by norms. This, in turn, helps explain the observed agent behaviors, which greatly increases the benefits of using ABMs as experimental sandboxes in which to explore possible policies and their outcomes.

We include values in order to be able to solve the issue of prioritizing between motivations. This is especially relevant when dealing with norm-policy conflicts because we are not interested in giving agents hard rules about which they should prioritize in a given context. This would completely invalidate the whole point of this agent architecture, which aims to let the agent act according to its own motives, not ad-hoc rules. Since we cannot impose a winner in any norm-policy conflicts, the agents must have access to a higher order motivation, which they can use to resolve the conflicts as they arise.

4.4.4.1 Goals

Goals are some of the most basic agent motivators and are necessary for driving the behavior agents in the absence of norms and values in the modelled population.

We take goals to be simply states of the world the agent wants to bring about. Some goal states are preferred over others, whether by definition, or through norm and value 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, which we call the condition set of the goal.

We define a goal as a structure consisting of:

- a function which determines whether the current world state fulfills the goal:

  $G : B_H :\longrightarrow \{true, false\}$

For example, let's consider the goal of going fishing. There are a number of ways this can be conceptualized, but we chose to use the $catch(w) = \{size(w), cost(w)\}$ entity to determine whether the goal is fulfilled or not. The goal function checks whether $size(w) > 0$ and the only actions which ensures this outcome are whichever fishing actions the agent has available. If there's more than one, the agent's reasoner will have additional ways of choosing between them. It is also possible to build an action tracker entity such as $tracker(w) = \{hasFished(w), hasPayedDebts(w), hasSoldCatch(w)\}$ where $\forall c \in tracker, c : W \longrightarrow \{true, false\}$. This entity would need to be updated by the transformation functions of the actions it tracks and reset by the agent's perception as appropriate. In this case the goal function would check if $hasFished(w) = true$. Such an approach would be useful if the model allows the possibility for a fishing trip to result in zero catch, but the simulation step can't allow for multiple fishing trips. We used fishing functions which always return positive catch (with varying magnitudes) because our time step is one month, which would account for multiple fishing trips in real life and we assume at least some of them are successful to some degree. If the time step were a day and we allowed for failed trips, then we would have used a different goal function and different entities to represent the agent's beliefs.

4.4.4.2 Norms and policy

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 actions and goals which are context specific, becoming active in certain contexts and remaining inactive otherwise.

Policies are similar to norms, but rather than being imposed socially, they are imposed by a ruling authority.

In short, social norms and policy are two versions of the same concept: a rule imposed from the outside, applicable within a specific context, which, once adopted, becomes preferred over all other courses of action. In the kinds of models we are concerned with, social norms are assumed to be already adopted, while policy is yet to be so and carries a lower preference compared to a norm if they are both active at the same time. This similarity is reinforced in our architecture by using the same structure for both norms and policy:

- an activation function, which checks whether the world state falls into the activation context for the norm

  $N_C : B_H \rightarrow \{true, false\}$

- if the agent has a value system, a function which estimates the value fulfilment provided by compliance with the norm/policy, relative to the value system of the agent

  $N_V : V \rightarrow [0, 1]$, where V is the value system of the agent

- an action or a goal
- a flag indicating whether the structure is a norm or a policy

Note that the activation context doesn't need to be explicitly specified, it is enough to have a function that returns the activation status of the norm/policy.

If the norm/policy is active, the action or goal become the preferred choice for the agent, but this doesn't mean they are immediately acted upon. They must undergo the same reasoning process as any non-normative action or goal in order to determine feasibility or fulfillment status, respectively, as well as resolve any conflicts that arise in cases where more than one norm or policy is active at a time.

4.4.4.3 Values

Norms are powerful motivators of behavior, but any set of norms, just like any set of goals, typically lacks an internal structure that can be used to derive priorities among them. Even predefined priorities between classes of motivators – such as declaring that norms take precedence over social practices – are only sufficient in some models. In ours, this outside imposition of priorities over drivers of behavior cannot help an agent solve conflicts between norms and policies in a flexible manner. We are forced to declare a winner beforehand, either in general – norms win every time or policies win every time – or on a case by case basis. Both of these defeat the purpose of the models, which is to study the choices of the agent when faced with these conflicts, which is impossible if we've already imposed the outcome.

As such, we must have a higher order motivator which would allow the agent to decide how to solve conflicts between motivators of all classes as they arise. For the reasons described in 3, we choose Schwartz values to act as these higher order motivators in our architecture.

Schwartz values are well documented and are seen as a reflections of human needs [128], which all people share, although different people show different preference orders over their value set. Schwartz defines ten abstract values and observes that some of these values support one another while others are in conflict. These relationships between values are represented in the Schwartz value wheel 4.5 and allow us to understand how values relate to one another. The farther two values are from each other on the wheel, the more "adversarial" they are said to be and seeking to fulfill one of them automatically means depleting the other.

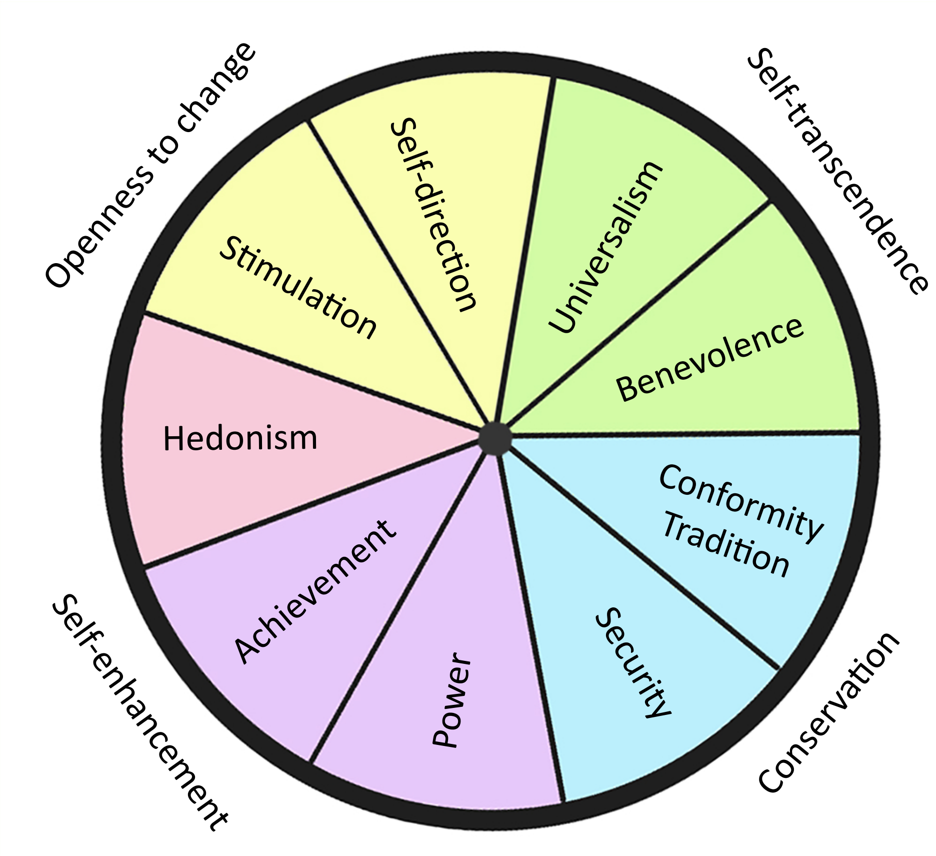


Fig. 4.5: Schwartz value wheel.

4.4.5 Value trees

These fundamental values are highly abstract and as such are not easy to use by the agent. In order to render them usable in a decision making process, they are made progressively more concrete until they map out onto real world states. For instance, care for the environment is an abstract, non-operationalizable value, but it can be made more specific by breaking it down into sustainable catch size, environmentally-friendly fishing gear, fuel-efficient vessels etc. Not every agents care for the environment needs to be specified the same way, and not all specification apply to all agents, which can be due to context – fuel efficiency is not a consideration if you're not using a powered vessel – or alternative conceptualizations – sustainability of the catch might be measured by time of the year rather than by size. These more specific values can in turn be specified further – type of fuel, electrical engines, partially wind-powered. As the specification process continues, we eventually end up with real, tangible, measurable elements of the world. The agent's preferences over abstract values – such as more environmental care is better than less environmental care – can now be mapped onto preferences over real-world attributes – such as fishing with long-lines is better than fishing with nets. Once these very specific preference functions are in place, agents can decide what behaviors to enact such that they bring about more preferred world states.

Even though we refer to the fundamental values as roots of trees, which implies each value has its own separate tree structure attached, this is not strictly true since we allow trees belonging to different values to share nodes as needed. This is a more parsimonious structure than duplicating nodes in order to keep trees separate. We continue to refer to this structure as a tree because, for each fundamental value, a tree that follows the successive concretization of the root value into world attributes can be extracted and separated from the larger structure. The actual structure used in this framework more closely resembles the bastard offspring of trees and directed graphs which is to say its a directed graph with:

- a set R_V of nodes called roots. No directed path through the structure can continue beyond these nodes. This set cannot be empty.
- a set L_V of nodes called leaves, which are functionally related to world states. No directed path through the structure can start with a node leading into a leaf node. This set cannot be empty.
- a set I_V of intermediary nodes that sit between leaves and roots on any directed paths which start in the leaves and end in the roots. This set can be empty.

In order to represent this abstract value to concrete world mapping operationalization, we use a tree structure, with a fundamental value in the root getting more and more specific with each level further from the root, until it maps out onto real-world states in the leaves.

We can define the structure of a basic node in the value tree as a preference function, with the caveat that the domain differs between types of nodes:

- a set L_V of leaf nodes, where a leaf node l_V is defined as a preference function over a subset of the world state

  $l_V : W \rightarrow [0,1]$

- a set I_V of mid-nodes, where each node $i_V \in I_V$ is defined as a preference function over their child nodes' preferences:

  $i_V : \{Children(i_V)\} \rightarrow [0,1]$,

  $$i_V(Children(i_V)) = \begin{cases} i_V(Children(i_V)) & \text{if } Children(i_V) \subseteq L_V \\ i_V(Children(Children(i_V)) & \text{if } Children(i_V) \subseteq I_V \end{cases}$$

- a set R_V of root nodes, where each node $r_V \in R_V$ is defined as a preference function over their child nodes' preferences:

  $r_V : \{Children(r_V)\} \rightarrow [0,1]$,

  $$r_V(Children(r_V)) = \begin{cases} r_V(Children(r_V)) & \text{if } Children(r_V) \subseteq L_V \\ r_V(Children(Children(r_V)) & \text{if } Children(r_V) \subseteq I_V \end{cases}$$

  Additionally, root nodes also contain a decay function, which lowers their preference value over time in the absence of any changes in the preferences of their children, in accordance with the "water tank" model of values from [68].

  $decay : [0,1] \rightarrow [0,1]$ with $r_V(x) < decay(r_V(x))$

As such, nodes in the value tree all contain preference functions that determine whether a (partial) world-state is preferred relative to the root fundamental values.

In the leaves, the preference function determines whether a subset of the entities making up the current world state is preferred or not, such as determining whether long lines are preferred over nets. It is entirely possibly for multiple leaf nodes to share the same entity subset and yet return different preference values. This is because leaf preference reflects root value preference. For example, long lines may be preferred over nets when it comes to universalism because they are more eco-friendly, but not preferred when it comes to achievement because they yield smaller catches.

For mid-nodes and root nodes, the preference function measures preferences over states that get more abstract the higher up the tree we climb. Because it's rather difficult to define abstract concepts in a way that would make it simple to assign them domains and distance measures, mid-nodes do not attempt any of this, instead aggregating their descendants preferences. This aggregation reflects the priority of the root value in the agent's value system.

For instance, lets say we decompose universalism into eco-friendliness and further into fuel efficient engines and long lines. If the current world state is **fuel efficient vessel with nets**, I have one preferred and one not-preferred world element. What the agent's eco-friendliness preference is depends on how it aggregates preferences over the two concrete elements. If the agent's universalism (or at least its eco-friendliness) doesn't have very high priority, the agent may be content with any of the two, and the current state of the world fills the eco-friendliness meter to a high degree. If, however, universalism is a high priority value, the agent may want both long-lines and fuel-efficiency in order to be satisfied with the current world state (see figs. 4.6, 4.7, 4.8). If the agents considers long lines to be more important than fuel-efficient engines, the aggregate preference is weighted in favor of the long lines leaf node preference.

Depending on the depth of the tree and the specificity of the leaf nodes, the agent's preferences over its world can be described in as much or as little detail as required by the model.

Finally, the agent's value system contains an overall fulfillment function, which aggregates the preference of all the root nodes of the value tree.

$f : R_V \rightarrow [0, 1]$

4.4.6 Reasoners

In our architecture, motivators and behaviors are more than inert blocks of information to be used by a central reasoning process. Rather, any instance of a motivator or behavior also includes the part of the reasoning process that is specific to it. Actions include the reasoning mechanisms required to determine whether they are feasible and what their effect will be. Norms include the determination of whether they are currently active and what their value effect is. Goals contain the means to determine whether they are fulfilled. This separates the particular reasoning required for each individual motivator or behavior from the general reasoning *about* motives and behavior, and allows us to build agents capable of reasoning generally about classes of motivators and behaviors rather than any combination of specific actions, goals, norms or policies. In formal terms, a reasoner is a function which operates over the agent's beliefs, motivator set and action set, and outputs the next action to be performed.

$R : B_{\mathcal{A}} \times G_{\mathcal{A}} \times N_{\mathcal{A}} \times V_{\mathcal{A}} \times A_{\mathcal{A}} \rightarrow A_{\mathcal{A}}$, where $B_{\mathcal{A}}$, $G_{\mathcal{A}}$, $N_{\mathcal{A}}$, $V_{\mathcal{A}}$ and $A_{\mathcal{A}}$ are agent $\mathcal{A}$'s belief set, goal set, norm/policy set, value tree and action set, respectively.

This separation of instance specific reasoning from general reasoning leads to an architecture that is strongly modular. The central reasoning module is not concerned with the particulars of any of the agent's motivators or behaviors, which means it can operate on any number of specific instances of motivators or behaviors, as long as they are part of a class the reasoner understands. This is a significant advantage for agent models being used with a variety of different scenarios because the main reasoning loop only needs to be developed once, and the other elements can be mixed and matched as the scenarios demand.

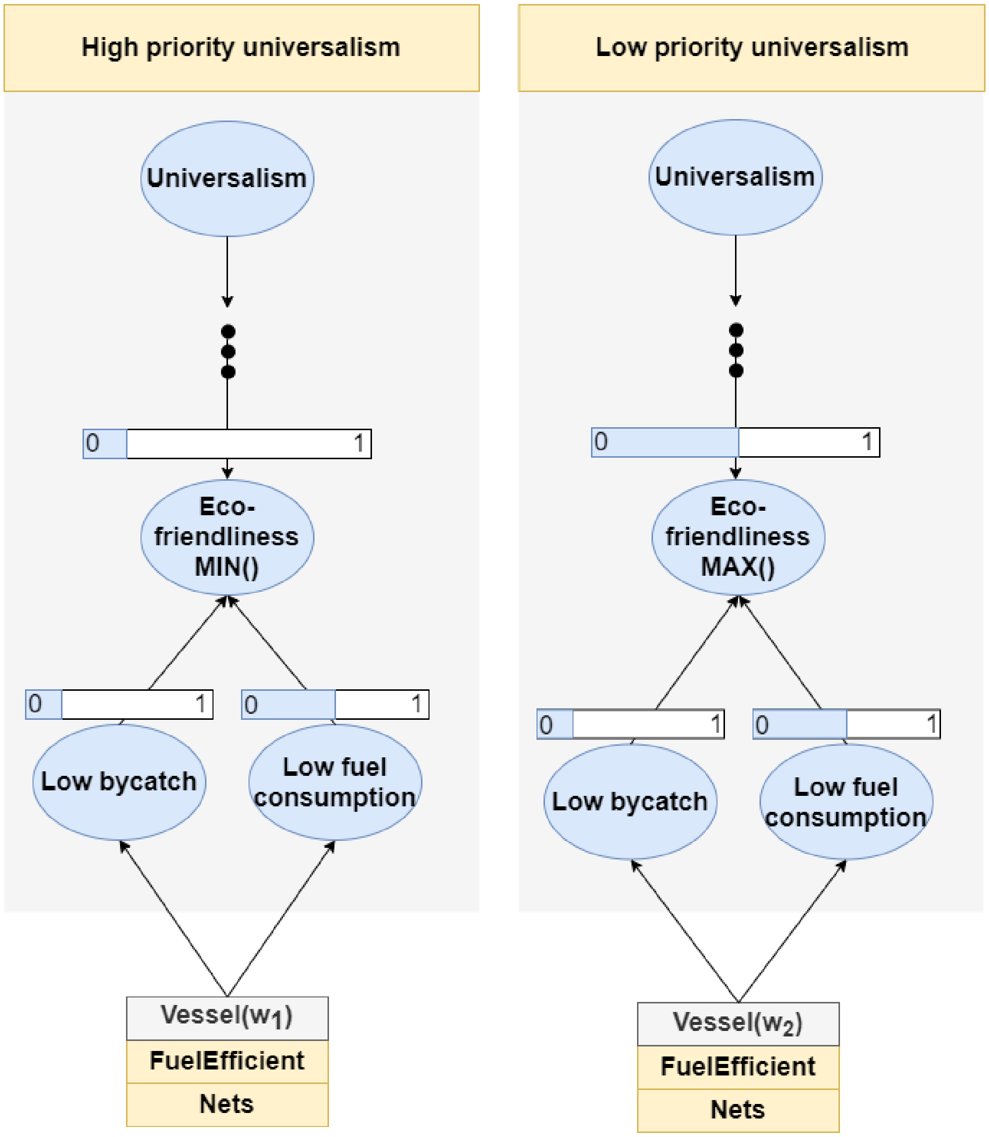


Fig. 4.6: Example of controlling the priority of a value through preference functions in tree nodes. Left, the Eco-friendliness node prefers that $Vessel(w_1)$ be both fuel efficient and use low bycatch fishing gear. Right, the same node prefers that at least one of the $Vessel(w_1)$ attributes be either fuel efficient or low bycatch.

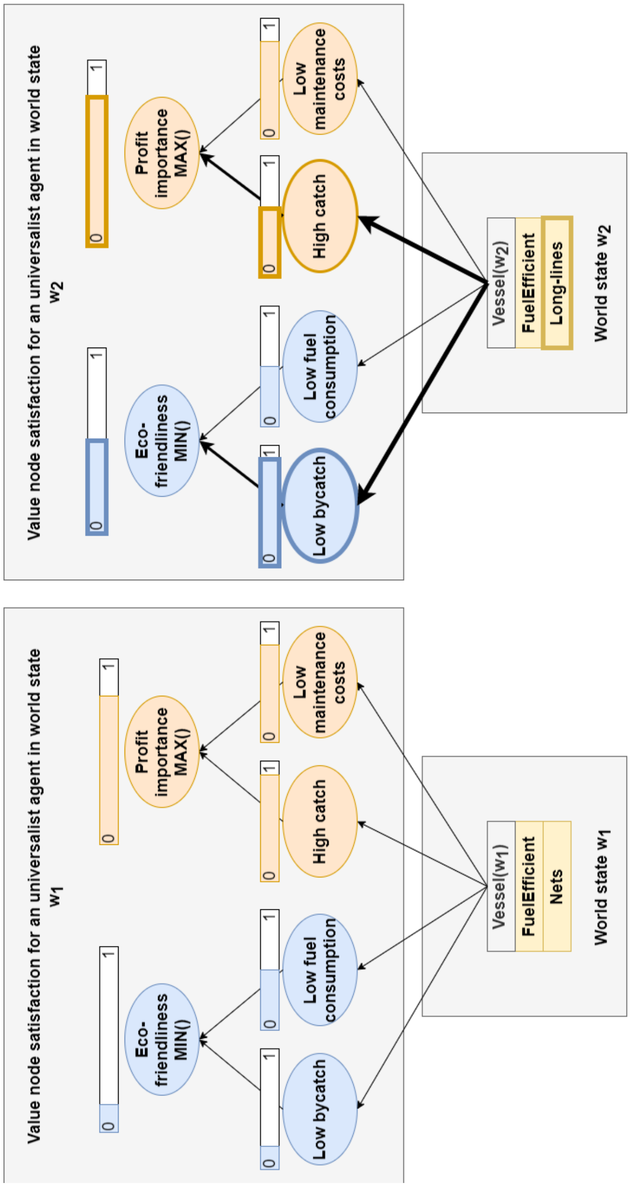


Fig. 4.7: Example of controlling the priority of a value through preference functions in tree nodes. The Eco-friendliness node prefers that $Vessel(w_1)$ be both fuel efficient and use low bycatch fishing gear, while the Profit importance node prefers at least one. Switching to long-lines improves the preference of eco-friendliness, while Profi importance remains the same, even though the High catch node preference is now lower.

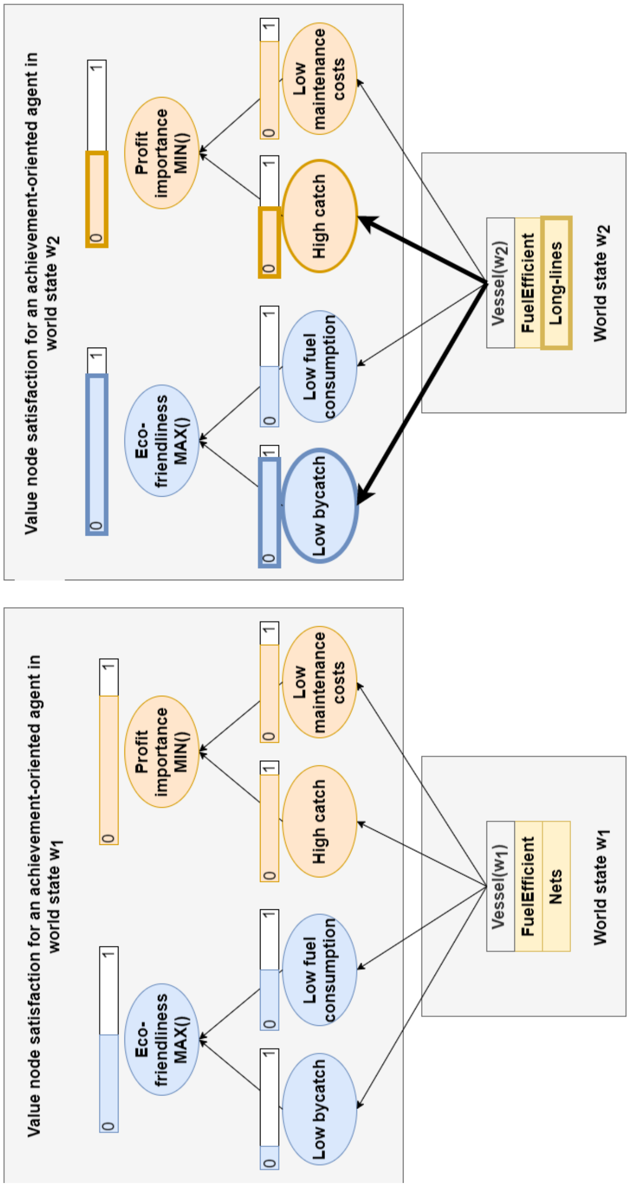


Fig. 4.8: Example of controlling the priority of a value through preference functions in tree nodes. The Eco-friendliness node prefers that $Vessel(w_1)$ be either fuel efficient and use low bycatch fishing gear, while the Profit importance node prefers it be both fuel efficient and use high catch gear. Switching to long-lines improves the preference of eco-friendliness, while Profit importance decreases because the High catch node preference is now lower.

To illustrate, see fig. 4.1 for an example of a simple reasoner that knows how to work with norms and policy. In this case, the reasoner assumes that agents will always choose to fulfill normative or policy goals first, and also favor any policy over their norms if they should ever be active at the same time. Note, from the pseudo-code, that the reasoner doesn't need to know anything about what the goals, norms, policies or actions are or do. It only needs to be able to call specific functions on each of the motivator types in order to determine activation, fulfillment and feasibility. For a full example, including the entities, actions and motivators used, see chapter 6.

For a reasoner that doesn't automatically favor policy over norms (or vice versa), see fig. 4.2. This reasoner uses values, and therefore can actually solve policy-norm conflicts. It, again, doesn't need to know anything about the motivators it works with, beyond their type, it just needs to be able to call the functions it needs to determine feasibility, fulfillment and activation. For a full example, including the entities, actions and motivators used, see chapter 6.

Because nothing is ever straightforward in ABM, while this kind of modularity can lend considerable flexibility to model design, the same modularity can reduce the reasoning power of the agent because the motivators and behaviors are not allowed to interact with one other, unless they do it through the reasoner.

In cases where these elements relate to one another in complex webs of influence, modularity is not as cleanly achieved, both for conceptual and computational reasons. First, conceptually, if the actions and motivators are interrelated, these relations will likely not be maintained if we substitute an action or a motivator for another. In such cases, the cognitive elements of the architecture are no longer as easily interchangeable. We can shift all the calculations taking place on this web of interactions into the central reasoning loop, resulting in complex and computationally demanding reasoners. The alternative would be to force the complex interactions between motivators/behaviors into a simplified version that can be neatly divided into instance specific reasoning processes that get attached to the motivators/behaviors, and general reasoning rules that get bundled up into a central reasoner. In the end, it is up to the modeler to decide whether they'd benefit more from: a strongly modular or a strongly interconnected approach.

4.5 Limitations

While we integrated a number of different drivers of behavior into our architecture so that we can allow agents to reason their way through conflicting motivations brought about by the introduction of new policies, as well as by their own internal makeup, we did not – and could not – account for every driver that could prove relevant in an arbitrary decision context. Indeed, we did not – and could not – even account for every relevant interpretation of the drivers we included. As such, this section briefly discusses the more salient limitations of this architecture.

```
if goalQueue is empty
    add all goals to goalQueue

forall (goals in goalQueue)
    if (goal is not fulfilled)
        if (goal is normGoal)
            if (goal is active)
                add goal to normGoalList

        else if (goal is policyGoal)
            if (goal is active)
                add goal to policyGoalList

        else add goal to goalList

if (normGoalList is not empty)
    nextGoal is random goal from normGoalList
else if (policyGoalList is not empty)
    nextGoal is random goal from policyGoalList
else
    nextGoal is random goal from goalList

forall (actions in availableActionsList)
    if action fulfills nextGoal
        if (action is normAction)
            if (action is active)
                add action to normActionList
        else if (action is policyAction)
                if (action is active)
                    add action to policyActionList
        else add action to actionList

if (normActionList is not empty)
    nextAction is random action from normActionList
else if (policyActionList is not empty)
    nextAction is random action from policyActionList
else
    nextAction is random action from actionList

execute nextAction
\label{code_basereasoner}
```

Code 4.1: Reasoning loop for base reasoner

First and foremost, our interpretation of norms and policies does not include provisions for punishments incurred for breaking said norms and policies. Punishments for straying from prescribed behavior, either socially (in the case of norms) or in relation to an authority (in the case of policy) are an integral part of the dynamics of these drivers. Any policy that is not enforced and lacks punishments for violations will have a hard time gaining widespread compliance

```
update value tree satisfaction
forall (actions in availableActions)
    if (action is normAction)
        if (action is active)
            add action to normActionList
    else if (action is policyAction)
        if (action is active)
            add action to actionList
    else add action to actionList

if (normActionList is not empty)
    nextAct = max(valueSatisfaction(action in normActionList))
    execute nextAct
    break

else
    nextAct = max(valueSatisfaction(action in actionList))
    execute nextAct
```

Code 4.2: Reasoning loop for value reasoner

through a population, especially one which is already governed by some existing prescribed behavior such as norms. Indeed, even when enforced, policies may meet with significant resistance. Norms are also enforced, albeit more subtly, through social mechanisms such as shaming, gossip, shunning or personal confrontations, and unenforced norms are likely to lose their power and disappear or get replaced over time.

However, we argue that, even in the absence of explicit enforcement mechanisms, simulations using this agent architecture can provide valuable insights into the effects of a policy. First, by forcing an explicit representation of all motivators and behaviors, we force the modelers, domain experts and stakeholders involved in the development of the models to uncover any implicit assumptions they have about how the people being modeled know, think and behave. Second, the agent's degree of compliance with the new policy in the absence of any overt enforcement can give clues about how strongly the new policy conflicts with their inner motivations, and thus how much of an overt enforcement effort will be required to compel large scale compliance. Furthermore, in some cases, enforcement is costly or nearly impossible to carry out – for instance, on the deck of a fishing vessel out at sea. Here, a lack of compliance in the simulation may suggest that the policy, in its current form, will turn out to be ineffective and encourage further experimentation with other policy options.

Our representation of norms is also limited in that we do not explicitly differentiate between permissions, obligations and prohibitions. Our norms are basically internal obligations, meaning the agent has to follow the norm once it becomes active, or find a way to resolve the conflict if more than one norm is active at a time. However, since our architecture doesn't account for punishments, we can argue that there is no meaningful difference between permission and obligation, since conflicts between permissions or between obligations both are solved in the exact same manner by using the agent's value system. This leaves prohibitions, and we do not offer a direct way to account for them.

We also don't include the life cycle of norms in this architecture. Our agents have no way of internalizing new norms or discarding old ones, and this is a relevant part of norm dynamics if we consider policy effects over a longer term. If a policy can become a norm, the effort required to enforce it would significantly go down and it may serve as a basis on which to introduce related policies in the future. For instance, the prohibition against stealing is a law, but it is also a norm. For most of us, its the norm that keeps us from stealing something, not the threat of being arrested. The social disapproval and shunning weighs more heavily than the legal repercussions when we consider the consequences of being caught. Digital piracy laws piggybacked on this anti-theft norm when they came out, equating pirating a movie to "stealing a car" for instance. If a policy could become a similar norm, it would be nearly self-reinforcing and being able to determine a possible path from policy to norm would be well worth the study.

However, experimenting with the life cycle of norms requires us to include enforcement, which we don't, as well as social dynamics such a trust, group dynamics or peer pressure, which are also absent. Their absence also means we could not include social roles, despite their importance to the study of policy effects. Leaders and other trusted or influential figures are prominent in determining how much acceptance and compliance a policy achieves. Well-liked and trusted individuals can speed up compliance when they themselves are seen to comply. Conversely, highly distrusted figures can substantially hamper compliance with a policy they are seen to support. Different stakeholder groups look to different high-profile individuals with varying degrees of trust or mistrust and this can result in highly informative dynamics when captured in a model.

Fortunately, the architecture can be fairly easily extended to account for at least some of these limitations, which are discussed in 7.

Part III

Implementation, models and results

Chapter 5

Implementation

Many agent models based on some interpretation of norms and/or values and their associated reasoning put forth well thought out conceptual architectures for their agents and their cognitive processes. But, when it comes to the implementation of models using said architectures, it's suddenly very difficult to find the same concepts in the code. The theoretical architecture gets crunched down into a form that serves the desired function, but the resulting code is often only usable for the one model the developers are interested in at that moment. One one hand, this makes perfect sense if there is no need, or intention, of building further models, either as variations of the first model or further applications of the theoretical architecture. Not to mention the particularities of the programming language being used might not lend themselves easily to building the elements required by the conceptual architecture. On the other hand, this approach virtually guarantees that the implementation has no reusability, nor in-built avenues for expansion. This means any future model would have to be coded from scratch, even when there's overlap with previous models, and any expansion of the model will hit a level of unmanageable complexity sooner than it needs to. Plus, without an underlying implemented architecture, it is very difficult to compare models which claim to be based on the same theoretical architecture.

The implementation described in this chapter sets out to provide an implemented architecture that resembles the conceptual architecture as closely as possible. All the elements described in the theoretical architecture are replicated explicitly in the implemented version, many in generic form, and all of them reusable in at least some capacity. This approach requires more effort in the beginning, both coding and design -wise, but it ensures that we can develop not just the occasional one-off model, but build a sandbox in which the user can easily run modelling experiments, both by quickly building variation on the same base model or by quickly prototyping different models. The implementation achieves this level of agility by preserving the modularity and flexibility of agents as described in the conceptual architecture. Components, entities, actions, norms, policies, goals and values remain explicit "building block" elements in their implemented versions, and the reasoning process of the agents operates on and with these building blocks.

In this chapter we present the generic "template" for agents and their building blocks, with model specific versions of these elements serving as illustrative examples. The models are discussed in detail in chapter 6. We present the implemented explicit representations of components, entities, actions, norms, goals, and values, and establish a clear separation between the reasoning process and its building blocks. We also show how this explicit modular approach can lessen the need for hard coding the agents, which allows for much faster iteration of simulation scenarios and therefore makes it much easier to run many different simulation experiments. We also took advantage of the modularity of the implementation to develop a visual interface for quickly prototyping agent types by mixing and matching cognitive building blocks with zero or minimal modifications required of the source code.

Before continuing, we must mention that the implementation presented here is by no means the only one possible. Programming is a vast field that offers many tools and many ways to use those tools to achieve desired goals. We chose this implementation of the architecture because we wanted it to mirror its conceptual counterpart as much as possible. In some ways, this code can be considered more of a didactic tool, meant to let us test the capabilities and limitations of our architecture, rather than a finished and optimized piece of software for agent based modeling and social simulation.

5.1 Hard vs. soft coding reprised

As mentioned in chapter 4.2.1, hard coding refers to the practice of embedding data directly into the source code of a program and requires changes to the code when the aforementioned data changes. Soft coding, on the other hand, is the practice of embedding arbitrary data in the code, which is then made specific at run time through user input, loading from files, databes etc. Agent modelling tools like Netlogo or Repast already allow the soft coding of certain types of data, mainly parameter values, which are represented as variables in the code that get their values at runtime from the user interface or customization files. This is often sufficient in the case of agents with a fixed set of behavior rules. If the rule set doesn't change from one scenario to another, hard coding the rule set is a perfectly serviceable practice.

However, we *are* dealing with agents which need to be able to change their rule set and, therefore, coding and keeping track of numerous rule sets quickly becomes unmanageable. For instance, going back to the example in chapter 4, in the fisheries case study, models would need to account for different types of fishers involved in a fishery and different policies that could be experimented with in a simulation. To complicate matters, some scenarios would change the policy mid-simulation, either from a "no policy" condition or from a different starting policy, and introducing new policy will often introduce whole new concepts. For instance, quota fishing policies differ both geographically and over time, but they share core traits: a TAC that is divided between fishers and restrictions on what one is allowed to do with their share of the TAC. How the TAC is divided, whether ones share is tied to a person, company or vessel, whether one can sell, buy or rent shares and to whom, these are important considerations when designing a quota policy. However, any quota policy is fundamentally different from a fishing license policy, both in their rules and in their conceptualization of fishing.

To go into some details, let's consider a simulation experiment which includes profit-motivated agents and two types of normative agents operating under an effort quota policy. This policy can lead to "Olympic fishing", where fishers increase their fishing capacity and efficiency as much as possible to get the most catch during the limited number of fishing days. Since this situation is often characterized by overfishing, two alternative policies are proposed for experimentation: a transferable catch-quota policy and a territorial use rights policy. These new policies are fundamentally different from each other and from the old policy. It's not just a matter of changing the number or distribution of fishing days, or limiting the size and gear of the ships. They introduce new concepts such as **catch-quota** and **territory**, new actions such as **buy** and **sell quota** or **monitor other fishers**, and new rules such as ***sell quota*** *if* ***end of year*** *and* ***quota remaining is more than zero*** or ***administer punishment*** *if* ***observed infraction***. All of these will need to be coded into all agent types, even though, only two policies will be active during the simulation. And, at any time, a request may be made for yet another policy to be introduced for experimentation, for instance a rights auction policy, which in turn introduces its own concepts such as **auction**, actions such as **bid** and rules such as *bid if* ***current cost is lower than available funds*** or *bid if* ***expected value of catch is larger than cost***.

What this example shows is that representing entities as variables is not going to offer any easy gains for behavioral flexibility. We'd either need to keep adding variables to the agent code as new concepts are introduced, or develop an inheritance-based hierarchy between types of agents. We'd have to decide between managing the increasing complexity of a few agent types (or one giant agent type if we want to develop models in which agents can change their decision makeup either over time, or from context to context), or trying to fit new agent types into one hierarchy, which, depending on the pace of development or the scope of the models, can easily spiral out off manageable bounds.

While the overly-complex agent is a fairly self-explanatory concept, and one many developers have struggled with at one time or another, the shortcomings of an agent hierarchy in the context of fast-paced model development might not be as intuitive so we'll use an example to illustrate the main issue we're trying to prevent. Say we've developed the **Fisher** agent type, who keeps track of the most basic elements a fisher agent needs in order to function in a fishing model, such as **VesselEfficiency**. We then want to build a model in which we compare two different policies. In one scenario start in an open access fishery and transition to a quota-based regime. In another, we start in open access and transition to a fishery in which highgrading is made illegal. For our first scenario, we create the **QuotaFisher** agent type, which inherits everything from **Fisher**, with an added **RemainingQuota** element. For our second scenario, we create the **HighgradeFisher** agent type, which also inherits from **Fisher**, with an added **MinFishSize** element. So far, so good. However, if we would like to add a third scenario, where both quota and no-highgrading come into effect together after the fishery has been operating in open access for a while, we need to create a new type **TwoPoliciesFisher**, which cannot inherit from both **QuotaFisher** and **HighgradeFisher**, at least not if we want to keep the hierarchy clean. This is called, somewhat dramatically, the Deadly Diamond of Death (see fig 5.1), and, even if we wanted to take the risk of a tangled hierarchy with its possibly deadly diamonds, many programming languages do not allow this kind of multiple inheritance precisely because of issues of complexity and ambiguity. A safer way would be to have our new type inherit either from **HighgradeFisher** or **QuotaFisher**, but, regardless of our choice, we'd have to rewrite part of the code (which already exists in the type we did not inherit from, but can't be used because we can't inherit it)

None of these scenarios are particularly desirable, not the overly-complex agent type, not the tangled hierarchy, and not the clean hierarchy that requires significant code duplication. Yet, this is not where the difficulties end because trying to soft-code rules can be even more challenging. We want a way to separate behavioral and reasoning rules from the main flow of the program, make them interchangeable and switch between them as needed at runtime. Going back to the previous example, the rules for fishing under open-access, no-highgrade and quota regimes are obviously different. If we code the actions **Fish**, **FishQuota** and **FishHighgrade**, and want them outside the hard-coded control flow of the program, their code cannot be part of the control flow of any agent type, but must be accessed some other way (see fig. 5.2).

It must be mentioned that while a soft-coding approach can offer substantial flexibility, there are downsides. Unless properly planned and directed, soft-coding can also fall into unmanageable complexity if the configuration process becomes too complex. We address this problem and our proposed solution towards the end of the chapter, in section Saving data.

It bears mentioning that soft-coding isn't unheard of when it comes to agent models or multi-agent systems. Indeed, this splitting of an element into its "description" and its "concrete" counterpart which contains the specific data or functionality can also be seen in the development of agent languages like 2APL [35], GOAL [74], Jason[21], or Jadex [106]. A good overview of these, and a discussion on environment interface standards (EIS) for agent models can be found in [14]. Our approach is different in a few key ways, as we will show in the rest of this chapter. First, we provide explicit representations for more elements than just percepts, actions and goals, and we even allow agents to have arbitrary reasoners as coded by the modeler. Second, we do not use reflection to dynamically bind types or invoke methods at runtime, relying instead on function pointers. And finally, many "concrete" elements do not inherit from their "description" since they are not in a parent-child class relationship.

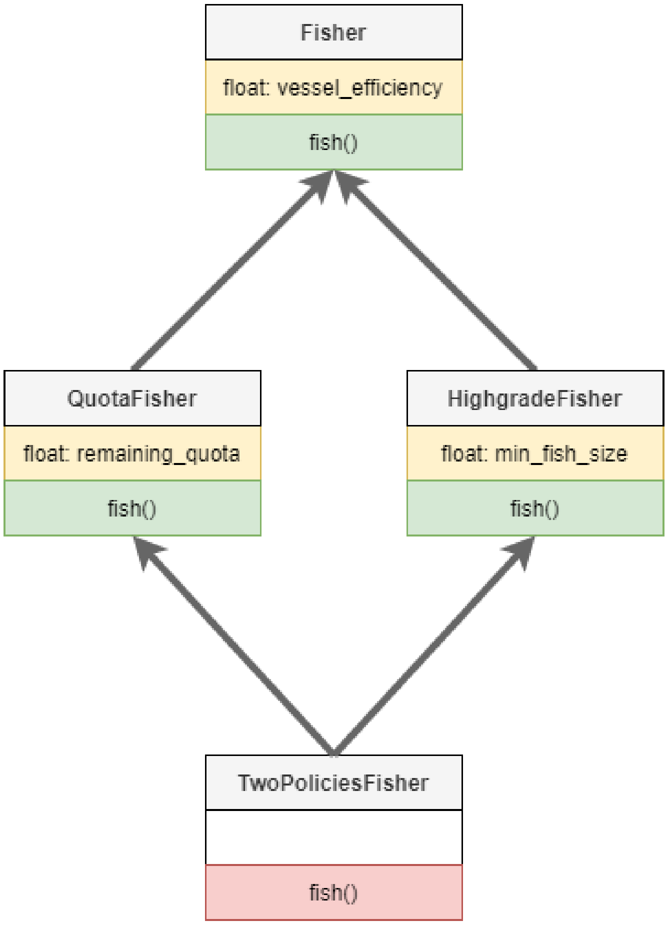


Fig. 5.1: The Deadly Diamond of Death: **TwoPoliciesFisher** agents cannot decide which **fish()** function to inherit. In order for such an agent to exist it needs to inherit one function from one parent and reimplement the other itself.

5.2 Architecture

As mentioned before, the implemented architecture mirrors closely its conceptual counterpart. We base the implementation on the principle of composition, rather than the inheritance common to object oriented applications. Our approach borrows from the entity-component-system (ECS) architectural pattern, which separates data from behavior. Data is represented as entities, which are nothing more than a collection of components, which in turn are nothing more

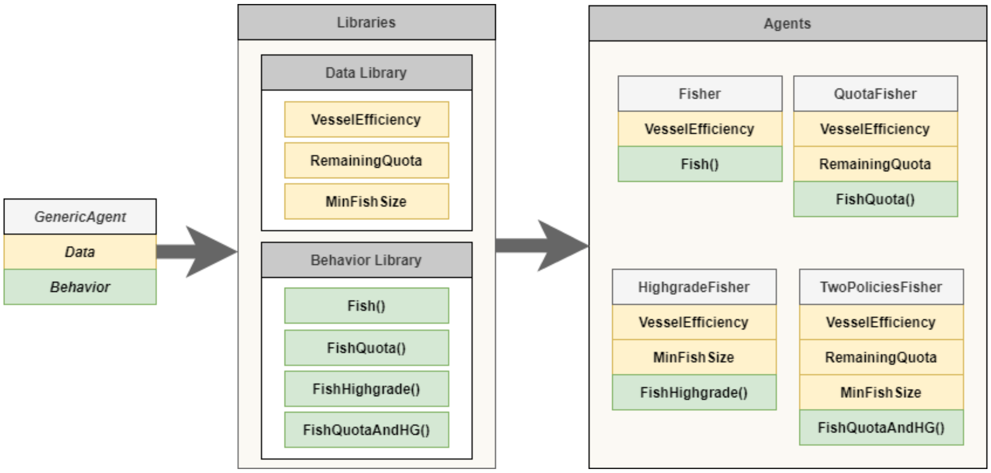


Fig. 5.2: Using data and behavior composition to avoid the Deadly Diamond of Death.

than data holding structures. Behavior is represented as systems, which operate over certain types of entities or entities containing certain kinds of components.

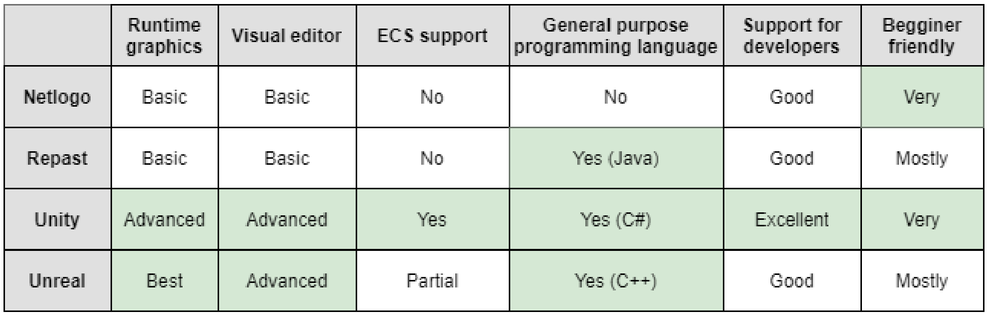
Our implementation does not go for the level of data and behavior separation present in ECS approaches. We use the entity-component representation of data, but no systems. According to our conceptual architecture, actions, norms, goals, policies and values would all need to be systems because they all operate on/with entities. However, systems are executed in a fixed order and we cannot know in advance which of them need to be executed during each time step since this decision is made by the reasoners. This could be solved with a bit of tinkering, but there is a more important reason we decided not to push systems here. An implementation that uses systems to operate on entities would break away from the conceptual architecture. This would defeat the purpose of having the implementation in its current form since, as mentioned in the opening of this chapter, our goal here is to maintain as close a correspondence as possible between the conceptual and implemented architecture, especially when it comes to the explicit representation of the architectural elements.

Thus, we borrow just enough from ECS to generate agents that do not have a fixed composition outside a minimum base template, acting as "containers" which can be loaded with any combination of actions, norms, goals and values which, together with a reasoner, determine the agent’s behavior.

In the following sections we will describe the types of elements an agent can contain, the visual interface that can be used to create an agent/simulation/batch configuration and show how a new agent configuration can be created.

5.2.1 Unity and C#

Given our soft-coding goals, we figured out early that we'd need a general-purpose programming language which would allow us much more freedom in the kind of types and structures we can create. At the same time, we would benefit greatly from support in creating, saving and organizing configuration files, preferably using a visual interface. And since we're interested to use this implementation in a participatory setting in the future, we would like the toolkit/environment/application we'll be using to implement our architecture, models and simulations to have the option of developing advanced runtime graphics. For a summary of our deciding factors and the applications we considered, see fig 5.3



	Runtime graphics	Visual editor	ECS support	General purpose programming language	Support for developers	Begginer friendly
Netlogo	Basic	Basic	No	No	Good	Very
Repast	Basic	Basic	No	Yes (Java)	Good	Mostly
Unity	Advanced	Advanced	Yes	Yes (C#)	Excellent	Very
Unreal	Best	Advanced	Partial	Yes (C++)	Good	Mostly

Fig. 5.3: Summary of agent-supporting software we considered for implementing the architecture.

We considered Netlogo because it is as close to an "industry standard" for agent models as the field has, and for good reason. It makes it easy and fast to set up agent models, and it has a decently versatile user and simulation interface. However, the programming language cannot easily handle the more advanced soft-coding elements we need for our implementation. Repast is a good alternative, because it does use a general purpose programming language, but the built-in model setup interface would take considerable time and effort to expand to the levels we require. The simulation interface would pose the same issues, possibly to an even higher degree, depending on how much interactivity we'd like to provide when engaging with stakeholders. Because of our unusually high (for the field) need for graphical interfaces and good, interactive graphics in general, our best choice came down to one of the game engines available for public use. Game engines in general provide at least basic support for agent simulations because many of the elements of any a game are, in fact, agents. They are also extremely proficient at providing developers with the ability to design state-of-the-art graphics and user interactions.

The list of game engines to choose from is fairly long, but it is generally agreed that Unity and Unreal are among the most popular options, with Unreal offering the better graphics, and

Unity being the more general-purpose user-friendly engine. Although it's rather unorthodox to go with a game engine, we settled on Unity as our development platform. As a game engine, Unity offers much more potential in terms of visualisation types and quality, as well as very fine and precise control over the simulation. Its editor allows the user to visually configure and organize many of the elements that go into creating agents, including its own type of configuration data files. On the programming side, C# is a far more powerful and versatile programming language compared to Netlogo, which makes the implementation of custom data structures and logic flows much faster and easier. Taken together, these features allow a developer the ability to implement much more powerful code, and also build interactive visual interfaces, both for building models and for running simulations.

While, this is true of Unreal and its programming language (C++), Unity won out for two main reasons. First, Unity's popularity is owed to its relatively leisurely learning curve (compared to other game engines) and massive community of developers/users which provide quick and reliable support, learning materials, tutorials and even fully developed assets. This makes it a better choice for someone with good programming experience, but low game engine specific experience, which is a profile many in the agent modeling community meet. Second, and more specific to this research, Unity has its own ECS solution, called DOTS[1]. Since it was under development while this research was carried out, it couldn't be used to build stable implementations, and so we did not make use of it in our code. However, having a functional example of how to build an ECS architecture, even one with ever-changing details, in the same development environment we planned to use for our models was a tremendous help in designing and implementing our own architecture.

On the downside, while Unity can be used for building agent simulations, it is rarely used as such, at least when it comes to academic models. The main reasons for this are twofold. First, Unity carries a far higher overhead in term of time and effort for implementing ABMs compared to dedicated frameworks such as Netlogo. This overhead comes mostly from the fact that Unity doesn't offer much out-of-the-box support for common ABM features, such as simulation stepping, grid or network structures, out-of-the-box visuals, or data gathering and plotting. Modelers who want to use Unity need to build their own, which can be time consuming. Second, Unity uses C# as a programming language and many of the researchers developing ABMs are not experienced coders. This makes ABM dedicated frameworks like Netlogo, which has its own programming language specifically designed to appeal to non-programmers, much more preferable as a development medium. However, in our case, the effort-to-payoff ratio skewed significantly towards Unity.

[1] https://unity.com/dots

5.3 Cognitive building blocks

Starting from the conceptual architecture, we require that an agent contains any combination of actions, norms, policies, goals and values, and one active reasoner. The agent must contain at least one action, normative or not. It must contain at least one type of motivator, either one goal, normative or not, or a value tree.

As such, the implementation treats actions, norms, goals, values and reasoners as arbitrary data, and we'll refer to them as the "(cognitive) building blocks" of the agent.

We distinguish between data-storing building blocks and behavioral building blocks. Data-storing building blocks are components and entities. Behavioral building blocks are actions, norms, policies, goals, value nodes and trees, and reasoners.

5.4 Data-storing building blocks

5.4.1 Components

The first step in building the implementation of these soft-coded building blocks is to render their associated data arbitrary, which requires a bit of explanation.

The difficulty with implementing this kind of architectural pattern comes from the fact that C# is a strongly typed language, which means that variables are tied to data types, and operations on these variables will throw errors if the wrong types are used. For example, if we want to create an entity **Vessel** whose list of components includes a **cost** of type **float** and an **owner** of type **string**, we run into the C#'s refusal to allow these two variables to exist in the same list because they are of different types. Our solution is to represent all components as custom types, all of which share the same base custom **BaseComponent** type. However, because **BaseComponent** needs to be a parent to components containing data of any type, it cannot itself contain any data, nor any typed data retrieval functions. In order to retrieve data from a specific **Component**, we add an intermediary layer in this shallow hierarchy, which contains one generic parent for every data type in the implementation. These intermediary types contain the necessary functionality for accessing their assigned data type. All specific component types containing the same type of data become children of the intermediary types. For a partial view of the component type hierarchy implemented for the models described in this book, see fig. 5.4.

While at first glance it appears that this would result in an explosion of component types, this is not the case. The codebase does contains a larger number of types compared to an implementation where this data is represented as variables, but components can be reused as needed to create new entities so the number of sibling component types in any given model is limited, and many are shared between models as well. For instance, things like **Cost**, **Owner** or **Location** are likely to be part of many different entities such as **House**, **Shop** or **Car**, which, in turn, are likely to be part of many different models or scenarios.

It is important to note that component types are not limited to the C# built-in types.

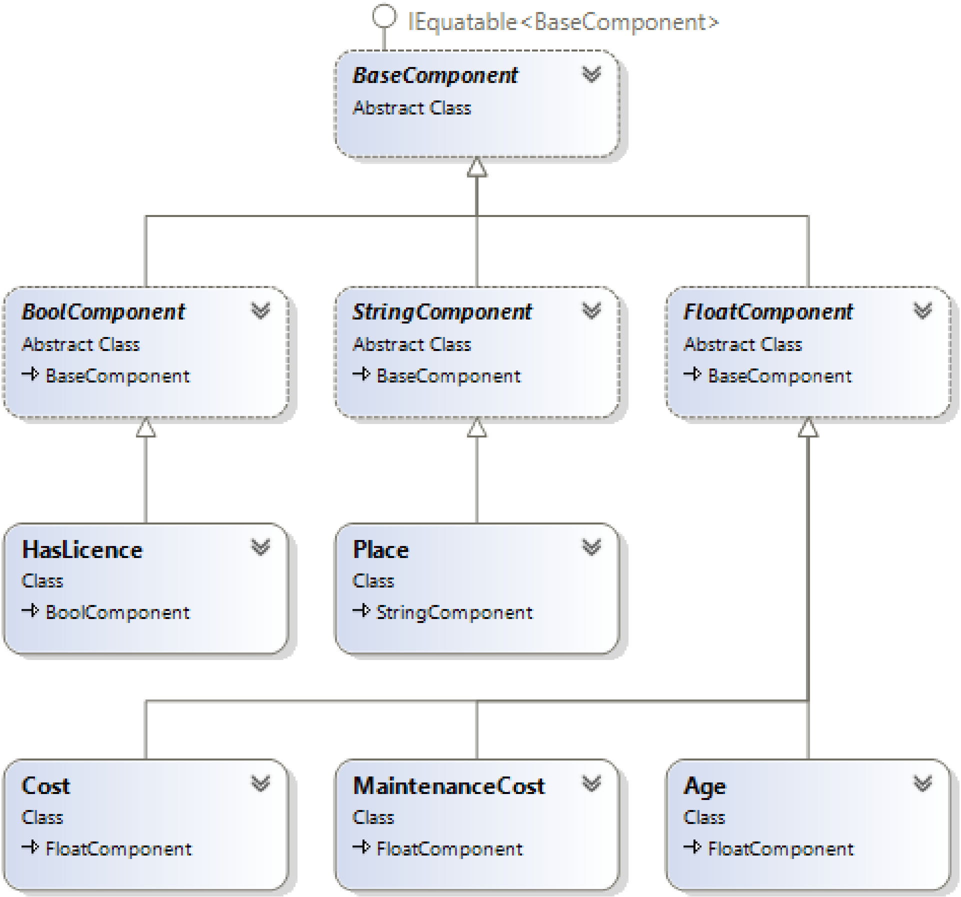


Fig. 5.4: The diagram shows some of the types of components in this implementation.

An example of the code of a component class can be found in code snippet 5.1.

```
public class Cost : FloatComponent
{
    public Cost()
    {
        name = "Cost";
    }

    public Cost(Cost c)
    {
        name = c.name;
        value = c.value;
    }

    public override BaseComponent Duplicate()
    {
        return new Cost(this);
    }
}
```

Code 5.1: Component code example

5.4.2 Entities

As mentioned previously, entities are basically just lists of components. As such, different entities are not actually different types. Once the **Entity** type is implemented, it can be used to build any number of entities, containing any number and combination of components.

To distinguish between different kinds of entities, we use names and archetypes (see fig. 5.5). In keeping with the conceptual architecture, archetypes are used to store the kinds of components the entity is made of. The components in archetypes do not have values, serving merely as a blueprint from which to generate new entities as needed.

Since they are basically lists, entities can be modified at any time, including at runtime, both in their values and their structure. It's important to remember to also modify the archetype when modifying an entity's structure since any future entities will need to use the archetype as a blueprint.

Since the entities are all the same type (class), an agent needs to contain only a list of entities in order to represent their internal state and their knowledge of the world. This means we do not need to build multiple agent types if our model calls for agents with different kinds of knowledge. Furthermore, this knowledge can be modified at any time, including at runtime, by adding or removing entities from the list. This spares us from having to plan agent types in advance if we plan to simulate scenarios in which the policy changes include changing associated concepts (such as removing **Licences** and replacing them with **Quota**) because we can simply give the agents the necessary entities when required.

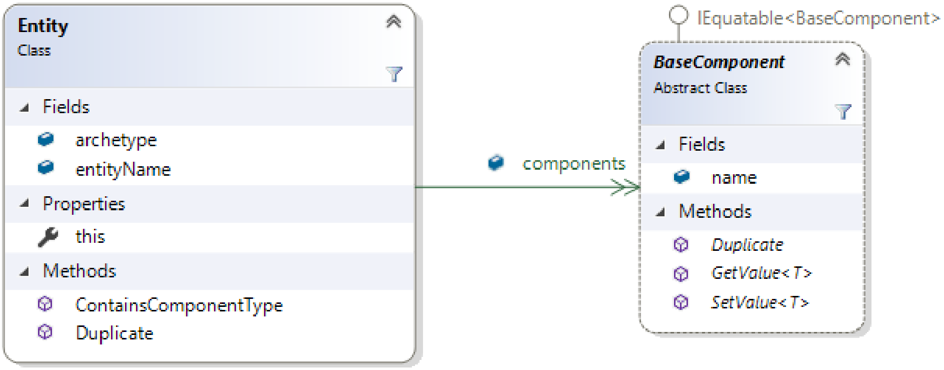


Fig. 5.5: Diagram of the **Entity** class.

5.5 Behavioral building blocks

All the behavioral building blocks defined in the conceptual architecture are composed of a number of functions which operate with/on specific entity kinds. The same structure is maintained in the code.

In order to create a type which contains an arbitrary function, we make use of C#'s **Func** and **Action** delegates, which are typed function pointers. Delegates contain references to methods, so, by using them, we don't need to specify *which* method should be invoked, just that *a* method with the parameter and return types specified by the delegate will be invoked. These methods don't need to be part of the same class as the delegate, which means we can literally store behavioral building blocks in dedicated libraries, completely separated from the actual agent code.

While the behavioral building blocks differ in structure between kinds (e.g. actions have a different structure from goals), they all share the same structure within the same kind (e.g. all actions have pre-conditions, transformations and effects functions). Since we've replaced the specific functions with delegates, each kind of building block can be represented by just one corresponding type. For instance, rather than having a **Fish** action, a **SellFish** action and a **PayDebts** action, we only have the one **ModelAction** type, which we configure into the three different actions by loading its delegates with the appropriate functions from the dedicated action library. This representation also lets us share functions between building blocks, as long as the function signature matches the delegate signature.

The one glaring issue with our approach is that these functions need access to their owner agent internal state in order to carry our their computation. In order to bypass this problem, the agent is passed as a parameter in all the functions. This way, the function doesn't need to know in advance who its owner is, just needs to request the entity list from any agent it receives as part of its input.

What all this means is we do not need different agent classes based on their behavioral capabilities. The agent itself can be fully agnostic when it comes to its behavioral building blocks, excepting perception, which we did not address, either in the conceptual or implemented architectures. All an agent needs is a list of actions (normative, policy or neither), a list of goals (normative, policy or neither), a value tree and an appropriate reasoner. As with data-storing building blocks, behavioral blocks can be changed at runtime, either by adding, removing or replacing them as needed.

5.5.1 Actions, normative actions and policy actions

The conceptual architecture defines actions as consisting of three functions: one to asses whether the action can be performed, one transformation that acts upon the world, and one to assess the possible effects of the transformation. The implemented version follows the same structure, with one significant difference. The conceptual architecture defines norms are being composed of a context assessing activation function and either an action or a goal. In the implementation, there is no separate **Norm** type. Rather, the **ModelAction** type already contains the necessary structure to turn into a norm at the flip of a flag variable. Flipping a different flag turns the action into a policy. The **ModelAction** type contains the delegates for the context activation function and for the value assessment function (see fig 5.6).

In addition to these delegates, the **ModelAction** type also contains entity lists corresponding to the domain of the feasibility function, the codomain of the effects assessment function and the domain of the context assessment function in the case of normative or policy actions. This is not strictly necessary from a functional point of view since all these functions receive the agent as one of their input parameters. However, these additional entity lists serve as quick filters for actions that require or affect certain entities, which is a handy bit of added utility to have. Figure 5.7 shows the visual representation of a basic action the model editor together with a norm as expansion of a basic action after the **Is Norm** box has been ticked.

5.5.2 Goals, normative goals and policy goals

Goals are represented conceptually as one function that assesses whether the goal is fulfilled or not. As with actions, there is no separate type for normative or policy goals, rather the **Goal** type already contains the delegate for context assessment and allows any goal to be turned into a normative or policy goal by flipping the corresponding flag variable (see fig 5.8). Goals also

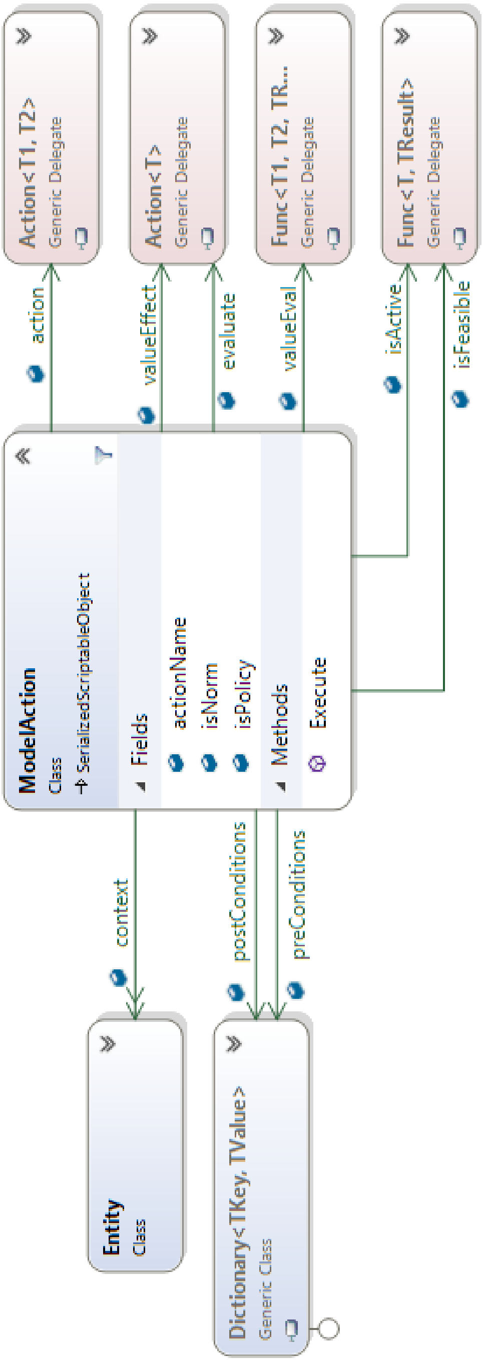
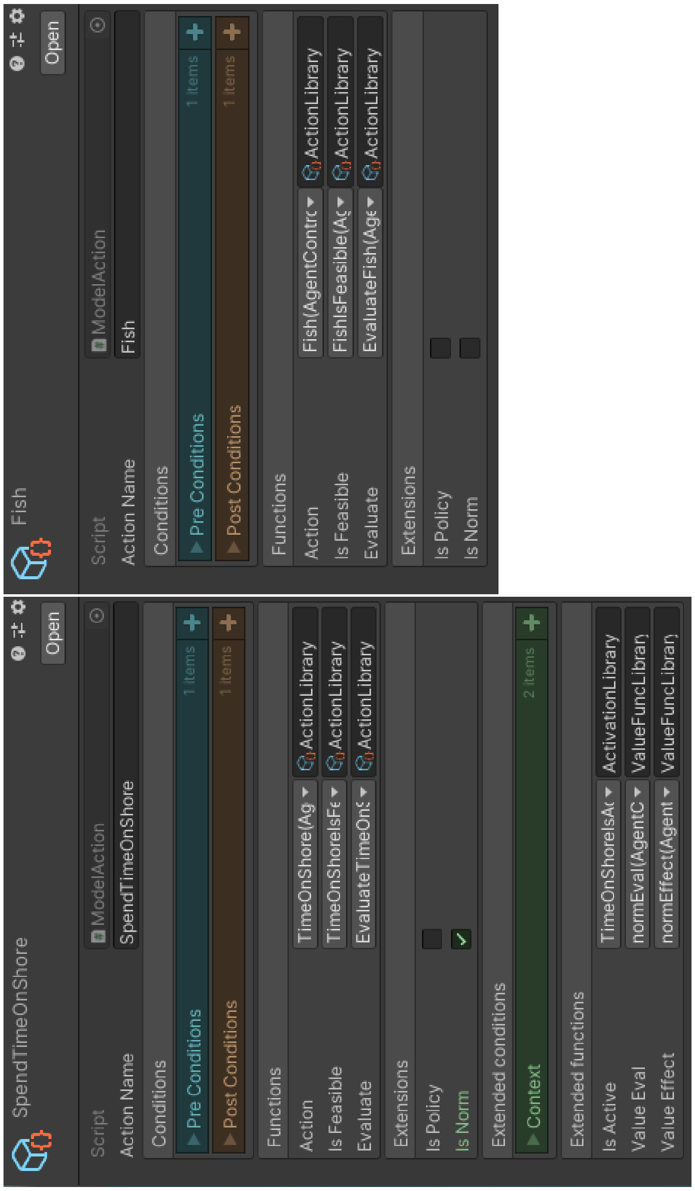


Fig. 5.6: Diagram of the action class



Fish

Open

Script ModelAction

Action Name Fish

Conditions

▶ Pre Conditions 1 items

▶ Post Conditions 1 items

Functions

Action Fish(AgentContr ▾ ActionLibrary

Is Feasible FishIsFeasible(Ag ▾ ActionLibrary

Evaluate EvaluateFish(Age ▾ ActionLibrary

Extensions

Is Policy □

Is Norm □

SpendTimeOnShore

Open

Script ModelAction

Action Name SpendTimeOnShore

Conditions

▶ Pre Conditions 1 items

▶ Post Conditions 1 items

Functions

Action TimeOnShore(Ag ▾ ActionLibrary

Is Feasible TimeOnShoreIsFe ▾ ActionLibrary

Evaluate EvaluateTimeOnS ▾ ActionLibrary

Extensions

Is Policy □

Is Norm ☑

Extended conditions

▶ Context 2 items

Extended functions

Is Active TimeOnShoreIsAc ▾ ActivationLibrary

Value Eval normEval(AgentC ▾ ValueFuncLibrary

Value Effect normEffect(Agent ▾ ValueFuncLibrary

Fig. 5.7: Visual representation of a basic action (top) and a normative action (bottom) in the model editor

contain lists of entities that act as the domains of both the fulfillment and context assessment functions, which serve no strictly necessary functional purpose, but are handy as filters.

Figure 5.9 shows the visual presentation of a basic goal in the model editor.

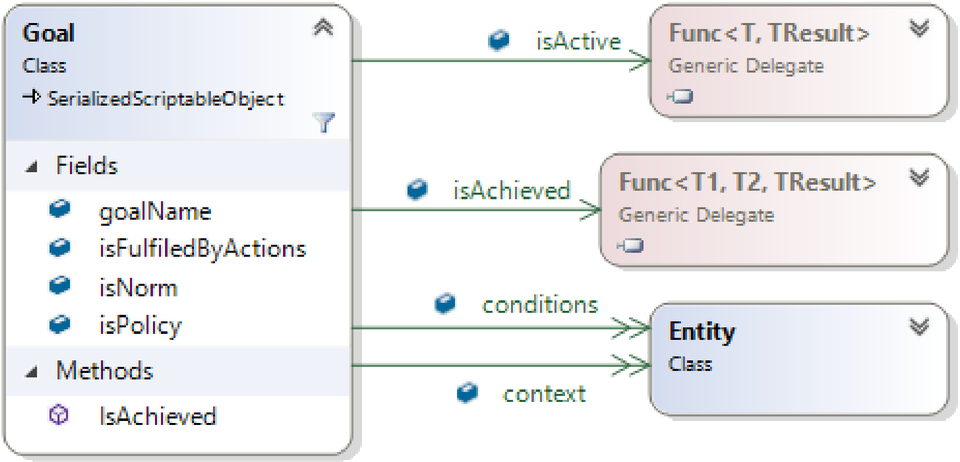


Fig. 5.8: Diagram of the goal class

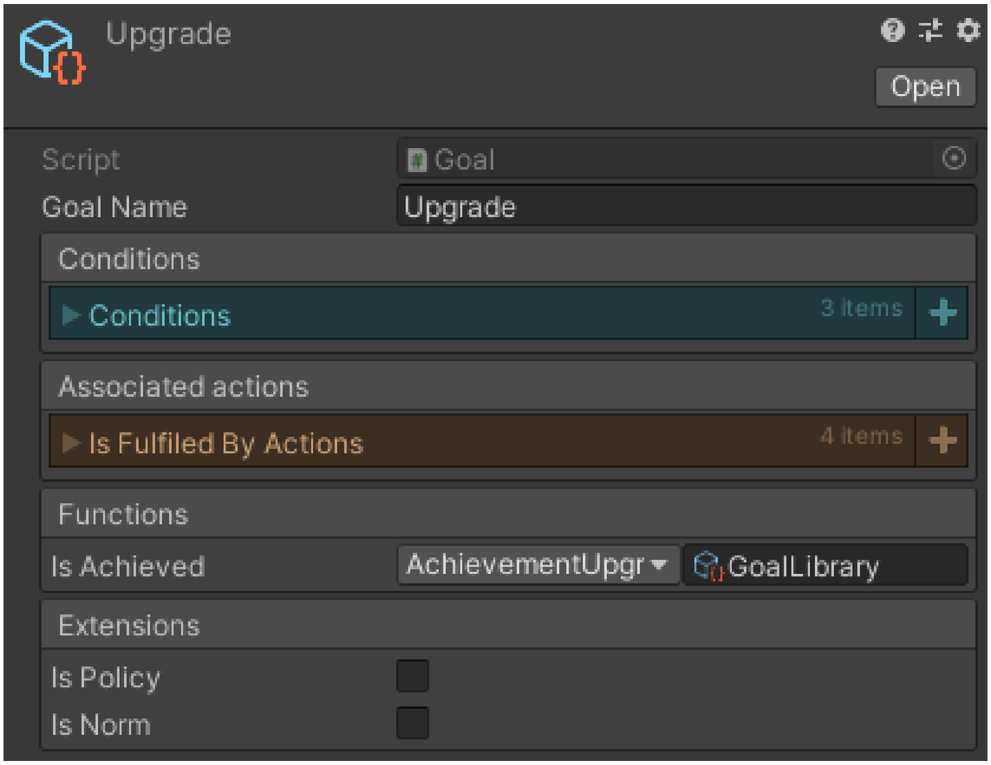
5.5.3 Value systems

The implementation of the value system closely follows the specifications of the conceptual architecture. The implemented value tree is composed of three different kinds of nodes: leaf nodes, intermediary nodes, and root nodes, just like its conceptual counterpart.

Leaf nodes are defined by a preference function over a subset of the entities that comprise the world state. These entities are not children of the nodes, but are stored in a list. As with actions and goals, the list of entities isn't strictly necessary when it comes to functionality, and exists in the implementation only as a handy utility.

Intermediary nodes are defined by a preference function over the preference values of their children, which can be leaves or other intermediary nodes.

Root nodes correspond to fundamental Schwartz values and are defined as preference functions over the preference values of their children, which can be intermediary or leaf nodes. Since these are the nodes that ultimately determine the overall value fulfillment of an agent, they also contain a depletion rate variable and a threshold variable used by the value system when calculating overall fulfillment.



Script: Goal

Goal Name: Upgrade

Conditions

Conditions — 3 items

Associated actions

Is Fulfiled By Actions — 4 items

Functions

Is Achieved: AchievementUpgr GoalLibrary

Extensions

Is Policy □

Is Norm □

Fig. 5.9: Visual representation of a basic goal in the model editor

All nodes are implemented as a single **ValueNode** type (see fig. 5.10) whose preference function is represented as a delegate, the same pattern we used for actions and goals. The actual functions are stored, as before, in their own dedicated libraries. Because they are all of the same type, we use labels to distinguish between kinds of nodes.

The **ValueSystem** type (see fig. 5.10) contains the list of root nodes and functions for calculating current fulfillment and evaluating future fulfillment should a given action be executed in during the current simulation step. The **ValueSystem** is one of the cognitive blocks that is almost fully hard-coded. This is because the current theoretical architecture doesn't allow for other conceptualizations of the value system. There was no need to invest effort in investigating ways to allow for the behavioral configuration of an element that currently exhibits a single behavior type in the absence of any information for what future variations of said behavior might look like.

Figure 5.11 shows the visual presentation of a value system in the model editor. For a more intuitive, but sadly not editable, visual representation of a value tree, see fig. 5.12

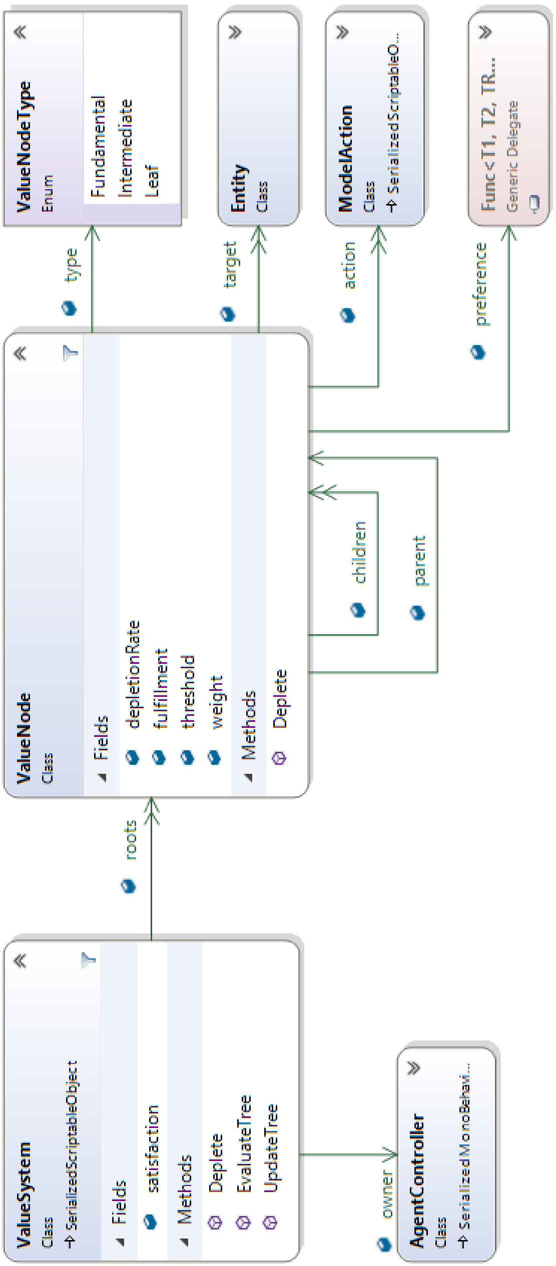
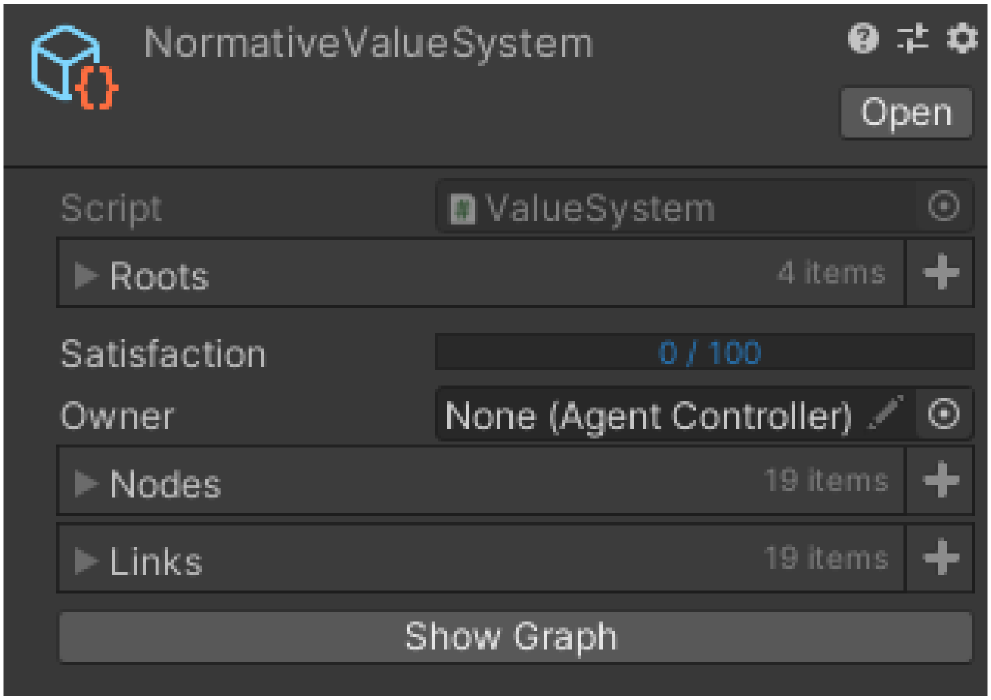


Fig. 5.10: Diagram of the **ValueTree** and **ValueNode** classes.



NormativeValueSystem

Script ValueSystem

Roots 4 items

Satisfaction 0 / 100

Owner None (Agent Controller)

Nodes 19 items

Links 19 items

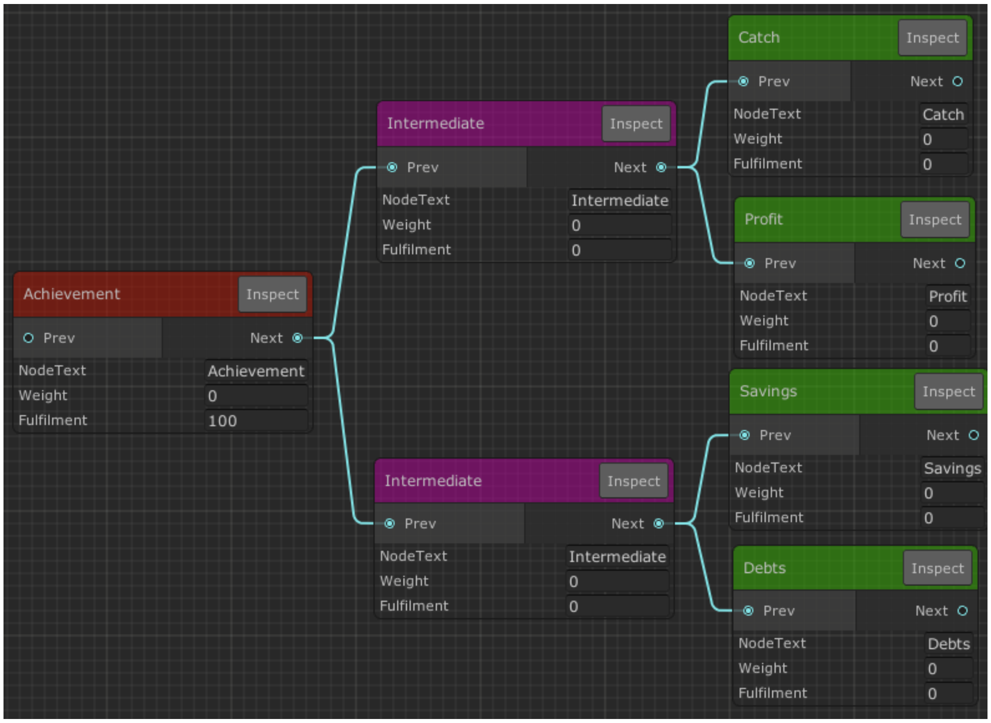
Fig. 5.11: Visual representation of a value system in the model editor. This visualisation allows the editing of the value tree

5.5.4 Reasoners

Just like all the other cognitive building blocks, the reasoners are implemented in a form that makes them interchangeable from the point of view of the agent. However, unlike the rest of the blocks, we did not deem it necessary to use a similar delegate and dedicated library pattern. This is in part because we do not expect reasoners to be configurable in the same way that actions, goals or value trees are. Reasoners need to be specific to the cognitive makeup of the agent. For example, if we build an agent with actions and goals, but no norms and no values, we have to pick a reasoner that can operate on actions and goals without the need for additional motivators. If we build an agent with actions and values, we would need a different kind of reasoner.

All reasoners are derived from the base **Reasoner** type, which contains one publicly accessible function: **Reason**, the entry point for the reasoning process. Any reasoner type derived from the base **Reasoner** overrides this function in order to provide its own specific type of calculations.

The architecture includes two out-of-the-box reasoners designed to work with the cognitive blocks that make up the different agent kinds we use in our models (see fig. 5.13). The base



NodeText Achievement
Weight 0
Fulfilment 100

NodeText Intermediate
Weight 0
Fulfilment 0

NodeText Intermediate
Weight 0
Fulfilment 0

NodeText Catch
Weight 0
Fulfilment 0

NodeText Profit
Weight 0
Fulfilment 0

NodeText Savings
Weight 0
Fulfilment 0

NodeText Debts
Weight 0
Fulfilment 0

Fig. 5.12: Visual representation of a value tree in the model editor. This visualisation is not editable

reasoner type operates on actions, norms, policies and goals (see code box 5.2) . The value reasoner is derived from the base reasoner and overrides its initialisation and main reasoning function in order to be able to operate on actions, norms, policies and values (see code box 5.3). To avoid the aforementioned risk of mismatch between the agent's makeup and its reasoner, the appropriate reasoner is instantiated at runtime, based on the composition of its agent owner, without input from the modeler.

Because the architecture, so far, makes no provisions for plans, especially interrupting and recalculating plans, whatever action the reasoner chooses is the action that will be executed. Because of this, the execution of the action is carried out by the reasoner itself, without the need of an additional **Excutive** element.

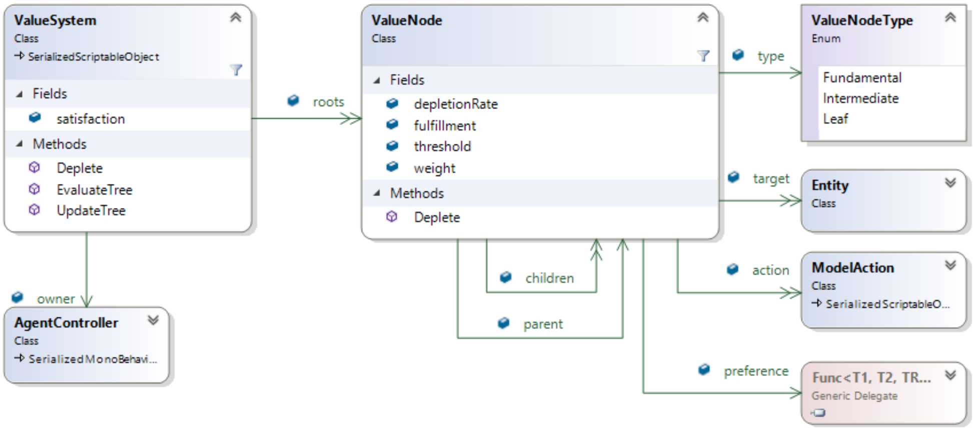


Fig. 5.13: Diagram of the Reasoner and ValueReasoner classes.

5.6 Agents

As discussed in somewhat more detail in previous sections, because of the modularity of the implementation, we can actually manage quite a few number of models by using only one **Agent** type (see fig. 5.15). This is possible in our case because the models have significant overlap in agent knowledge and thus we could simply hard-code the perception function as part of the agent type itself.

Since there's only one agent type, we distinguish between kinds of agents by using profiles. Each agent profile specifies the entities, actions, goals and value tree that will be used to configure its corresponding agents (see fig. 5.14).

5.6.1 Agent makeup

In addition to its hard-coded perception, the agent type contains collections of entities, actions, norms, goals, a value tree and a reasoner. These collections and elements are empty for every newly created agent instance and need to be populated before the agents can be used. This is done by referring to an agent profile.

Agents contain two separate collections of entities. The first serves as the perceived world state (including internal state). The second, called the hypothetical world state, is a copy of the first, and it is used by actions to evaluate their effects. In short, agents can apply an action, through its effects evaluation function, to this second collection of entities in order to see how the world is likely to change if they were to execute that action during the current simulation

```
if goalQueue is empty
    add all goals to goalQueue

forall (goals in goalQueue)
    if (goal is not fulfilled)
        if (goal is normGoal)
            if (goal is active)
                add goal to normGoalList

        else if (goal is policyGoal)
            if (goal is active)
                add goal to policyGoalList

        else add goal to goalList

if (normGoalList is not empty)
    nextGoal is random goal from normGoalList
else if (policyGoalList is not empty)
    nextGoal is random goal from policyGoalList
else
    nextGoal is random goal from goalList

forall (actions in availableActionsList)
    if action fulfills nextGoal
        if (action is normAction)
            if (action is active)
                add action to normActionList
        else if (action is policyAction)
                if (action is active)
                    add action to policyActionList
        else add action to actionList

if (normActionList is not empty)
    nextAction is random action from normActionList
else if (policyActionList is not empty)
    nextAction is random action from policyActionList
else
    nextAction is random action from actionList

execute nextAction
```

Code 5.2: Reasoning loop for base reasoner

step. The accuracy of their foresight depends on the action's evaluation function. If they intend to execute an action which operates on entities that are not affected by any other forces outside the agent itself, then the evaluation function can be the same as the transformation function, except it's applied to the agent's internal copy of the current perceived world state.

For instance, **PayDebts** is an action that affects only the agent's current amount of money and, unless the model contains something like theft or sudden financial disaster, nothing else

```
update value tree satisfaction
forall (actions in availableActions)
    if (action is normAction)
        if (action is active)
            add action to normActionList
    else if (action is policyAction)
        if (action is active)
            add action to actionList
    else add action to actionList

if (normActionList is not empty)
    nextAct = max(valueSatisfaction(action in normActionList))
    execute nextAct
    break

else
    nextAct = max(valueSatisfaction(action in actionList))
    execute nextAct
```

Code 5.3: Reasoning loop for value reasoner

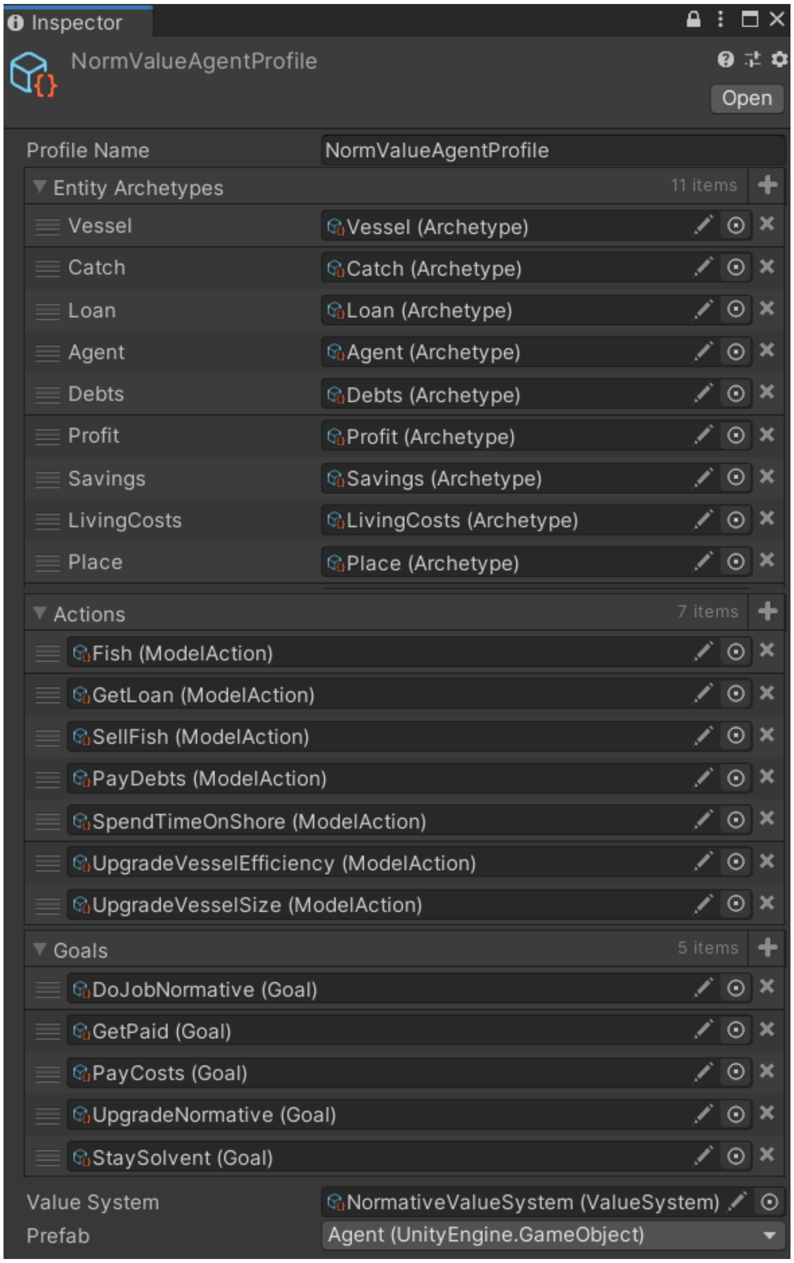
will act on the agent's money in the current simulation step. Therefore, the hypothetical future world state is the same as the actual future world state and the results of the effects evaluation function are the same as the ones of the transformation function. However, an action like **Fish** has results that depend on factors outside the agent's control or knowledge, such as the size of the fish population. In this case, the agent cannot know for certain what the effects of the action will be and the effects evaluation function needs to reflect that. In the implementation used in our model, the transformation function of this action communicates with the **FishingArea** object in order to determine the value of the **Catch**. The effects evaluation function estimates a likely value for the **Catch** based on the agent's **Vessel** entity and the value of previous catches.

5.6.2 Agent loop

Due to the modularity of this architecture, the basic agent loop is exceedingly simple, with most of the functionality being provided by whatever cognitive building blocks form the agent's makeup. It consists of three steps:

Step 1: Update internal state (perception) Step 2: Reason (assess motivators and choose next action to be performed) Step 3: Execute the chosen action

While it appears exceedingly simple, this loop is definitely a case of hidden complexity since the agent can contain any number and type of motivators and the reasoners can be as complex as desired. Our intention here is not to oversimplify the agent loop, but to distribute as much



NormValueAgentProfile

Profile Name	NormValueAgentProfile

Entity Archetypes 11 items

Vessel	Vessel (Archetype)
Catch	Catch (Archetype)
Loan	Loan (Archetype)
Agent	Agent (Archetype)
Debts	Debts (Archetype)
Profit	Profit (Archetype)
Savings	Savings (Archetype)
LivingCosts	LivingCosts (Archetype)
Place	Place (Archetype)

Actions 7 items

- Fish (ModelAction)
- GetLoan (ModelAction)
- SellFish (ModelAction)
- PayDebts (ModelAction)
- SpendTimeOnShore (ModelAction)
- UpgradeVesselEfficiency (ModelAction)
- UpgradeVesselSize (ModelAction)

Goals 5 items

- DoJobNormative (Goal)
- GetPaid (Goal)
- PayCosts (Goal)
- UpgradeNormative (Goal)
- StaySolvent (Goal)

Value System	NormativeValueSystem (ValueSystem)
Prefab	Agent (UnityEngine.GameObject)

Fig. 5.14: Visual representation of an agent profile in the model editor

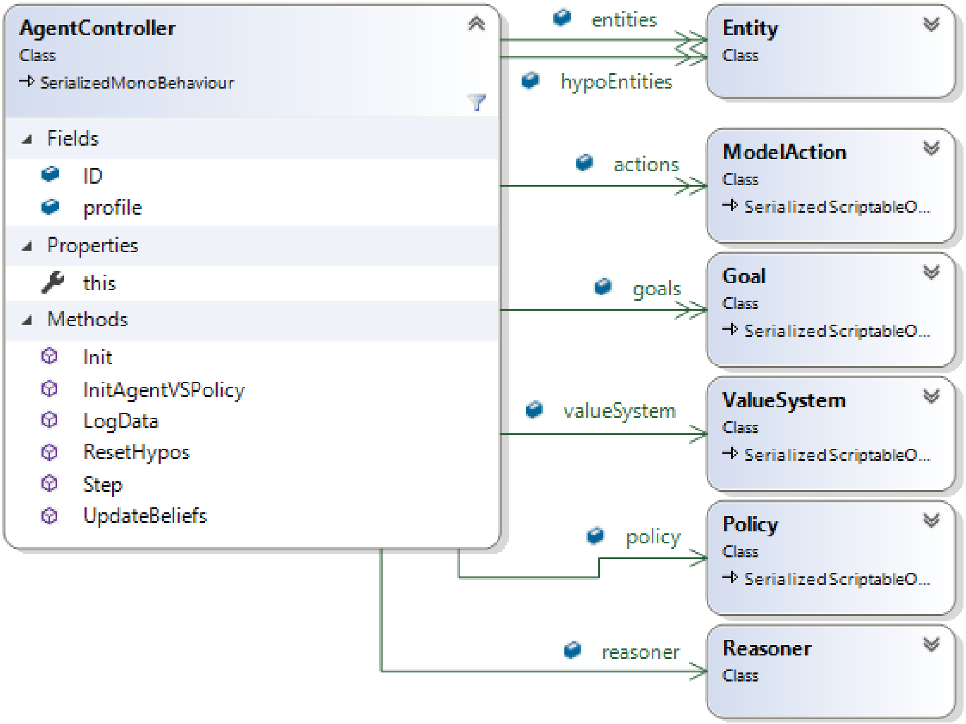


Fig. 5.15: Agent diagram

of its calculations as possible into independent elements that can be mixed and matched to create any number of agents on the fly.

5.7 The Policy type

In our implementation, the **Policy** type doesn't refer to the action or goal policies the agents use in their reasoning. Rather, the **Policy** type serves as a runtime scenario modifier. It contains lists of entities, actions and goals that get added to, modified or removed from the agents' makeup when the policy comes into effect. In addition, it may contain special functions that are needed to carry out policy specific calculations that none of the agents are actually empowered to conduct themselves. For instance, a quota **Policy** will reduce the **Vessel** entity of every fisher agent to a neutral version devoid of any components deriving from other policies, such as a **License** component, then add the **Quota** entity to all fisher agents. This particular policy involves the calculation of initial quotas and their yearly updates, actions which none of the fisher agents are authorized to perform. Thus, the implementation allows the quota **Policy** to carry these actions out itself through its special functions.

Same as the behavioral building blocks, there is only one **Policy** type. Its special functions are represented by delegates and the actual functions are stored in their own dedicated library.

For a visual representation of a **Policy** configuration, see figure 5.16.

5.8 Saving data

The level of modularity present in this implementation is at its most useful when the various agent types and their building blocks can be saved and reused as needed. Without support in managing the load of configuration data required to turn the soft-coded types into actual functional code, any actual model building would require much more effort, maybe to the point where the flexibility gains would not be worth the additional hassle of keeping track of the extra files.

Since we're using Unity, we took advantage of their particular type of data container called **ScriptableObject**. Specific configurations of building blocks and profiles of agents, simulations and batches are all saved as **ScriptableObjects** in their respective libraries (see fig. 5.17).

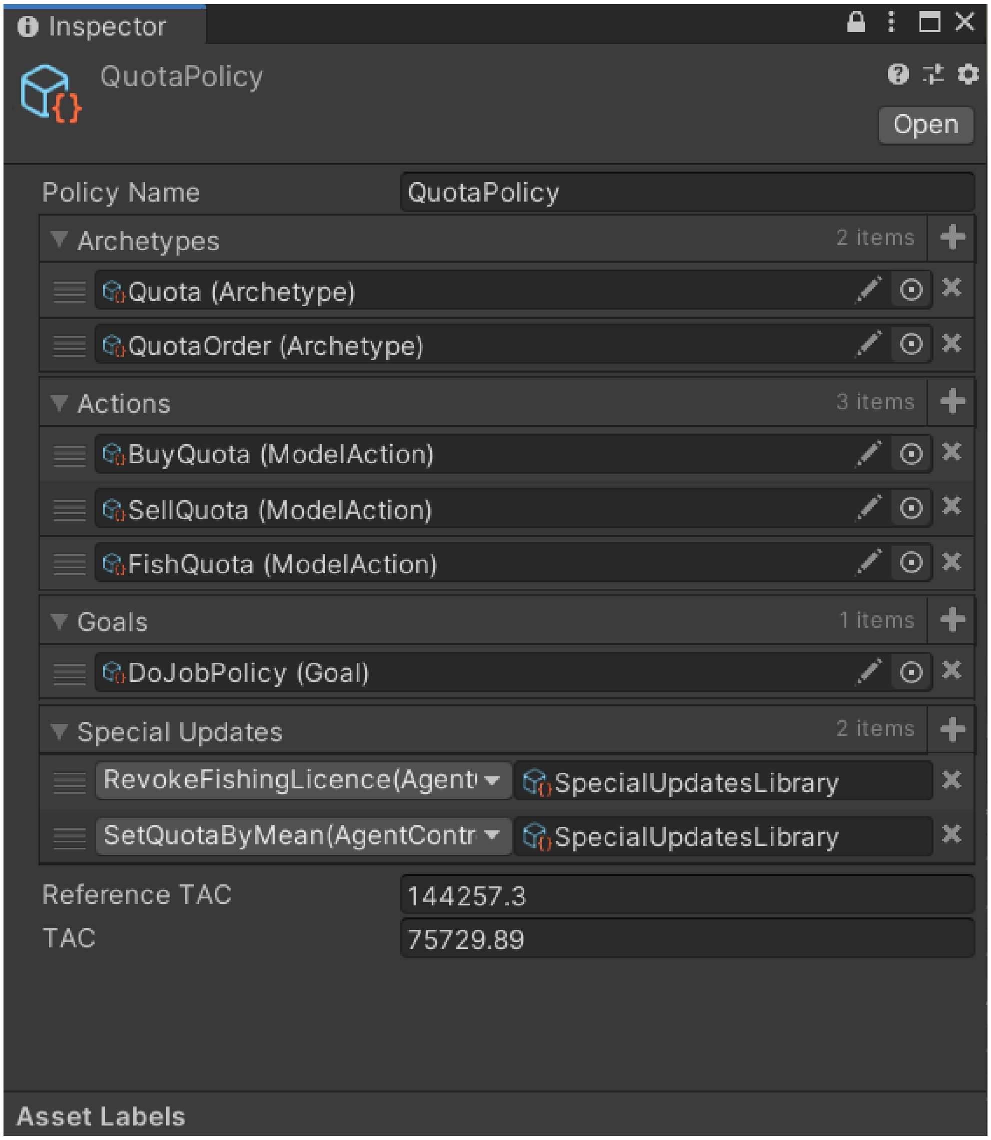
Aside from saving the data associated with implementation elements, **ScriptableObjects** can be created and edited from the Unity Editor, which makes it very easy to quickly create and configure new cognitive building blocks, without the need to write additional configuration code or to maintain setup files for new building blocks, agent types or simulation scenarios.

To have a look at what a **ScriptableObject** of an entity archetype, for example, looks like in our Unity model editor, see fig. 5.18. Similar visual representations exist for all the soft-coded types in the implementation, all of them displaying the **ScriptableObject** file which stores that specific configuration of the action, goal etc.

Further details about the visual model builder associated with our implementation, see the next section.

5.9 Visual interface

The visual interface we developed for this implementation serves two purposes. First, it significantly reduces the effort required to configure the elements of the model. As it stands, it is very easy to flip through the editor windows and quickly put together new actions or goals or agent profiles, despite the numerous elements that need to be brought together. Building a new model is fairly effortless because the interface allows the user to keep all the relevant data



Policy Name QuotaPolicy

Archetypes 2 items
- Quota (Archetype)
- QuotaOrder (Archetype)

Actions 3 items
- BuyQuota (ModelAction)
- SellQuota (ModelAction)
- FishQuota (ModelAction)

Goals 1 items
- DoJobPolicy (Goal)

Special Updates 2 items
- RevokeFishingLicence(Agent SpecialUpdatesLibrary
- SetQuotaByMean(AgentContr SpecialUpdatesLibrary

Reference TAC 144257.3

TAC 75729.89

Fig. 5.16: QuotaPolicy in GUI

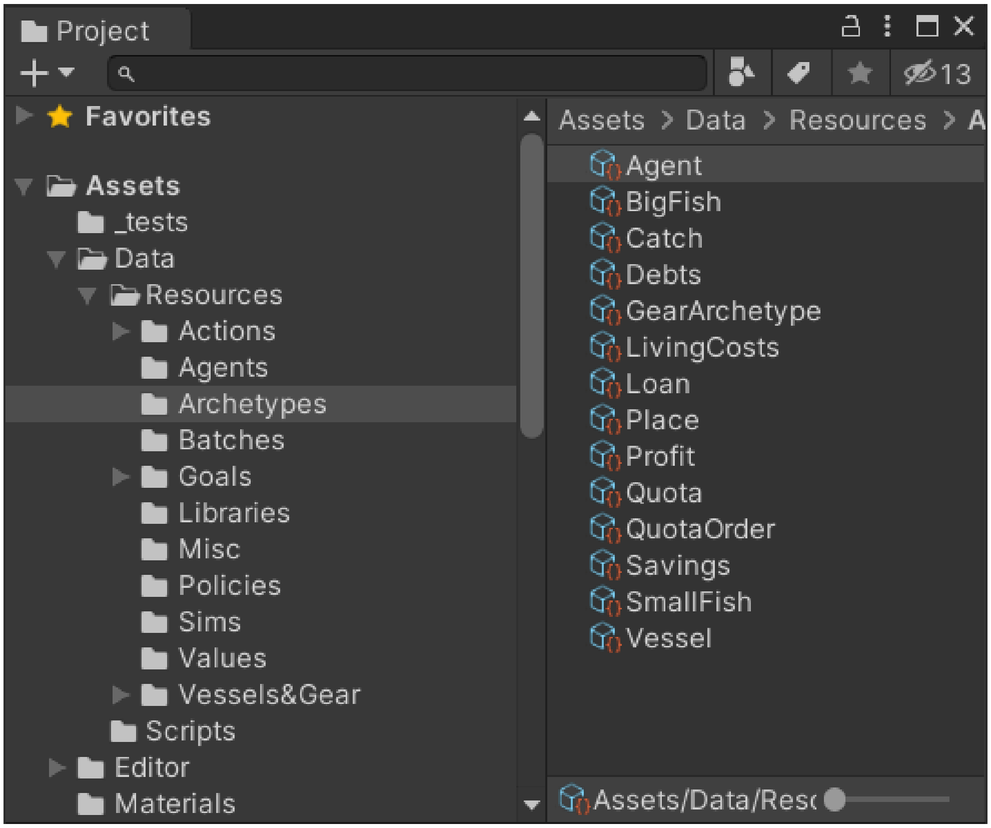
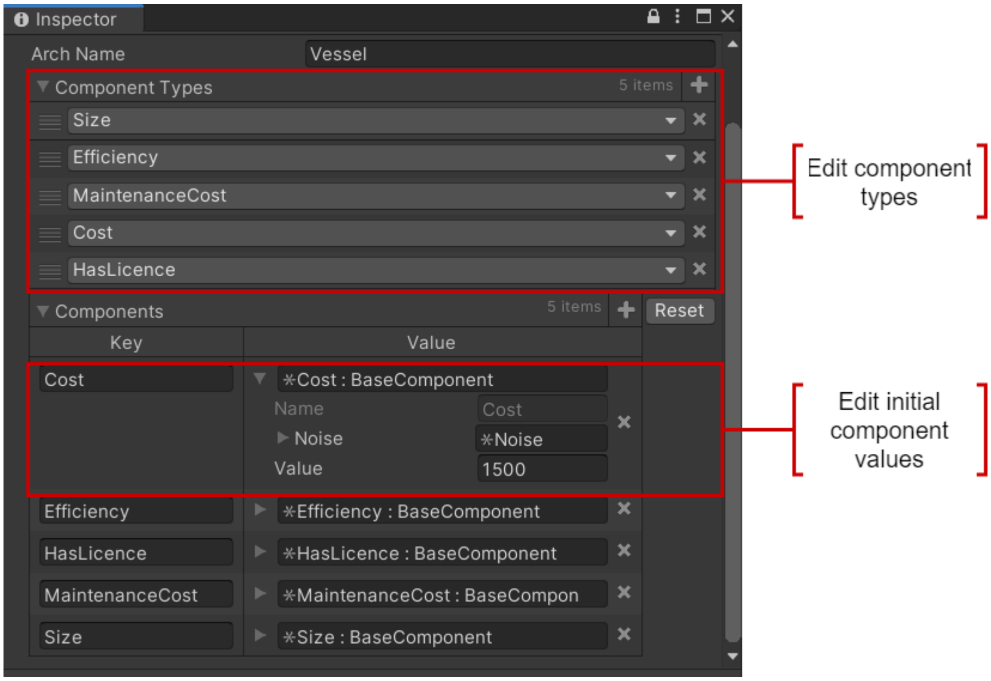


Fig. 5.17: Libraries of building blocks in the current implementation. Left: a list of library folders. Right: the archetypes saved in the Archetype library. Not all the archetypes listed in this figure are used in the models described in this book, but they have been used for various tests, and, once created, they are kept in case of future need.

within their visual field at all times, saving them the effort of memorizing the structure of every element present in the model builder libraries, and the time spent scrolling through config file after config file in search of previously created elements that could be reused in the model they're currently building. Like some other implementation choices described in this chapter, the visual interface is not *needed* from a functional point of view. However, from the perspective of convenience, the visual interface makes a world of difference.

The second purpose of the implementation is communication. We have talked extensively in chapter2 about the importance of including stakeholders in the policy design process, especially when the subject is a system as complex and dynamic as a SES. Involving users in the design of a model which includes said policy is part of that process, but it is not a straightforward one. Bridging the communication gap between the dry, information-dense academic style and



Arch Name: Vessel

Component Types (5 items)
- Size
- Efficiency
- MaintenanceCost
- Cost
- HasLicence

Components (5 items) Reset

Key	Value
Cost	*Cost : BaseComponent Name: Cost Noise: *Noise Value: 1500
Efficiency	*Efficiency : BaseComponent
HasLicence	*HasLicence : BaseComponent
MaintenanceCost	*MaintenanceCost : BaseCompon
Size	*Size : BaseComponent

Fig. 5.18: Vessel archetype in GUI

the more casual style of the public can be a surprisingly difficult feat with no simple solution. It can be even more difficult in the case of a complex agent architecture like the one described in this thesis, which has many interrelated parts and processes that need to be explained before any model design can begin.

One of the best ways to communicate complicated structures is to use visual aids. Because the architecture is highly modular and most of its elements are designed to be generic empty structures until they are loaded with data and behavior, it is relatively simple to build a visual interface for the model and simulation design process. Since Unity's editor can be extended with custom elements, we can effectively turns the visual model builder into an extension of the Unity editor itself, which further narrows the gap between the design and implementation phases of model design.

Unfortunately, it is not possible to completely do away with the need for writing code in this implementation. The method libraries, in particular, are nowhere near regular enough to allow any kind of meaningful visual representation or manipulation. Other elements that must be coded are the agent's perception, new components, and any data collection required by the modeller.

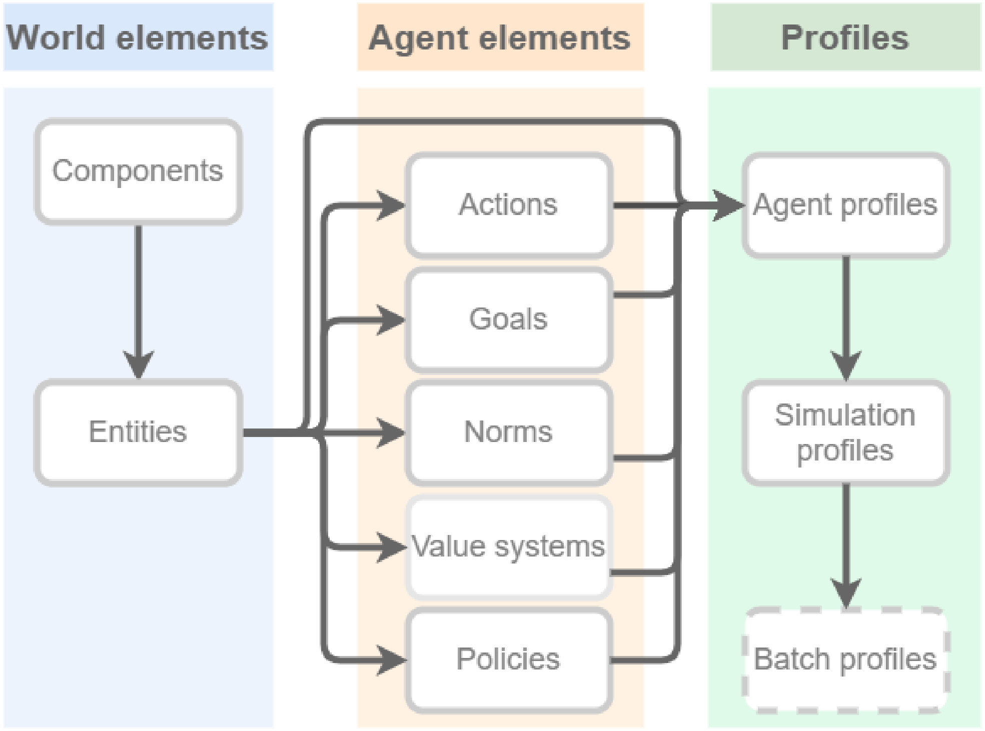
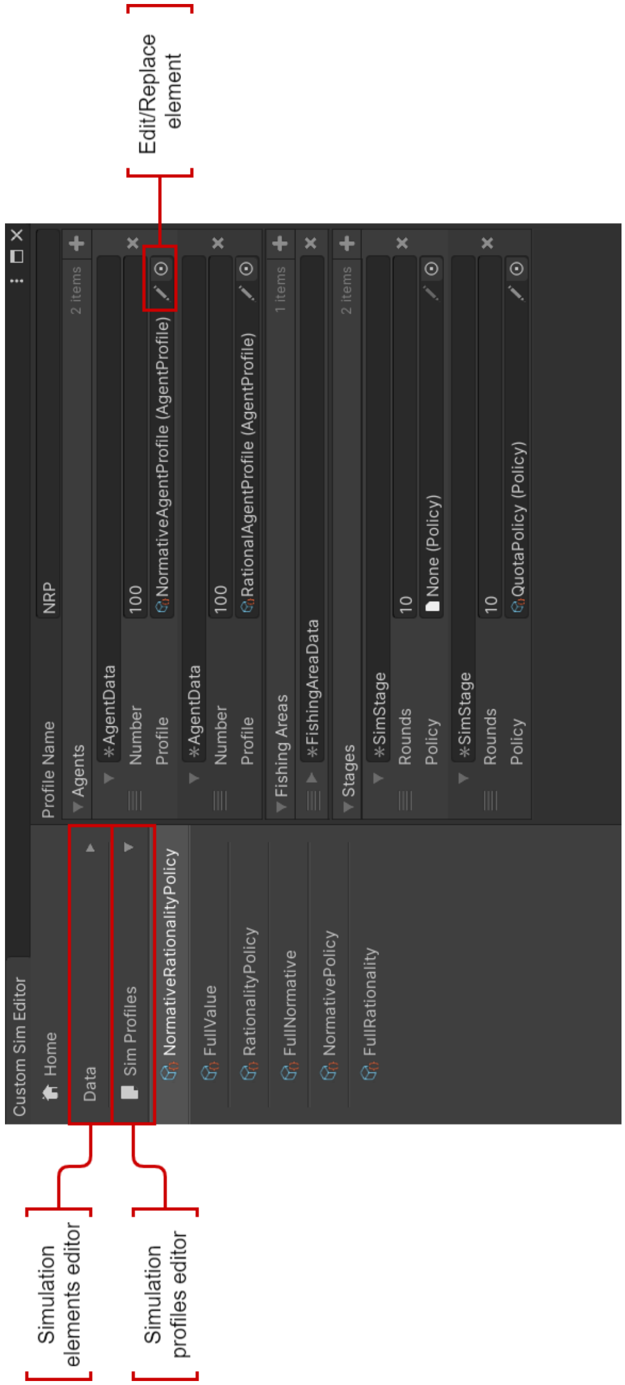


Fig. 5.19: Recommended build flow for the simulation elements using the visual interface

5.9.1 The model builder

In its current state, the implementation allows for the creation of new models through a combination of writing new code and using the visual interface. The architecture elements which require significant new code are new components, new functions for actions, goals, policies and values, new reasoners and agent perception. The recommended process for building a new model is illustrated in figure 5.19.

The start window for the model builder can be seen in figure 5.20. Every element of the architecture can be configured, saved and modified starting from here, or from the libraries shown previously in figure 5.17. Between these two, there should be no need to modify the definition of the existing types, unless the goal is to expand the architecture. Any newly created element, either code based (such as components or functions) or configuration based (such as entities, actions, goals, value systems) can be added to the corresponding library with minimal or no impact on the complexity of the code base.



Profile Name NRP

Agents 2 items

AgentData
Number 100
Profile NormativeAgentProfile (AgentProfile)

AgentData
Number 100
Profile RationalAgentProfile (AgentProfile)

Fishing Areas 1 items

FishingAreaData

Stages 2 items

SimStage
Rounds 10
Policy None (Policy)

SimStage
Rounds 10
Policy QuotaPolicy (Policy)

Fig. 5.20: Start window for the model and simulation builder

5.10 Limitations of the current implementation

5.10.1 Conceptual limitations

When it comes to the conceptual aspect of the implementation, we wish to mention two particular limitations. First, while the model builder interface is developed explicitly to support participatory modelling, no actual tests have as yet been performed in a participatory setting. As such, the interface must be considered a prototype or a proof-of-concept aimed primarily at proving that a visual model builder of this type can be created and used to build models on the underlying architecture described in this chapter.

Second, this interface can currently be used only during the model building phase. There is no corresponding interface that would allow the modeler to intervene in the simulation during runtime. All runtime changes, such as switching out one policy for another, must be configured and saved in a simulation profile. Despite the fact that the architecture is designed with the express purpose of allowing spontaneous runtime changes in the behavioral rules of the agents, the implementation lacks any controls, visual or otherwise, that would allow them to be performed. This is mainly because runtime interfaces in Unity carry significant overhead in terms of time and effort, and our main priority in this research was the development of the conceptual and implemented architectures, rather than user experience related to running simulations.

5.10.2 Architecture expansions

We mentioned that creating new cognitive building blocks based on types that already exist in the implemented architecture add very little complexity to the overall code base. This is because of the strict modularity and explicitness of the types and their configurations. The complexity doesn't grow because new actions, for instance, do not interact with existing actions, except through a reasoner. Even adding new types doesn't have to lead to an increase in complexity, except where reasoners are concerned and only if the new types are meant to interact with existing types. A new **Norm** type, for instance, doesn't have to exist in the same model as the existing normative versions of **ModelAction** and **Goal** types, which means there is no need for a new, more complex reasoner, which could operate on both at the same time. The codebase expands in data volume far more than it expands in complexity.

However, extending the existing types and adding new types specifically designed to interact with existing types (for instance, a **SocialPractice** or **Habit** type) will increase the local complexity of any model that uses them. At this point in our research, we cannot confidently offer either a way to quantify this complexity increase, nor a well-defined methodology for how to manage it in the long run. Again, this is because the main focus of the research was to develop the architecture and implementation first, with their optimization planned for later in the future.

5.10.3 Code dissemination

While Unity allows users to create custom expansions to the editor, the level of interaction displayed by our visual interface would have taken considerable time and effort to build by ourselves. In order to build this interface, we used the **Odin Inspector** asset[2], which makes the creation of custom editors much faster and easier. The only downside to this asset, in an academic environment, is that it is proprietary and lacks a free version that could be openly shared together with the model builder and model code.

[2] https://odininspector.com/

Chapter 6

Models

As we saw in chapter 3, the behaviour of fishers is often governed by different motivations to fish, norms regulating effort (in terms of catch, days at sea, efficiency in use etc.), discards and by-catch, social status, fishing grounds etiquette (do they push, do they crowd, are there favours, is there honest communication etc.), and reasons for entering and exiting the fishery. Despite this, models of fishing policy often fail to consider fishing anything more than an economic activity and rarely account for any other drivers of behavior outside of profit maximization. While such models have their place and purpose, we also want to be able to build models that capture more of the complexity of the actual decision making process of fishers, hence the architecture we developed and presented in chapters 4 and 5. In this chapter, we demonstrate how this architecture, and the "sandbox" setup that comes with it, can be used to build models and what kinds of information such models can impart that may be lacking in those built from ad-hoc rules or based on purely "rational" agents.

We base elements of our sandbox on the real life case of Norwegian small-scale fisheries and their reaction to the introduction of quota policy, as described in [89]. The introduction of the quota system reduced overall catch in the fishery, but a subset of fishers actually increased their fishing effort in response to the policy. If the usual assumption of fishing as a predominantly economic activity, with profit as the main driver, were correct for all fishers, this subset of fishers who got to fish more and thus make more money should have been quite happy with the new policy. But this is not what happened. Rather, the fishers complained about having to spend more time at sea in order to fish their quota or risk losing it. For this group of fishers, fishing was not driven by profit maximization, but rather by a different set of norms. Their fishing effort varied with their needs, with young fishers who had more debts to repay fishing more, and older fishers with less pressing needs for money fishing less and dedicating their remaining time to activities on shore. Their capacity for fishing, as indicated by the size and equipment of their vessels, did not correlate with fishing effort either, but rather with the fishers desire to increase their comfort while at sea or to gain status among their peers.

While not as dramatic as some of the bigger policy failures in SES (or even just in fisheries), this is a good example of overall sound policy failing to account for the needs and behaviors of at least some of the people it set out to manage. In this chapter we set out to show how including different drivers of behavior in models can change the outcomes and thus point towards different possible policy approaches.

The model scenarios in this chapter are nothing more than toy models, spun up to illustrate the functionality of the architecture and the sandbox approach we present in this thesis. We pick up elements from the Norwegian fisheries scenario (such as the norm of correlating fishing effort to personal needs), but others, like the value trees, are mostly educated guess work. As such, these models do not capture any real world scenario, and only illustrate that changes in the decision making process of agents using our decision architecture is enough to ensure significantly different results, even when agents are fairly similar in their capabilities and motivations. Since our focus is on the interaction between norms, values and policy, the models in this chapter include minimal environmental dynamics, so that the differences in agent behavior can be easily traced to differences in agent motivations and reasoning processes.

6.1 Sandbox setup

In this section, we will present all the elements we put in the sandbox in order to build the model scenarios. We start with the sandbox and not model scenarios because the scenarios are all built by combining elements from the sandbox and the scenarios presented in this chapter are not the only ones that can conceivably be built just from this sandbox alone. Even with the limited ambition of simply demonstrating sandbox and architecture functionality rather than building a full, complex model meant to serve a role in a policy design process, the resulting sandbox and the opportunities for building and experimenting with different types of agents are fairly substantial.

Most of the sandbox elements concern agent makeup since the goal of this work is to showcase an agent decision architecture that allows agents to reason about norms and values, among other cognitive elements. The following sections detail these cognitive elements, the reasoners associated with them, and the combinations we use to make up the four different types of agents we use in our simulation scenarios. Again, because the architecture conceptualises these cognitive elements as explicit - as opposed to implicit rules or constraints hardcoded into the flow of the program - the agents are very modular. As such, the four agent types we use in our scenarios are not the only ones we could have built using the elements described in this chapter, but they are the ones that best fit the case study we are basing the model on.

6.1.1 Agent makeup

All the agents in this model are fishers, split into two categories: fishers who aim to maximize their profit, and fishers who follow a social norm that limits their fishing effort. This limitation is determined by the size of their debts, which include vessel maintenance, living costs and loan repayment rates. Once their fishing profit and/or savings are enough to cover their debts for the rest of the year, they switch from fishing to spending time on shore. Profit maximizing fishers, on the other hand, fish as much as they can and do not willingly limit their fishing effort.

Agents share much of their makeup across types and scenarios, since they live in and perceive the same environment, have shared motivations and behave similarly for the most part. We distinguish between types of agents based on whether their makeup contains norms and/or a value system, which gives us "rational" goal-oriented agents, normative goal oriented agents, self-interested value driven agents, and community-oriented value driven agents. In this section, we will break down the cognitive elements the agents use in these simulations. The particular makeup of agent types is described further into the chapter.

6.1.1.1 Entities

In all scenarios, agents have the same list of entities (see table 6.1.1.1, and table 6.1.1.1 for their components). They are all present at the start of any simulation, except for the **Quota** entity, which is introduced to the agents by the **Policy** object if the scenario includes a switch of policy. The **Policy** object also removes the **HasLicence** component from all **Vessel** entities, which makes **Vessel** the only entity in the model to change structure during certain scenario runs.

Component	Type
Cost	Float
MaintenanceCost	Float
Size	Float
Efficiency	Float
Experience	Float
Age	Float
Rate	Float
HasLicence	Bool
Place	String

Table 6.1: The components and their types used in the models

Entity	Components
Vessel	Size
	Cost
	Efficiency
	MaintenanceCost
	HasLicence
Catch	Size
	Cost
Loan	Size
	Rate
Debts	Size
Profit	Size
Savings	Size
LivingCosts	Size
Place	Place
Quota	Size

Table 6.2: The entities, and their components, used in the models

6.1.1.2 Actions

The actions available in the models are in table 6.3, together with a description of their precondition, transformation and effect functions. Actions that are either norms or policies are also included, but are marked as such, and their context activation function is also described. Actions that are also policies are added to each agent at runtime in scenarios where policy becomes active during the run.

Actions	Description
Fish	**Pre-condition** agent has a vessel with a valid license

Table 6.3 – continued from previous page

Actions		Description
	Transformation	agent requests catch from Resource object
	Effect	a catch is calculated, proportional to vessel efficiency and size
GetLoan	**Pre-condition**	agent has no other loans, debts exceed threshold and savings <= 0
	Transformation	agent gets loan
	Effect	agent's loan size >0
SellFish	**Pre-condition**	agent's catch size >0
	Transformation	agent's catch size set to 0, value of catch added to agent's profit
	Effect	agent's catch size set to 0, agent's profit size increased by value of catch
PayDebts	**Pre-condition**	the sum of agent's savings and profit exceeds its debt
	Transformation	debts get paid from profit first, then from savings if debts exceed profit
	Effect	savings modified by difference between profit and debts, debts set to 0
UpgradeVesselEfficiency	**Pre-condition**	agent has vessel and agent's savings >cost of upgrade
	Transformation	vessel efficiency increased, savings decreased by cost of upgrade
	Effect	vessel efficiency increased, savings decreased by cost of upgrade
UpgradeVesselSize	**Pre-condition**	agent has vessel and agent's savings >cost of upgrade
	Transformation	vessel size increased, savings decreased by cost of upgrade

Table 6.3 – continued from previous page

Actions		Description
	Effect	vessel size increased, savings decreased by cost of upgrade
Norms		
SpendTimeOnShore	**Context**	savings >yearly debts
	Pre-condition	savings >yearly debts
	Transformation	agent’s location is "on shore"
	Effect	agent’s location is "on shore"
Policies		
FishQuota	**Context**	agent has vessel
	Pre-condition	agent’s quota >0
	Transformation	agent requests catch from Resource object
	Effect	a catch is calculated, proportional to vessel efficiency and size
SellQuota	**Context**	quota size >0 at end of year
	Pre-condition	quota size >0 at end of year
	Transformation	QuotaMarket receives all quota from agent, agent’s quota is set to 0
	Effect	agent’s quota is set to 0
BuyQuota	**Context**	quota size <= 0 at end of year, savings >0
	Pre-condition	quota size <= 0 at end of year, savings >0
	Transformation	agent requests quota from QuotaMarket object, proportional to vessel’s unfulfilled fishing potential
	Effect	a quota increase is calculated based on vessel’s unfulfilled fishing potential

Table 6.3: All the actions used by the agents in the models

6.1.1.3 Goals

The goals the agents can have across the different scenarios are in table 6.4, together with a description of their achievement function. Goals that are either norms or policies are marked as such and also include a description of their context activation function. As before, goals that are policy are not part of an agent's makeup until introduced by the **Policy** object when the running scenario decides the policy is in effect.

Goals		Description
DoJob	**Conditon**	Fish action has been performed or catch >0
GetPaid	**Conditon**	Profit >0
PayCosts	**Conditon**	Debts = 0
StaySolvent	**Conditon**	Savings >0
Upgrade	**Conditon**	Not (end of year and loan >0)
Norms		
DoJobNormative	**Conditon**	yearly debts not paid and (fish action has been performed or catch >0); or yearly debts paid and location = "on shore"
Policy		
DoJobPolicy	**Conditon**	Fish action has been performed and quota >= 0

Table 6.4: All the goals available to the agents in the models

6.1.1.4 Value systems

There are two value systems the agents use and they are built to reflect the motivations of the "rational" agent type and the "normative" agent type from the goal-oriented decision making scenarios. In order to distinguish between them, we call the "rational" equivalent a "self-interested" agent type, and the "normative" equivalent a "community-oriented" agent type.

The value systems are presented in tables 6.5 and 6.6, together with descriptions of the preference functions of all their nodes.

Root name	Root preference	Intermediary preference	Leaf preference	Entity
Achievement	OR	OR	MoreThanZero	Catch
			MoreThanZero	Profit
		OR	BetterThan	Savings
			IsPaid	Debts
Benevolence	OR		Savings > YearlyDebts	Savings
				Debts
			InFishery	Place
Conformity	XNOR		Quota >= Catch	Quota
				Catch
			InFishery	Place
Power	OR		BetterEfficiency	Vessel
			BetterThan	Vessel
			BetterThan	Quota
		Order		
Achievement	**Power**	**Conformity**	**Benevolence**	

Table 6.5: Self-interested value system

6.1.1.5 Reasoners

The agents use two kinds of reasoners, one which knows how to deal with goals, norms and policy, and a second one which knows how to deal with values, norms and policy. The reasoners are distinguished through the kinds of motivators they recognize and operate with, and not the actual motivators themselves. Thus, goal-oriented agents use the same reasoner, regardless of what goals they have in their makeup. Value-driven agents also use the same reasoner, regardless of which value system they have as motivator. This means that any behavioral differences observed between agents of the same type (goal-oriented or value-driven) are not due to their reasoning process, which is the same, but are due to their motivators and capabilities.

We have already described these reasoners in chapters 4 and 5, but we present them again here for convenience, in fig 6.1 and 6.2.

6.1.2 Agent types

For the purposes of this chapter, we classify agents according to two main criteria. First, agent types based on the kinds of motivators drive their behavior. As such, we have two main types of agents: goal-oriented and value-driven. Goal-oriented agents try to fulfill their goals, while

Root name	Root preference	Intermediary preference	Leaf preference	Entity
Achievement	OR	OR	NotZero	Catch
			NotZero	Profit
		OR	BetterThan	Savings
			IsPaid	Debts
Benevolence	XNOR		Savings > YearlyDebts	Savings
				Debts
			IsOnShore	Place
Conformity	OR	XNOR	IsOnShore	Place
			Savings > YearlyDebts	Savings
				Debts
			Quota >= Catch	Quota
				Catch
Power	OR		BetterEfficiency	Vessel
			BetterThan	Vessel
			BetterThan	Quota
		Order		
Benevolence	**Conformity**	**Achievement**	**Power**	

Table 6.6: Community-oriented value system

value-driven agents try to maximize the fulfillment of their value system. Both these types can, in addition, have norms and/or policies as motivators, in the form of actions and/or goals which have an additional context activation function.

Second, we also distinguish between agents based on their "personality". In the case of the goal-oriented type, we have "rational" agents, who adhere to no norms and their goals are meant to maximize their profit, and "normative" agents, who follow the norm of fishing enough to cover their costs and spend whatever time they have left on shore. Their value-driven equivalents are called "self-interested" and "community-oriented", respectively, in order to distinguish between agents with values and agents with goals.

The types of agents we use in our scenarios can be found in table 6.7. The table shows all the elements any of the agent types can contain across scenarios. It is important to note that none of the agents contain both goals and values at the same time in any of the scenarios in this chapter. The specific agent makeups present in each scenario will be described further in the chapter.

```
if goalQueue is empty
    add all goals to goalQueue

forall (goals in goalQueue)
    if (goal is not fulfilled)
        if (goal is normGoal)
            if (goal is active)
                add goal to normGoalList

        else if (goal is policyGoal)
            if (goal is active)
                add goal to policyGoalList

        else add goal to goalList

if (normGoalList is not empty)
    nextGoal is random goal from normGoalList
else if (policyGoalList is not empty)
    nextGoal is random goal from policyGoalList
else
    nextGoal is random goal from goalList

forall (actions in availableActionsList)
    if action fulfills nextGoal
        if (action is normAction)
            if (action is active)
                add action to normActionList
        else if (action is policyAction)
                if (action is active)
                    add action to policyActionList
        else add action to actionList

if (normActionList is not empty)
    nextAction is random action from normActionList
else if (policyActionList is not empty)
    nextAction is random action from policyActionList
else
    nextAction is random action from actionList

execute nextAction
\label{code_basereasoner}
```

Code 6.1: Reasoning loop for base reasoner

6.2 Fishing grounds

The fishing grounds object keeps track of how much resource there is in the simulation at any given time and how much of it the agents can harvest. The fish "reproduces" once every simulation year and the resulting amount is determined by a logistic growth function. In scenarios

```
update value tree satisfaction
forall (actions in availableActions)
    if (action is normAction)
        if (action is active)
            add action to normActionList
    else if (action is policyAction)
        if (action is active)
            add action to actionList
    else add action to actionList

if (normActionList is not empty)
    nextAct = max(valueSatisfaction(action in normActionList))
    execute nextAct
    break

else
    nextAct = max(valueSatisfaction(action in actionList))
    execute nextAct
```

Code 6.2: Reasoning loop for value reasoner

where the agents fish a "finite" resource, the carry capacity and growth rate are calibrated such that if the scenario contained only "rational" agents, they would crash the resource by the end of the simulation's time scope.

6.3 Simulation setup

Each simulation contains 200 agents, regardless of makeup, and runs for 20 years, unless otherwise specified. Agents start out with varying loan size, vessel size (in the same size-class), and efficiency. They differ in the kinds of actions they have available to them, and different agent types are motivated differently, either by goals, norms, policy, values, or a combination thereof. The agents use two types of basic reasoning: goal-oriented or value-based. Both reasoner types can also deal with norms and policy.

No agents exit or enter the fishery. Exiting and entering a fishery come with their own issues and can factor prominently in the policy design process as there can be huge differences in the ways a policy can impact these actions. Buyouts by the state can encourage less prosperous fishers to exit the fishery and pursue a different occupation. Quota policies come with significant barriers to entry because quota is usually very expensive and not many aspiring fishers can afford the investment. However, regardless of how interesting it would be to simulate this particular aspect of fisher behavior, the scenarios in this chapter aim at showcasing the functionality of the architecture and of the sandbox setup, not to replicate all elements of interest to

Rational Agent	Normative Agent
Actions	
Fish	Fish
GetLoan	GetLoan
SellFish	SellFish
PayDebts	PayDebts
UpgradeVesselEfficiency	UpgradeVesselEfficiency
UpgradeVesselSize	UpgradeVesselSize
	SpendTimeOnShore
BuyQuota	BuyQuota
SellQuota	SellQuota
FishQuota	FishQuota
Goals	
DoJob	DoJobNormative
GetPaid	GetPaid
PayCosts	PayCosts
StaySolvent	StaySolvent
Upgrade	UpgradeNormative
DoJobPolicy	DoJobPolicy
Value system	
RationalValueSystem	NormativeValueSystem

Table 6.7: The two complete agent profiles used in the models. Blue highlighted elements are added by the policy when it is introduced into the simulation. Yellow highlighted elements are only present in scenarios involving values.

a model meant to inform or support a real life policy design case. As such, keeping the model simple can be more informative since there is less chance of confusion when interpreting the results.

There is one fishing grounds object, which determines how much fish is available for harvesting, as well as how much fish the agents get based on their vessel size, vessel efficiency and their own experience in fishing. The fish "reproduce" once a year, according to a logistic function. In scenarios with a "finite" amount of fish, the resource's carry capacity and growth rate are calibrated such that a population of 200 "rational" agents will crash it before the simulation ends, but not sooner than its halfway point.

The time granularity of the simulations is one month, split into 5 steps. This is because the agents are allowed to perform a number of actions each month: catch fish, sell fish, pay their debts for the month, take out a new loan, and upgrade their vessel or buy/sell quota if the policy is in effect. They can spend time on shore instead of fishing, but cannot do both in the same month. Also, upgrading a vessel or trading quota is only allowed once a year.

The simulations vary in the kinds of agents they run, whether they can crash the resource, and whether a policy is introduced. If the scenario run includes a policy, the policy is always the same ITQ policy and is introduced midway through the run. Once introduced, the policy allocates quota to the agents based on the last few years of their collective fishing history.

6.4 Scenarios

The scenarios are very simple, and are mainly used to observe how behavior varies between different kinds of agents. We are also interested in the explanatory power the scenarios get from the kinds of agents included.

We build and run two main kinds of scenarios: ones with goal-oriented agents and ones with value-driven agents. We use the goal-oriented scenarios to demonstrate the use of the sandbox setup, starting with the simplest scenarios we could design and gradually adding elements to increase their informative power.

6.4.1 Scenario 1: Goal-driven Agents - Rational vs. Normative

This scenario contains goal-driven agents, both "rational" and "normative". Their respective makeups can be found in table 6.8.

In order to show the differences in behavior between the two types of agents, we run a number of variations on this scenario: scenarios with only one type of agent, scenario with both types of agent, scenario with both types of agents and policy. Each variation runs both under conditions of limited and unlimited resource.

6.4.1.1 Scenario 1.1: Single agent type

These are the simplest scenarios we built, and, as the name implies, this variation is run with only one type of agent. It’s run with both limited and unlimited resource, but no policy. The purpose of this setup is to have a good look at how each agent type behaves when the only constraints on their behavior is internal (in the case of infinitely abundant resource), and when we add the simplest external constraint (in the case of finite resource).

Rational Agent	Normative Agent
Actions	
Fish	Fish
GetLoan	GetLoan
SellFish	SellFish
PayDebts	PayDebts
UpgradeVesselEfficiency	UpgradeVesselEfficiency
UpgradeVesselSize	UpgradeVesselSize
	SpendTimeOnShore
BuyQuota	BuyQuota
SellQuota	SellQuota
FishQuota	FishQuota
Goals	
DoJob	DoJobNormative
GetPaid	GetPaid
PayCosts	PayCosts
StaySolvent	StaySolvent
Upgrade	UpgradeNormative
DoJobPolicy	DoJobPolicy
Value system	
None	None

Table 6.8: The two complete agent profiles used in the scenarios goal-oriented agent. The differences between them are highlighted in green. Blue highlighted elements are added by the policy when it is introduced into the simulation.

As shown in 6.1, When they are given an infinitely abundant resource, the "rational" agents go fishing every chance they get, while normative agents gradually decrease their fishing effort over time. This can also be seen in fig. 6.10: the "rational" curve is a straight line indicating a constant increase in total cumulative catch over the years, while the normative curve bends downwards marking a decrease in the total cumulative catch, and later continues as a straight line indicating a constant increase in total cumulative catch again. This second period of stable fishing effort marks the point where agents have paid off their loans and are now fishing to cover their maintenance and living expenses, which vary little.

When constrained by dwindling resource abundance, "rational" agents also decrease their fishing effort, albeit they are much slower to do it. Unlike normative agents, who decrease their effort because of their social norm, "rational" agents only reduce their effort when it becomes unprofitable to keep going out to sea (see the "wavy" curve in fig. 6.4). They deplete the resource fast enough that they cannot earn enough to upgrade their vessels at all, whereas they can afford significantly more upgrades (even when compared to normative agents) when they can catch as much fish as they want (see fig. 6.7 vs. none in the finite condition).

Normative agents voluntarily curb their fishing effort when the resource is finite too, as seen in figure 6.4.1.1. However, whether the resource is finite or not, the reduction isn't necessarily strict from year to year. If they have access to an unlimited resource, normative agents can earn enough money fast enough to repay their debts fairly quickly and, thus, get the opportunity to upgrade their vessels (see fig 6.7). This means they either deplete their savings or take out a new loan (see fig. 6.5), which forces them to temporarily increase their fishing effort in order to cover their yearly costs. On the other hand, if the resource is being noticeably depleted, they do not earn enough to afford to upgrade their vessels (as such, no figure is shown). Their increase in fishing effort is due to the fact that they do not catch as much each fishing trip and are, thus, forced to spend less time on shore and more time in the fishery in order to cover their costs for the year.

6.4.2 Scenario 1.2: Goal-driven Agents - Rational-Normative mix

This scenario includes an equal number of "rational" and normative agents (100 of each) in order to compare the two when they share the same resource, whether finite or infinite.

As in the previous version of this scenario, "rational" agents fish either to their maximum capacity or as long as fishing is profitable, whereas normative agents gradually reduce their fishing effort commensurate with their costs, and spend as much time on shore as they can afford (see fig. 6.8 and fig. 6.9 for action counts, and fig. 6.10 and fig. 6.11 for cumulative catch). The loans they take out follow the same approximate trend (see fig. 6.12 and fig. 6.6), as do the upgrades they buy (see fig. 6.14 vs. none in the finite condition).

This scenario doesn't add a whole lot of new information regarding the behavior of the agents, but it does illustrate how significant the difference in behavior between the two agent types really is. This difference will become extremely relevant in the next scenario variation when we introduce policy.

6.4.3 Scenario 1.3: Goal-oriented, mixed populations and policy

In this scenario variation we run the same mix of agents as in the previous variation (100 "rational", 100 normative), and we add an ITQ policy halfway through the simulation. The policy calculates the total allowable catch (TAC) at the beginning of each year and distributes it to the agents. Each agent receives an equal share of the TAC, regardless of its makeup, catch or

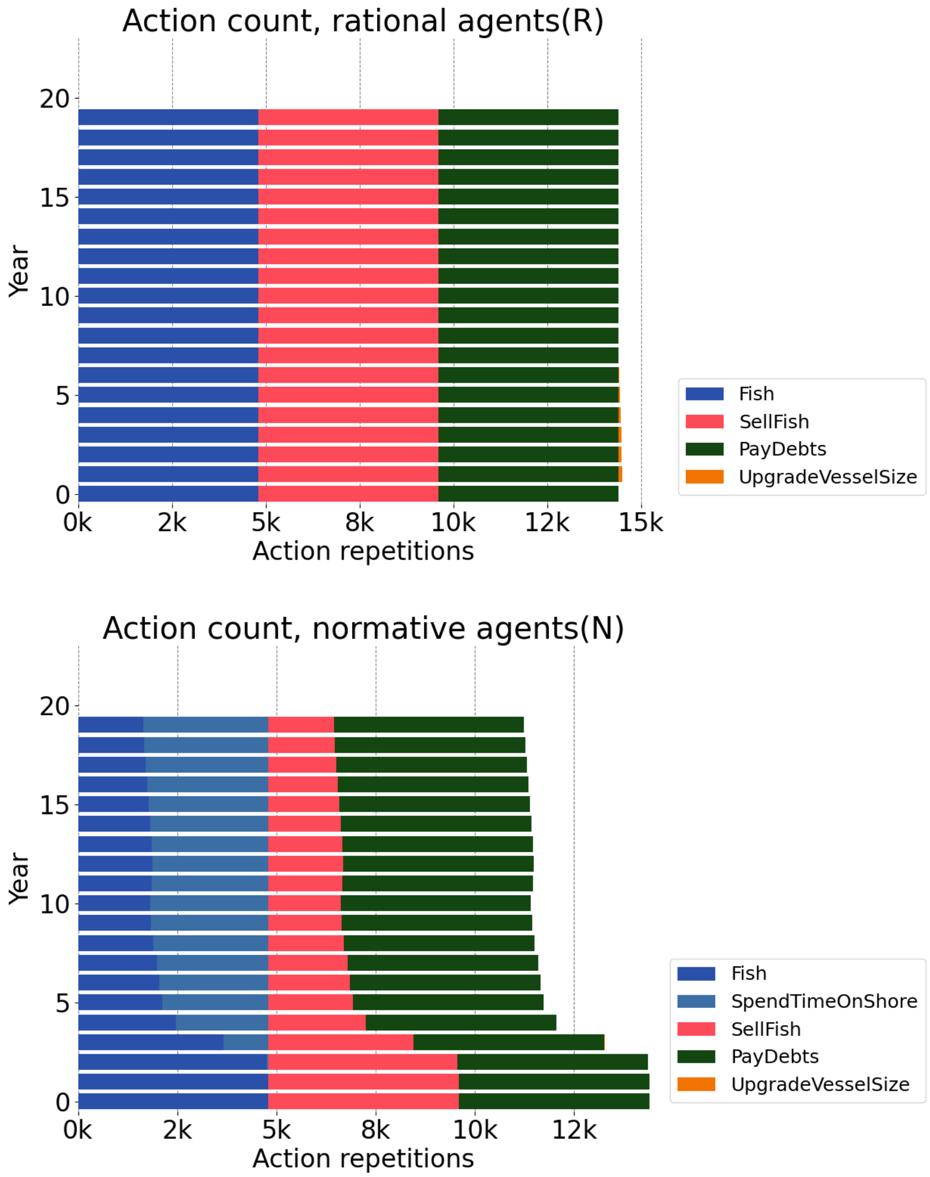


Fig. 6.1: Action counts for simulation runs where the population contains only one kind of agent fishing an infinite resource. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

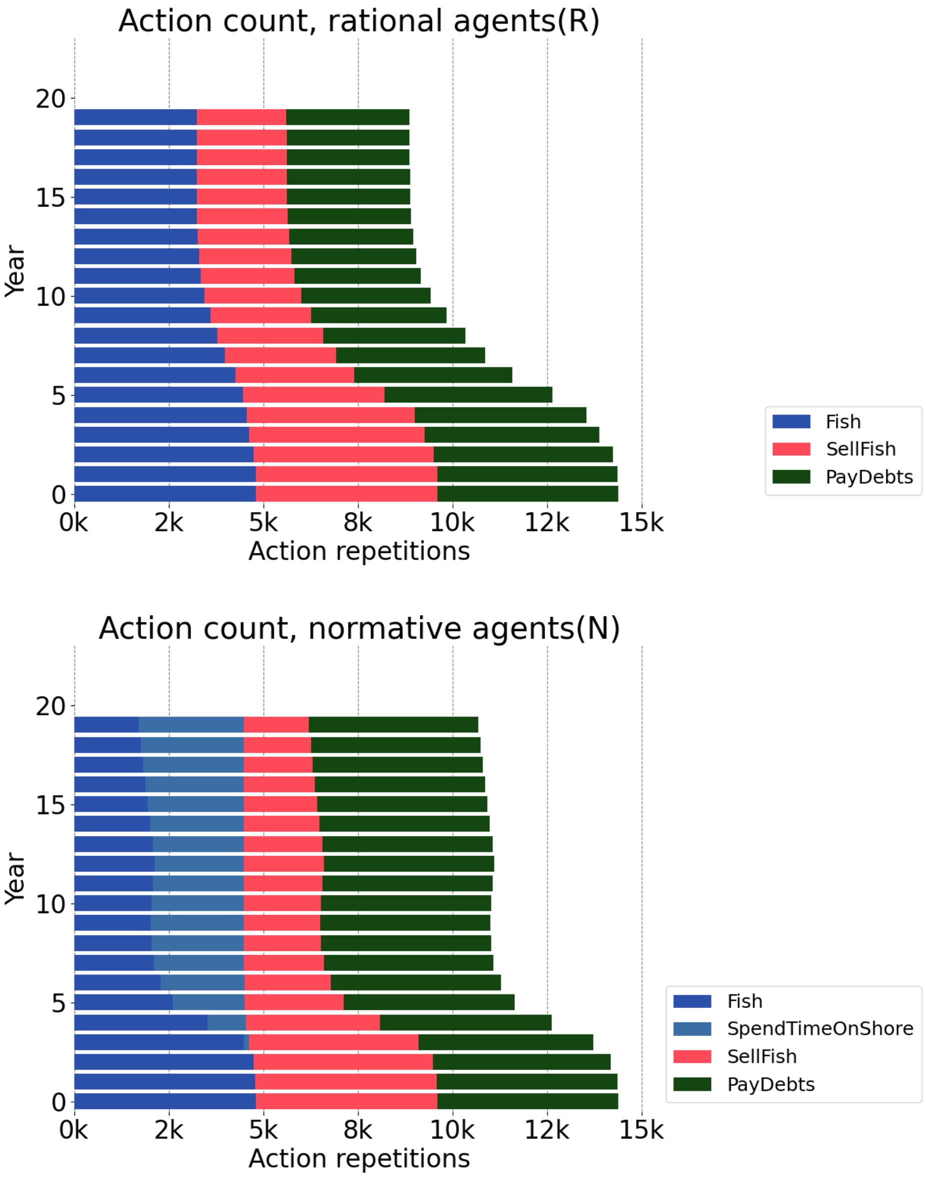


Fig. 6.2: Action counts for simulation runs where the population contains only one kind of agent fishing a finite resource. Rational agents (up) fish as much as they can, as long as it's profitable, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

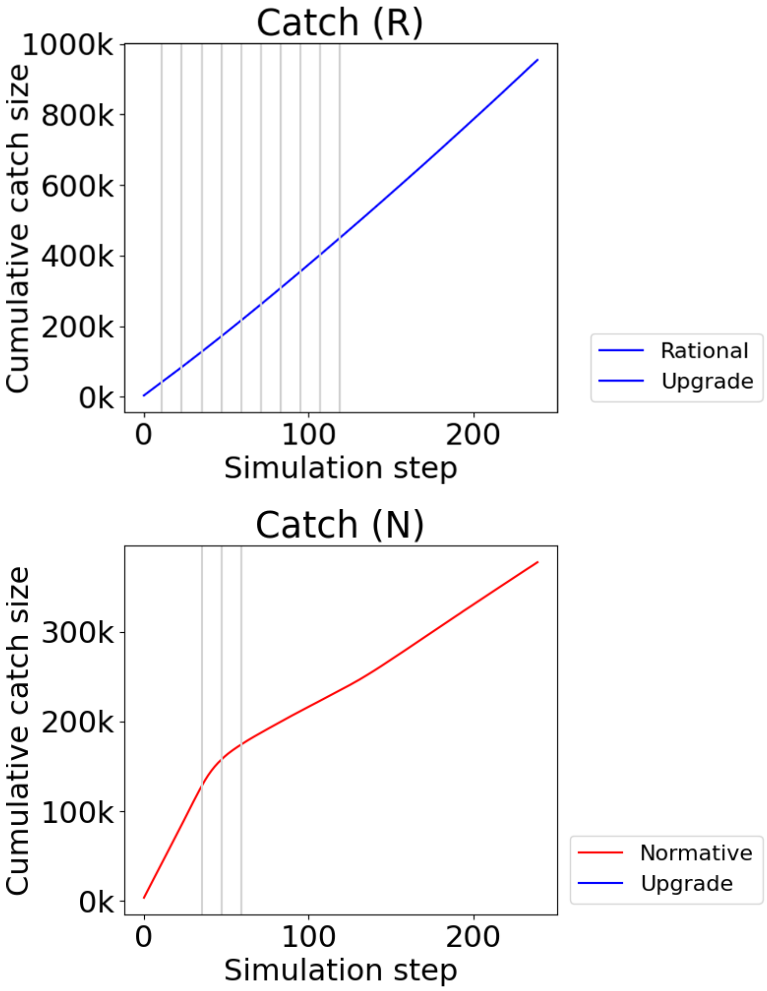


Fig. 6.3: Cumulative catch for simulation runs where the population contains only one kind of agent fishing an infinite resource. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

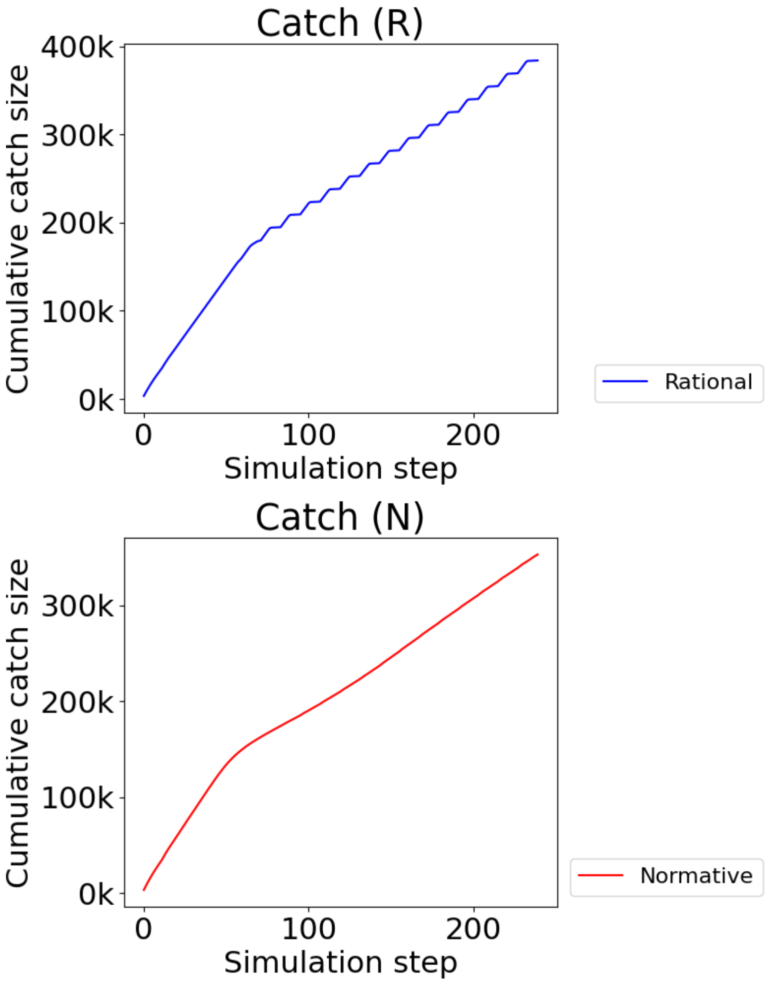


Fig. 6.4: Cumulative catch for simulation runs where the population contains only one kind of agent fishing a finite resource. Rational agents (up) fish as much as they can, as long as it's profitable, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

effort history. If the resource is finite, the TAC is calculated as the maximum amount of fish that can be harvested such that the resource can rebound fully in the next year's reproduction step. If the resource is infinite, the TAC is calculated as a percentage of the average historical yearly catch of all the agents in order to have the same restrictive effects.

What we want to observe in this scenario variation is whether the policy may seem to achieve its overall goal, but unintentionally disrupt the behaviors of some agents, given that it is set

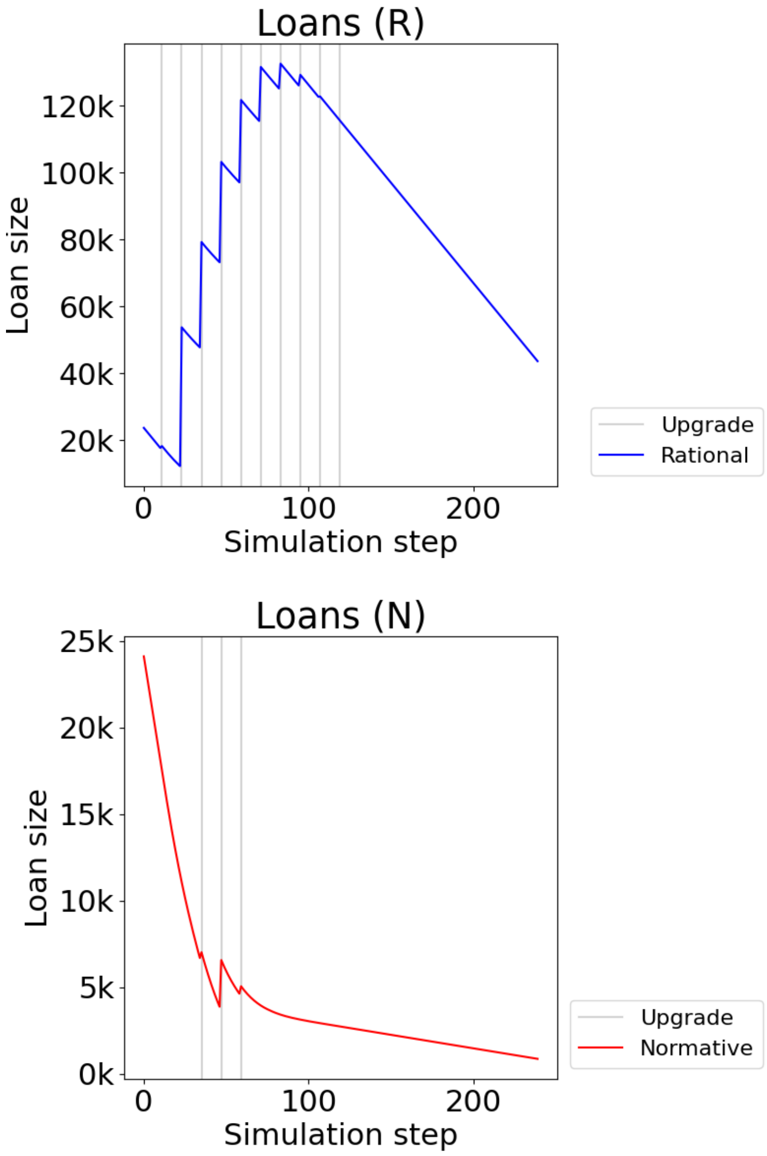


Fig. 6.5: Average loan for simulation runs where the population contains only one kind of agent fishing an infinite resource. Rational agents (up) often take new loans in order to afford vessel upgrades, whereas normative agents upgrade and take out new loans much less infrequently. At the end of the run, the normative population has nearly finished repaying their loans, while the rational population collectively has bigger loans than they started with.

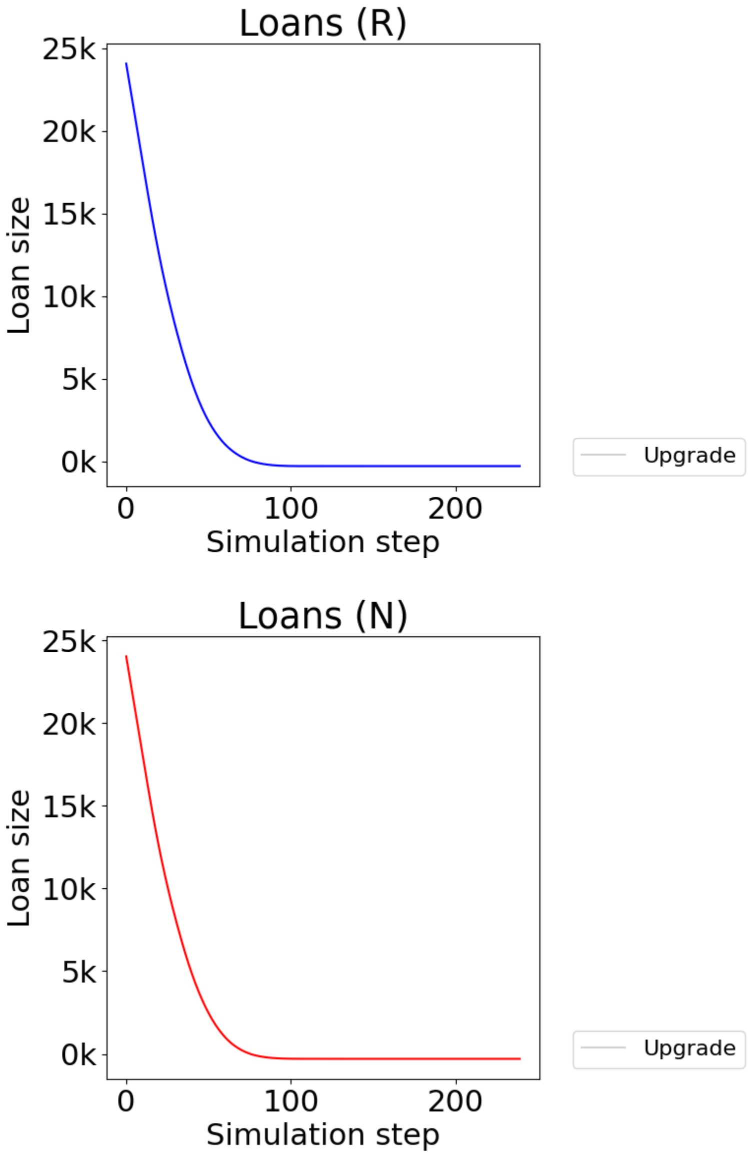


Fig. 6.6: Average loan for simulation runs where the population contains only one kind of agent fishing a finite resource. Rational agents (up) take no new loans because fishing isn't as profitable under these conditions. It is the same for normative agents.

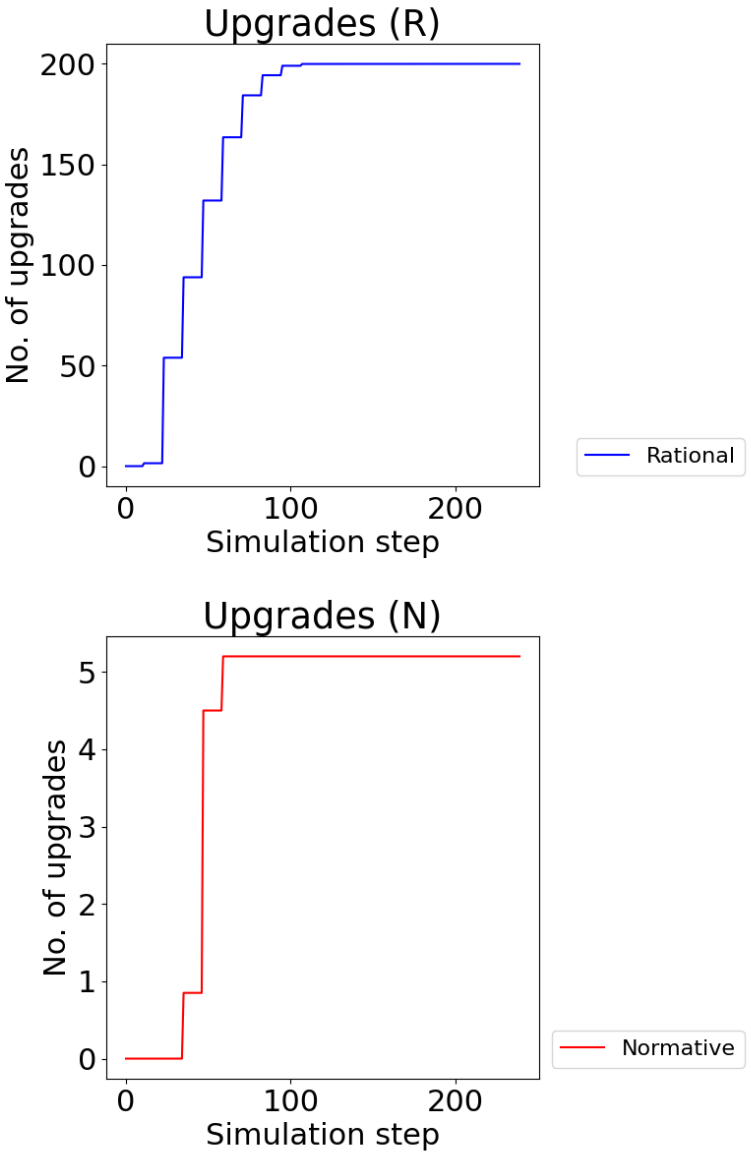
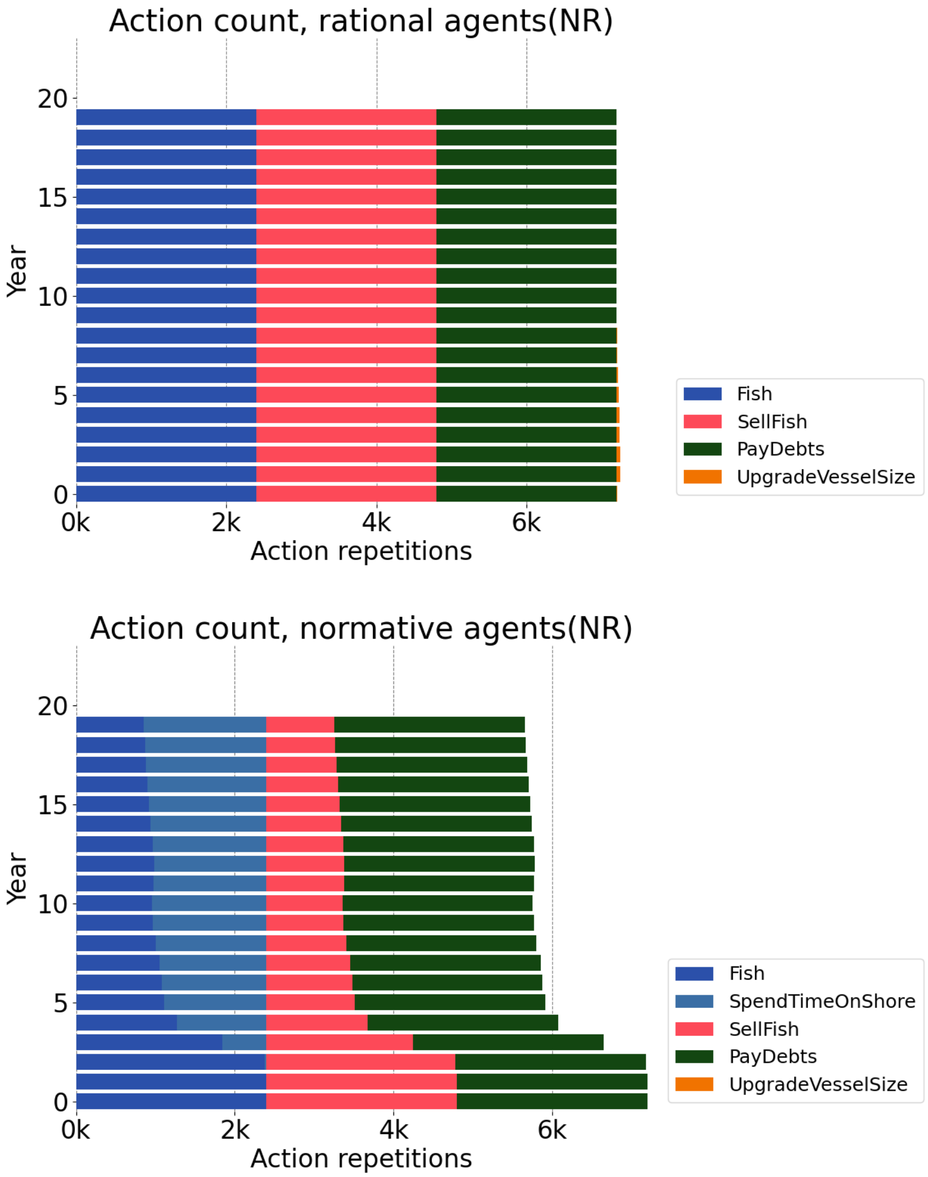


Fig. 6.7: Number of upgrades at any given simulation step for simulation runs where the population contains only one kind of agent fishing an infinite resource. Rational agents (up) upgrade their vessel often because they can afford to repay their loans. While normative agents can also afford to repay their loans, they upgrade much less frequently, since they prefer to spend time on shore instead.


Action count, rational agents(NR)

Action count, normative agents(NR)

Fig. 6.8: Actions counts for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

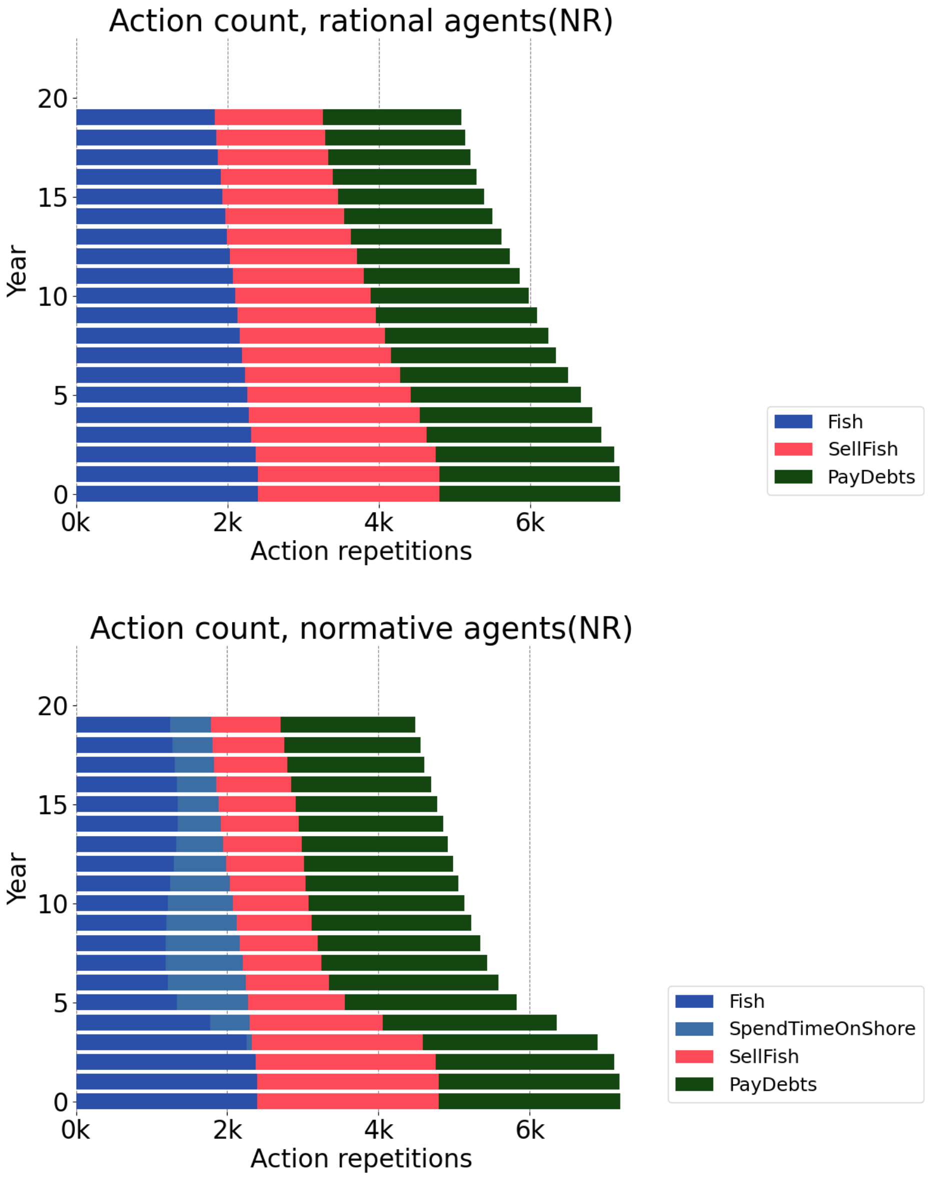


Fig. 6.9: Actions counts for simulation runs where the population contains both rational and normative agents fishing finite resource. Rational agents (up) fish slightly reduce their effort over time as fishing becomes less and less profitable, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

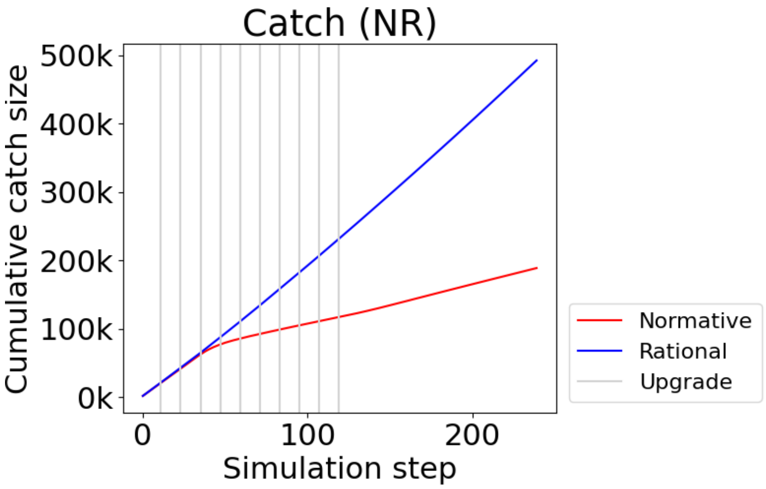


Fig. 6.10: Cumulative catch for simulation runs where the population contains both rational and normative agents. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

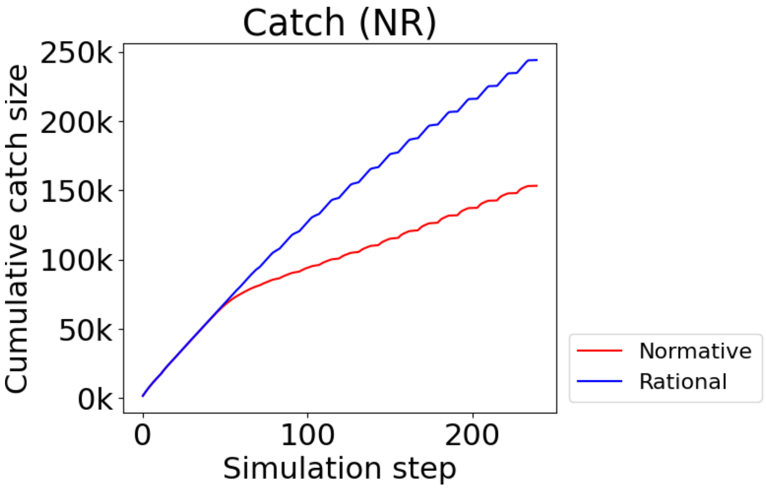


Fig. 6.11: Cumulative catch for simulation runs where the population contains both rational and normative agents. They still diverge around the same cumulative catch value, but it occurs later in the simulation because the resource is more scarce and the normative agents need to fish more in order to save enough money to switch to spending time on shore.

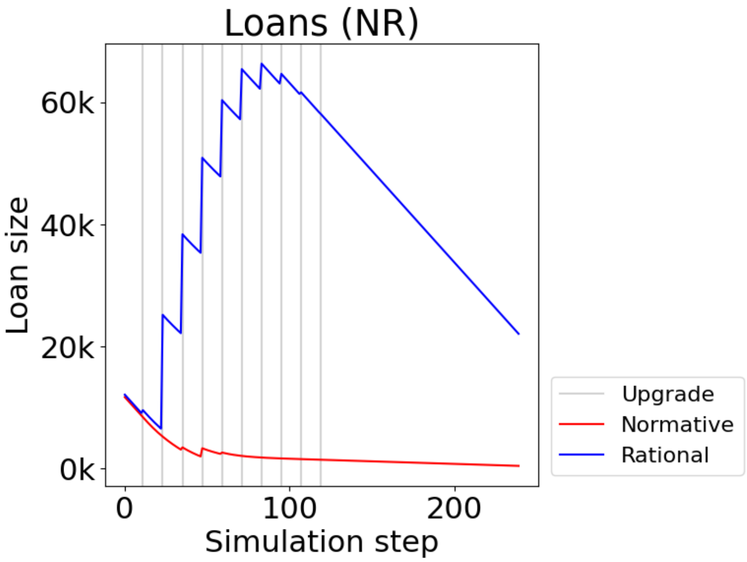


Fig. 6.12: Average loans for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents take on significantly more debt than normative agents, both because their larger fishing effort means they can afford to pay back more, and because normative agents would rather spend time on shore than repay additional debt.

based on historical fishing effort in a given tonnage class, and that class contains agents with different behaviors and incentives. In order to observe whether the policy eventually restores the resource, we run simulations lasting 40 years instead of the standard 20, since 20 years is not quite enough to see this effect. The policy is introduced midway through the simulation, which means the agents have plenty of time to deplete the fish stocks and are well on their way to completely collapsing them. Under these circumstances, the policy is very restrictive when it is first introduced, but it becomes less so as the resource, hopefully, recovers.

What the results show is that the policy has the overall desired effect of cutting down fishing effort and lowering overall catch. However, these effects are not the same for the two types of agents. "Rational" agents are, as expected, forced to cut down on their fishing effort (see fig. 6.16 and 6.15) and their total catch is lowered as well (see fig. 6.18 and fig. 6.17. However, the same figures show that normative agents increase their effort and their total catch. Because the quota is calculated by taking the average historical catch of the entire population of agents, normative agents are allotted more quota than they would fish in the absence of the policy. Where before their norm compelled them to gradually reduce their fishing effort over time, now they keep fishing past where their gains would have covered their expenses for the year.

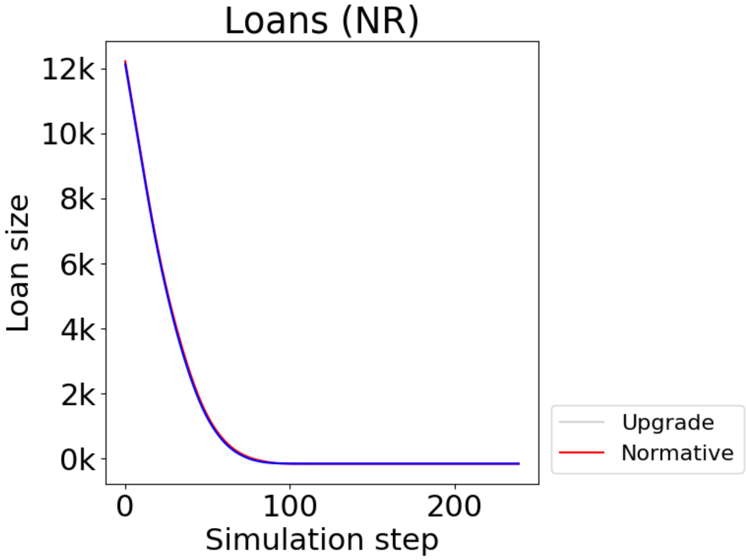


Fig. 6.13: Average loans for simulation runs where the population contains both rational and normative agents fishing a finite resource. Due to the resource becoming more scarce and less profitable as the simulation advances, none of the agents have the means to take on extra loans and limit themselves to repaying the ones they started with.

Regarding upgrades, normative agents do not change their behavior from previous runs. On the other hand, when the resource is infinite (meaning the policy is only mildly restrictive), "rational" agents appear to stop upgrading once the policy is introduced. However, in the 40 year runs, it becomes apparent that the introduction of policy merely slows down the pace at which they can accumulate the money they need and the pace of upgrades picks up again eventually (see fig. 6.7). What's more interesting, at first glance anyway, is that, when the resource is finite, "rational" agents cannot afford upgrades until the policy is introduced (see fig. 6.22). This apparently deviant behavior is easily explained by the fact that buying and selling quota are counted as upgrades too, and the action count charts show agents buying quota.

This also tracks with the loans we observe in these runs. When the resource is infinite, "rational" agents take out loans to finance their upgrades until the policy is introduced, then they appear to stop. However, the 40 year runs show that this is a temporary situation due to the fact that the rate at which they can accumulate wealth has slowed (see fig. 6.19 and fig. 6.20). Interestingly, once the start loaning money again, they do not loan as much as before the policy was in effect. This is because they now buy and sell quota, and they can do it in increments that are much cheaper than a vessel upgrade (at least according to the current balance of the sandbox).

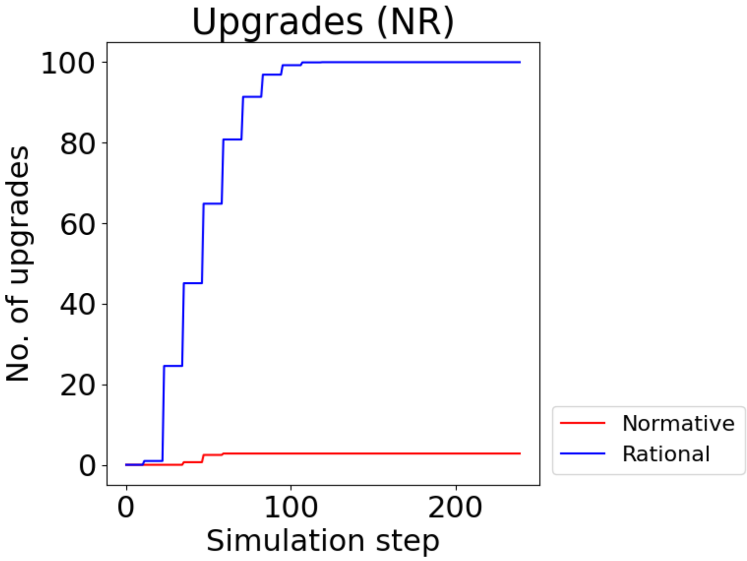


Fig. 6.14: Number of upgrades at any given simulation step for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents are much more likely to take on extra loans to buy vessel upgrades, while normative agents prefer spending time on shore to taking on more debt in order to afford more vessel upgrades.

It bears mentioning that, even though the policy increases the fishing effort for normative agents, the resource still recovers in time. This is because quota shares are calculated with regards to the TAC, which is calculated with the preservation of the resource in mind. As such, even if this isn't the best quota distribution scheme when it comes to resource recovery, it still achieves its purpose.

6.4.4 Scenario 2: Agents with values

One crucial element missing from the previous scenarios is the ability of agents to break norms or policy. Because the goal-based reasoner forces norms over non-normative actions/goals and policy actions/goals over norms, agents always follow their norm if it is active, and always obey the policy once it comes in effect. However, in order to understand the effects of any given policy, we need to be able to see the ways and reasons agents may choose to break it, and the goal reasoner cannot decide between a norm and a policy that are active at the same, or decide to ignore an active norm/policy if some other action would be more beneficial.

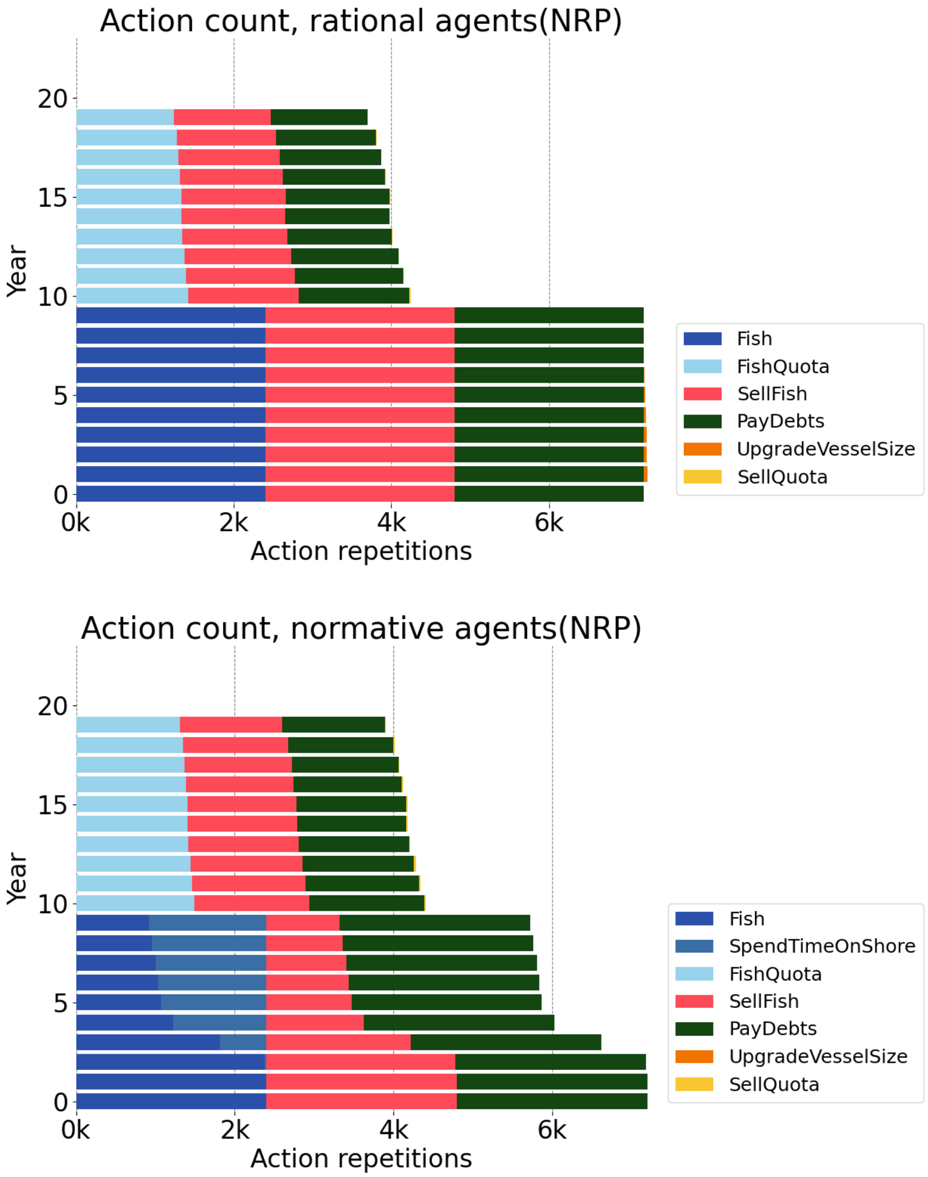


Fig. 6.15: Actions counts for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

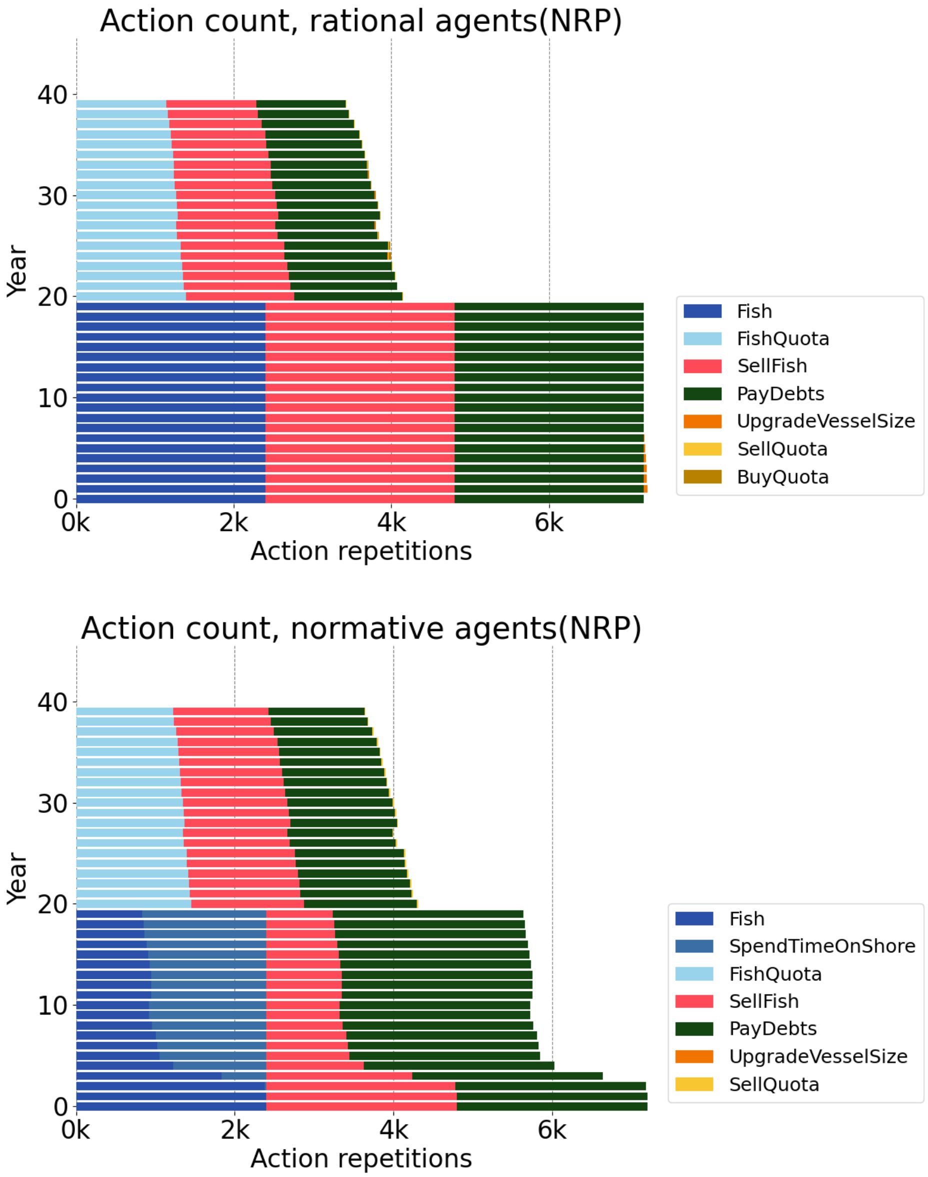


Fig. 6.16: Actions counts for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

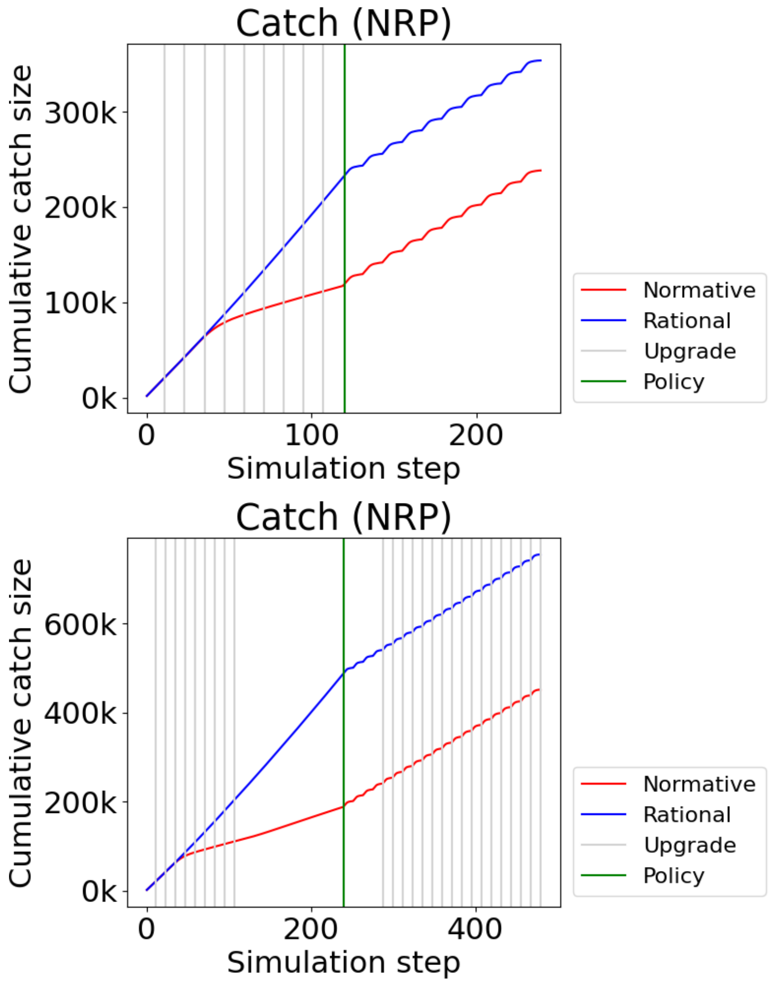


Fig. 6.17: Cumulative catch for simulation runs where the population contains both rational and normative agents. Rational agents (up) fish as much as they can, whereas normative agents gradually reduce their fishing activity over time and spend more time on shore instead.

In order for the agents to be able to resolve norm-policy or action-norm/policy conflicts, they need another system for prioritizing their behavioral options and this is where values come in.

In this scenario, the agents do not have goals, but are driven by a value system. Agents use the value reasoner to choose between their available options, which allows them to move beyond the hard rule of always choosing active policy over active norms that the goal reasoner imposes.

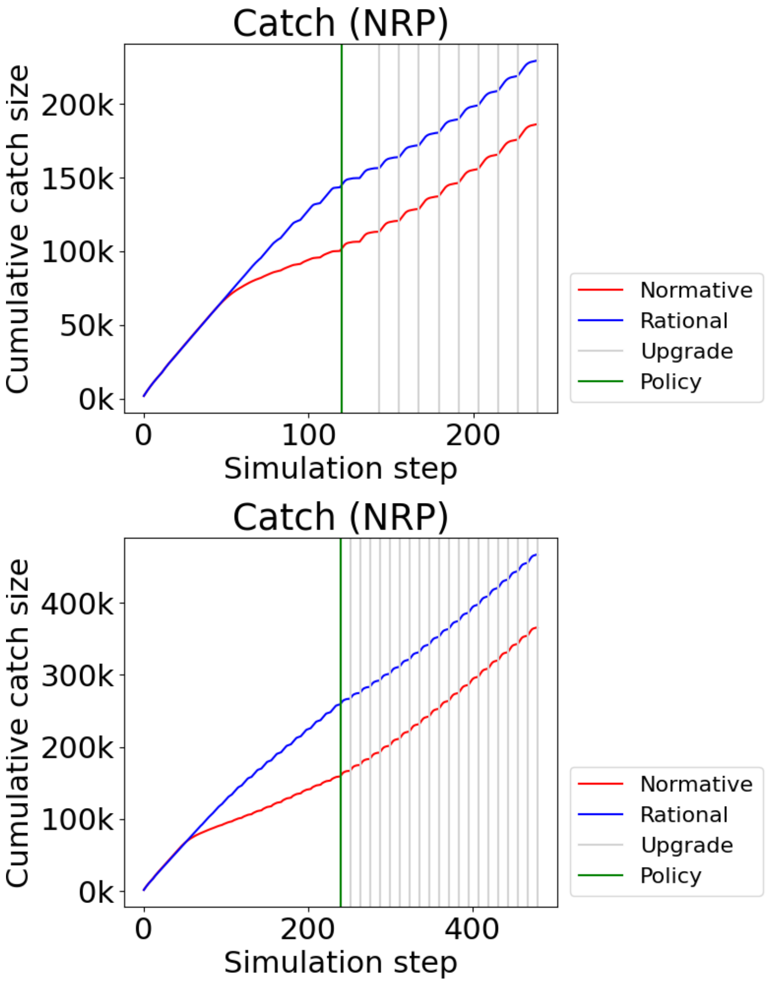


Fig. 6.18: Cumulative catch for simulation runs where the population contains both rational and normative agents. They still diverge around the same cumulative catch value, but it occurs later in the simulation because the resource is more scarce and the normative agents need to fish more in order to save enough money to switch to spending time on shore.

There is no punishment for breaking either the policy or the norm beyond the effects on the value system, which means the results reflect agent satisfaction with the policy (the norm is by definition beneficial to their satisfaction) and not their aversion to punishment for breaking it. The more they break the policy, the more unhappy they would have been if they had stuck to it.

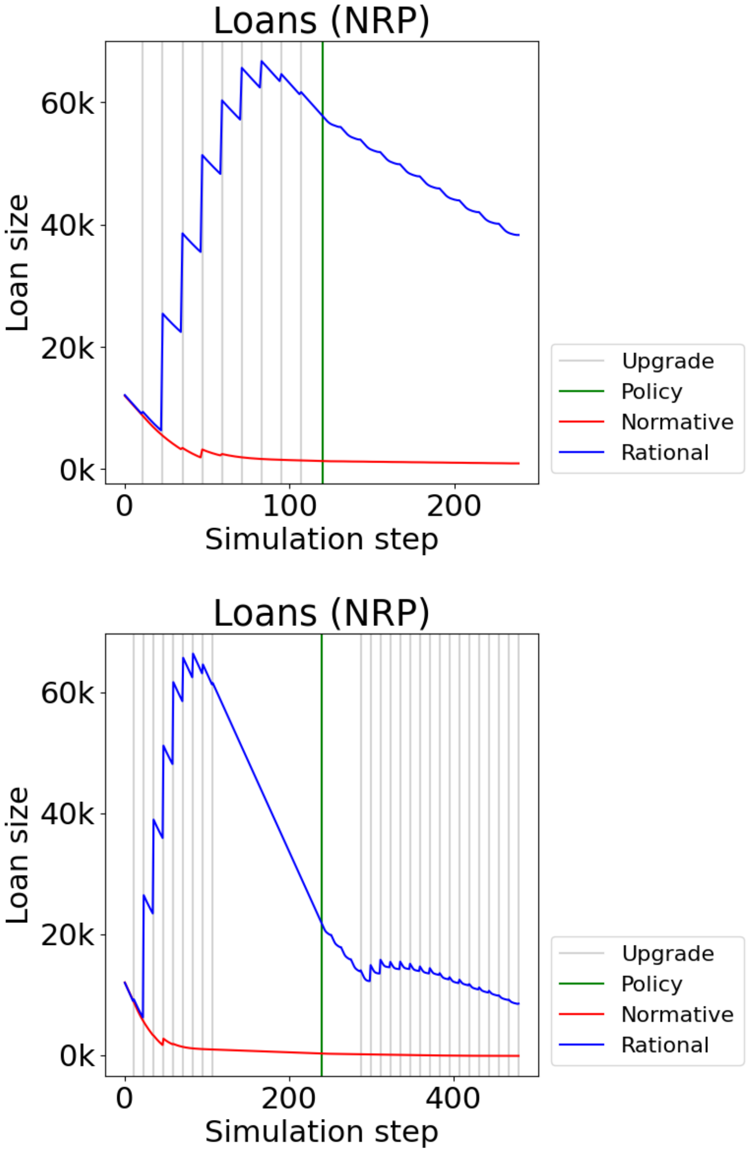


Fig. 6.19: Average loans for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents take on significantly more debt than normative agents, both because their larger fishing effort means they can afford to pay back more, and because normative agents would rather spend time on shore than repay additional debt.

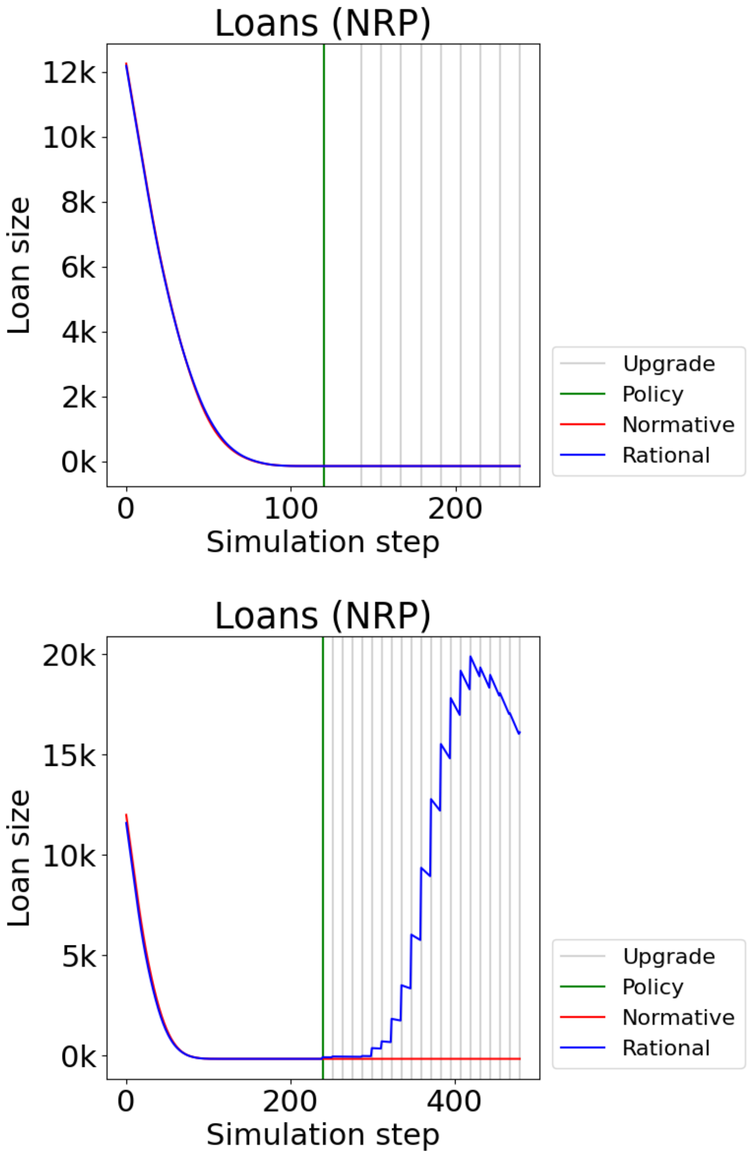


Fig. 6.20: Average loans for simulation runs where the population contains both rational and normative agents fishing a finite resource. Due to the resource becoming more scarce and less profitable as the simulation advances, none of the agents have the means to take on extra loans and limit themselves to repaying the ones they started with.

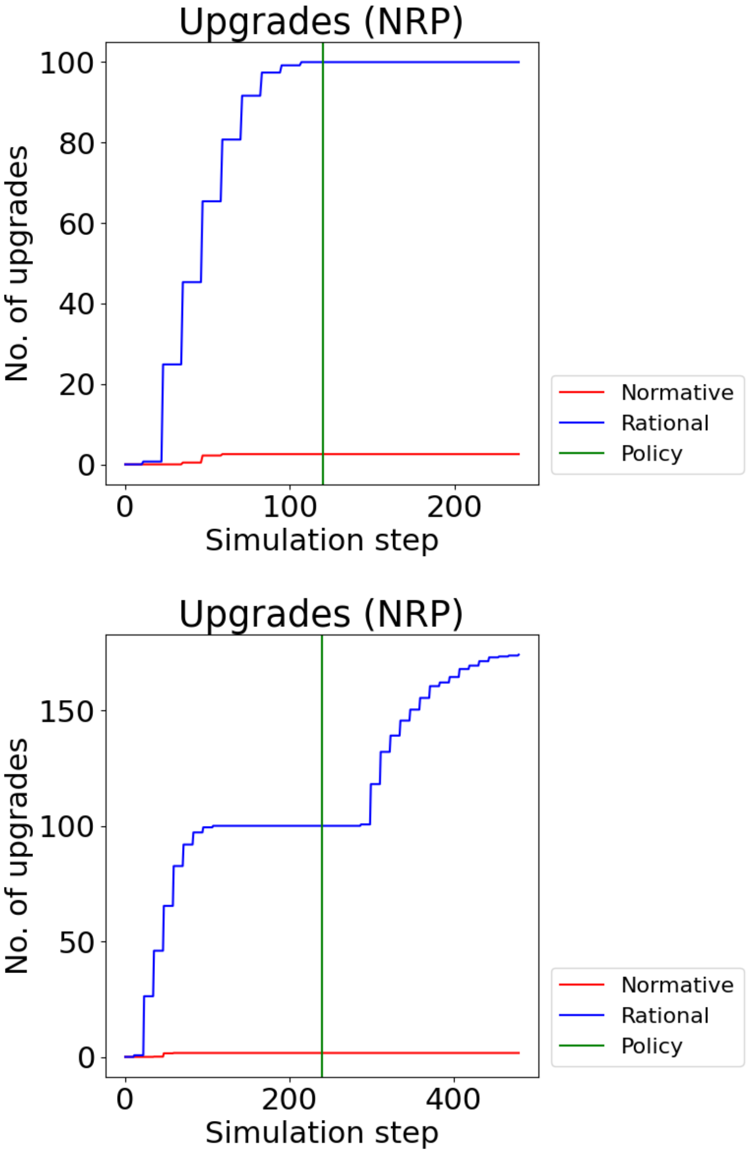


Fig. 6.21: Number of upgrades at any given simulation step for simulation runs where the population contains both rational and normative agents fishing an infinite resource. Rational agents are much more likely to take eon extra loans to buy vessel upgrades, while normative agents prefer spending time on shore to taking on more debt in order to afford more vessel upgrades.

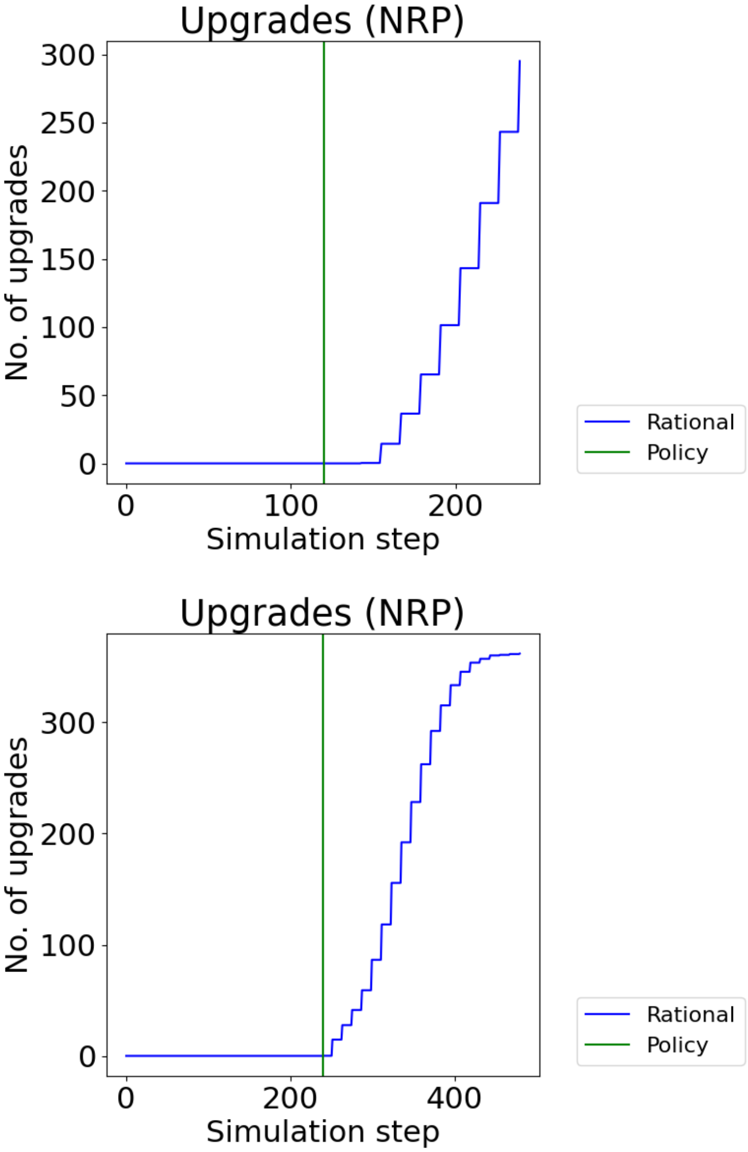


Fig. 6.22: Number of upgrades at any given simulation step for simulation runs where the population contains both rational and normative agents fishing a finite resource. Due to the resource becoming more scarce and less profitable as the simulation advances, neither type of agent can afford to pay their loans or amass sufficient savings to afford any vessel upgrades under these conditions.

Rational Agent	Normative Agent
Actions	
Fish	Fish
GetLoan	GetLoan
SellFish	SellFish
PayDebts	PayDebts
UpgradeVesselEfficiency	UpgradeVesselEfficiency
UpgradeVesselSize	UpgradeVesselSize
SpendTimeOnShore	SpendTimeOnShore
BuyQuota	BuyQuota
SellQuota	SellQuota
FishQuota	FishQuota
Goals	
None	None
DoJobPolicy	DoJobPolicy
Value system	
RationalValueSystem	NormativeValueSystem

Table 6.9: The two complete agent profiles used in the models. Blue highlighted elements are added by the policy when it is introduced into the simulation. Yellow highlighted elements are only present in scenarios involving values.

Because the agents now have the freedom to break both policy and norm, we run slightly different scenario variations this time. All variations include an equal number of agents of each type (100 rational, 100 normative), run for 20 years and the policy comes in effect halfway through a run. The difference lies in the severity of the restriction the policy imposes on the agents. We have one scenario where the resource is finite, which comes with strict restrictions, and one scenario where the resource is scarce and therefore the restrictions are very strict. We don't have a 40-or-more years scenario this time because we are interested in observing how the agents respond to the policy being introduced, which, in turn, can give us an inkling into whether the policy would succeed in its goal of eventually bringing the resource and harvesting to a sustainable point, both economically and ecologically.

The results show that normative agents fish less than "rational" agents, similar to previous scenarios (see fig. 6.24 and fig. 6.28). In the case of normative agents, the fishing effort is much more evenly distributed through the years compared to previous scenarios where it matched "rational" effort for a while, only to curb downwards later. There is also a much smaller difference in upgrades between rational and normative agents (see fig. 6.26 and fig. 6.30). This matches better the real world scenario which inspired these models, as both normative and "rational" fishers upgrade their vessels in real life, but normative fishers do not use their increased effort capacity to fish more, but to fish more comfortably. This is also reflected in the smaller difference in loans (see fig. 6.25 and 6.29). What this means is that, for this calibration of the value trees and the sandbox in general, there is a much slower decline in fishing effort for normative agents compared to previous scenarios.

Both types of agents cheat, albeit to different degrees (see fig. 6.23 and fig. 6.27). "Rational" agents break policy with abandon, regardless of restriction severity since they're very strongly driven by their need to fulfill their **Achievement** value, which happens by maximizing their fishing effort and their profits. Normative agents are driven by **Benevolence**, which is fulfilled by their norm, more than they're driven by **Achievement**. As such, they spend most of their time on shore when they're not fishing their quota.

It is interesting to note that both agent types sell some quota and cheat to make up the difference. This happens because selling quota promotes Achievement by increasing Savings and has no immediate downsides for any other values. It's an artifact of the way the sandbox is built and calibrated. Nothing prevents them, and they stand to gain from doing it, therefore they do it.

6.5 Conclusion

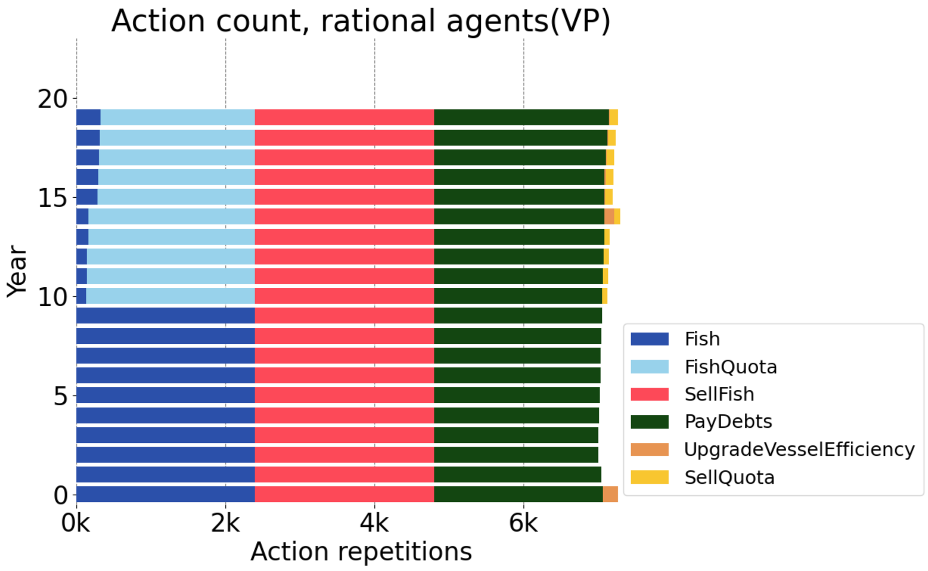
Our results show that agents with a decision architecture that can operate with norms and values behave differently and more realistically (even unexpectedly or unpredictably), compared to self-interested, goal-driven, profit maximizing agents that are more common in ABM.

6.5.1 Normative vs. self-interested agents

Normative agents diverge significantly in their fishing effort from their "rational" counterparts, as evidenced both by their actions and their cumulative catch.

Depending on how many of them have loans to repay at any point during the simulation, normative agents may appear to behave as rational agents. However, normative agents voluntarily curb their fishing effort, regardless of resource abundance, while self-interested agents only do it when the resource is scarce and it becomes unprofitable to continue fishing.

Once we introduce the policy, we can see that both types of agents change their behavior, but in opposite ways: self-interested agents lower their fishing effort, while normative agents raise their fishing effort. This happens because of the quota allocation function of the policy, which assigns fishers their share of quota based on the average fishing effort of the whole small-scale fleet. The differences in fishing effort are significant enough that, if we assume strict adherence



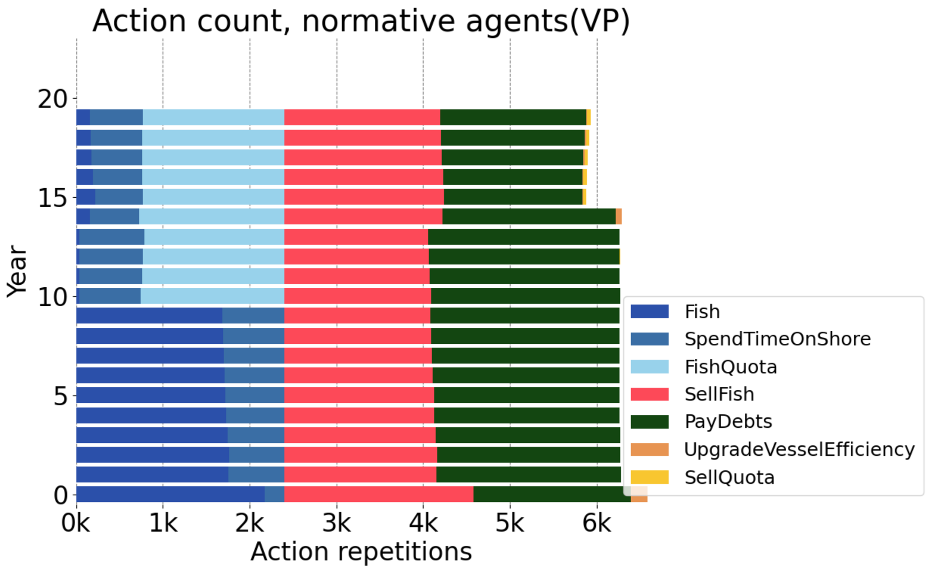


Fig. 6.23: Action count for rational (up) and normative agents, under strict policy conditions. Both agent types cheat, but rational agents cheat considerably more than normative agents

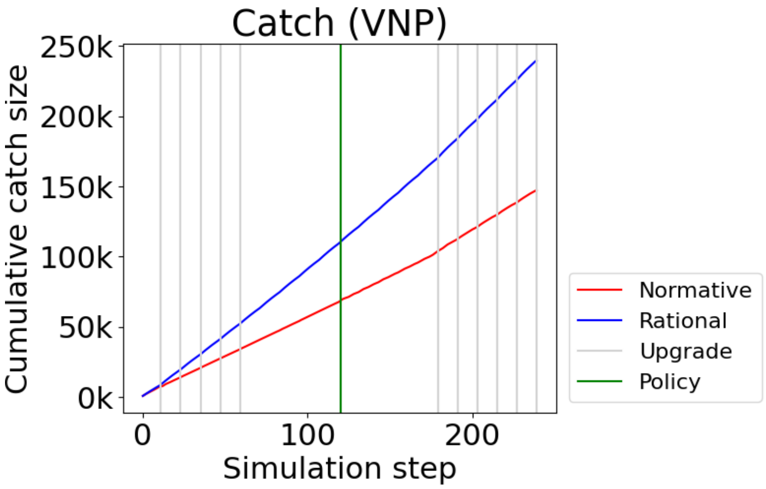


Fig. 6.24: Cumulative catch for rational and normative agents, under strict policy conditions. Normative agents fish less than rational agents, as expected, but their effort is more evenly distributed over time compared to goal-oriented scenarios

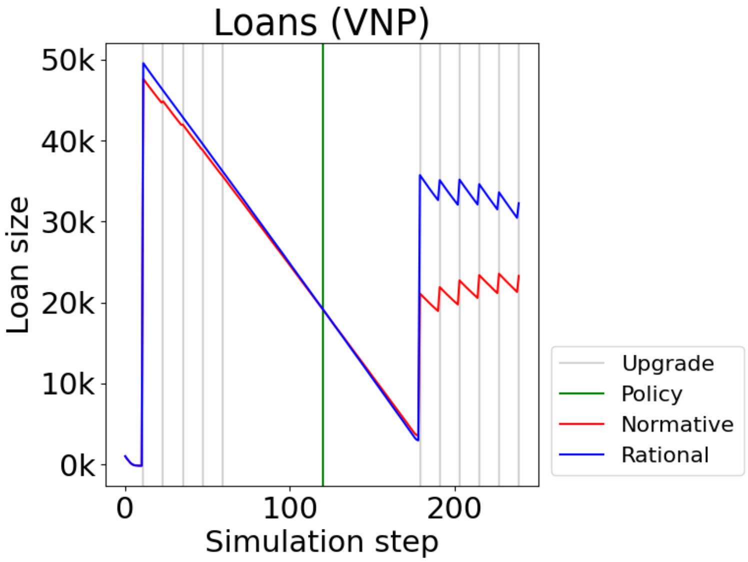


Fig. 6.25: Loans for rational and normative agents, under strict policy conditions. The distance between rational and normative loans is considerably smaller than in goal-oriented scenarios

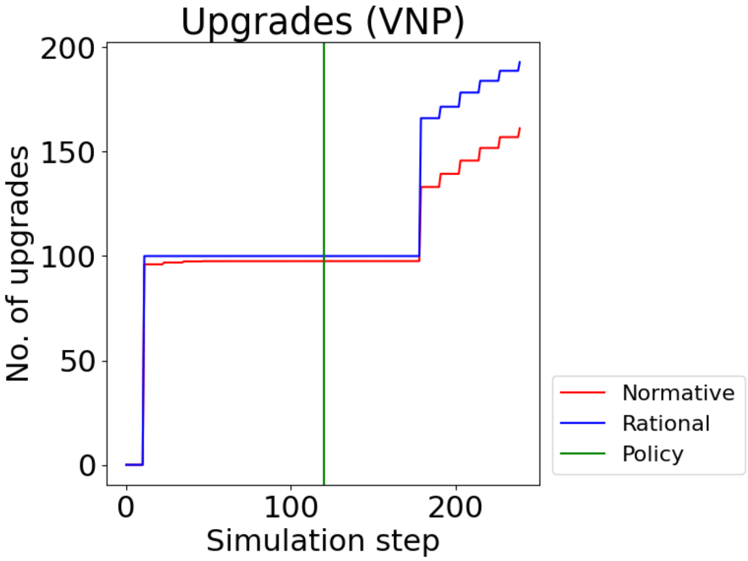
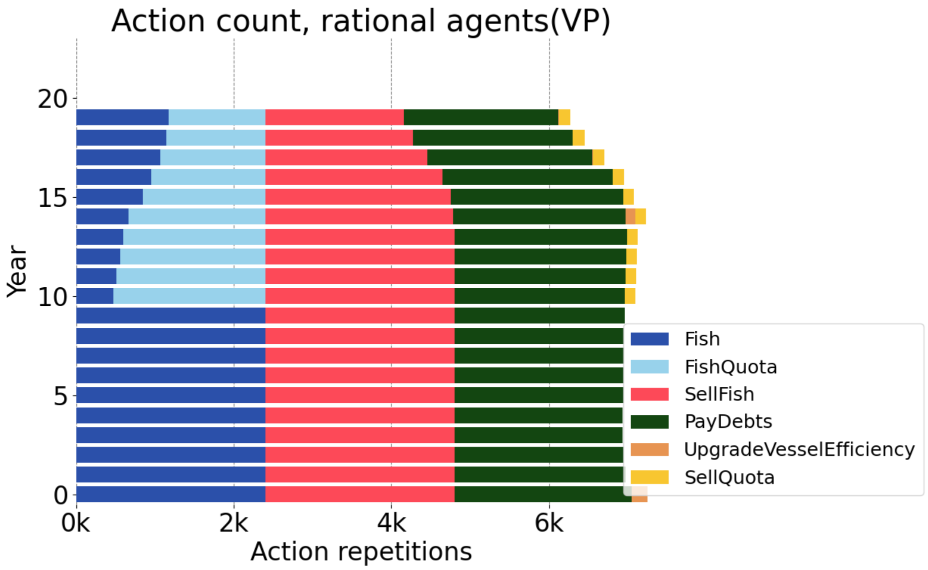


Fig. 6.26: Upgrades for rational and normative agents, under strict policy conditions. The distance between rational and normative upgrades is considerably smaller than in goal-oriented scenarios

to the law, normative fishers have to up their fishing effort to meet their allotted quota, which forces them to break their self-limiting norm.

Introducing norms into an agent's decision making process means the models can start to reflect back to us the ways in which policy affects different groups. In this particular case, while the policy achieves its overall goal of reducing the total yearly catch of the small-scale fleet, it increases the overall fishing effort of the part of the fleet governed by a self-limiting fishing effort norm. However, these scenarios assume that all agents will follow the policy at all times, which limits the insights they can offer. This choice is set as default in the models because agents do not have a way to evaluate whether they would rather follow their social norms or the policy, so the resolution of such conflicts needs to be imposed by the modeler. As such, we can see the effects a given policy will have if all agents behave as directed, but not what would happen if agents could chose to disobey, nor why they would want to.



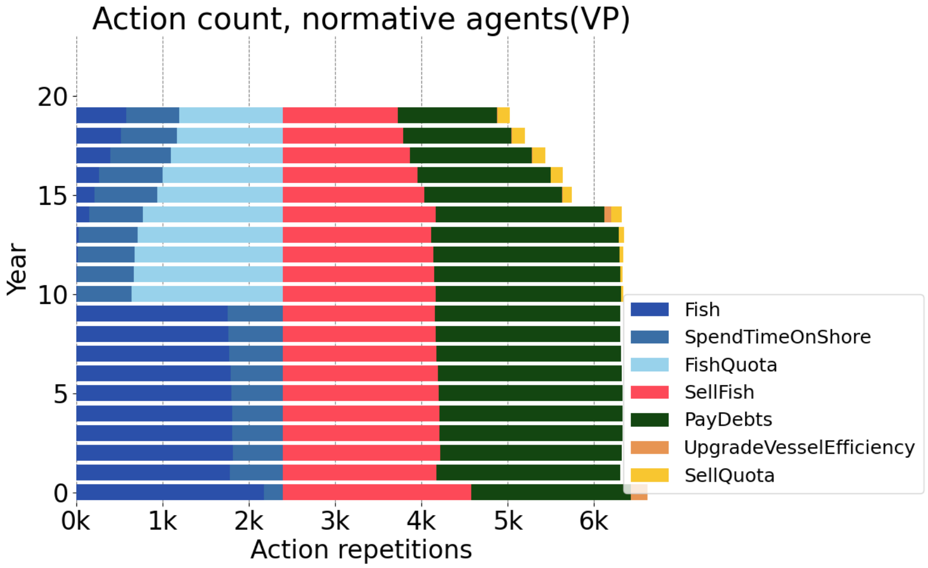


Fig. 6.27: Action count for rational (up) and normative agents, under very strict policy conditions. Given the severity of the restriction imposed by the policy, there is significant fishing over quota from both agent types, which leads to stricter restrictions, which in turn leads to even stricter restrictions and so on.

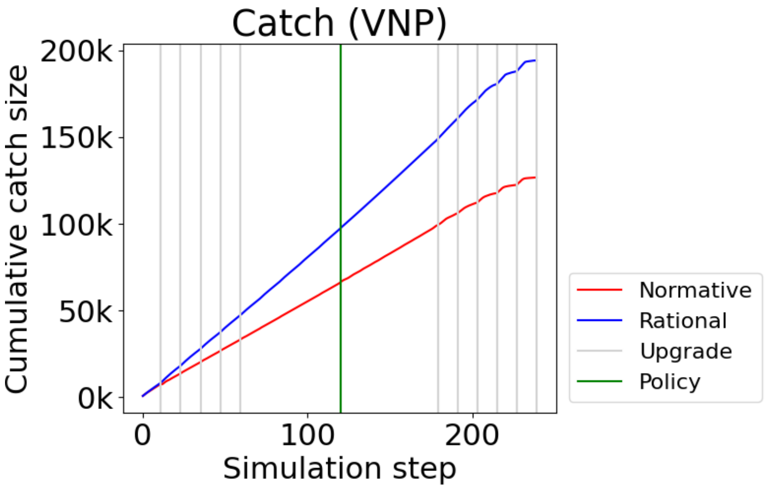


Fig. 6.28: Cumulative catch for rational and normative agents, under very strict policy conditions. Because of the severity of the restrictions imposed by the policy, the agents fish over quota and thus the policy doesn't significantly curb the amount of fish being harvested.

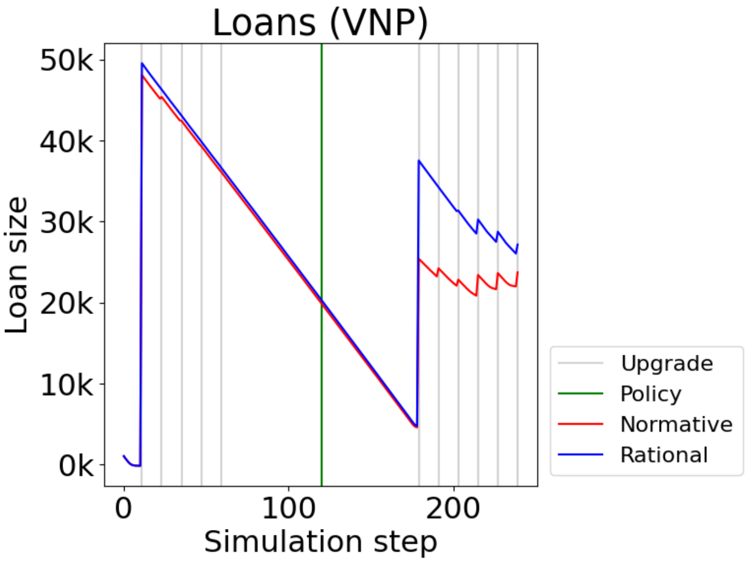


Fig. 6.29: Loans for rational and normative agents, under very strict policy conditions. Despite stricter restrictions, the distance between rational and normative loans is considerably smaller than in goal-oriented scenarios

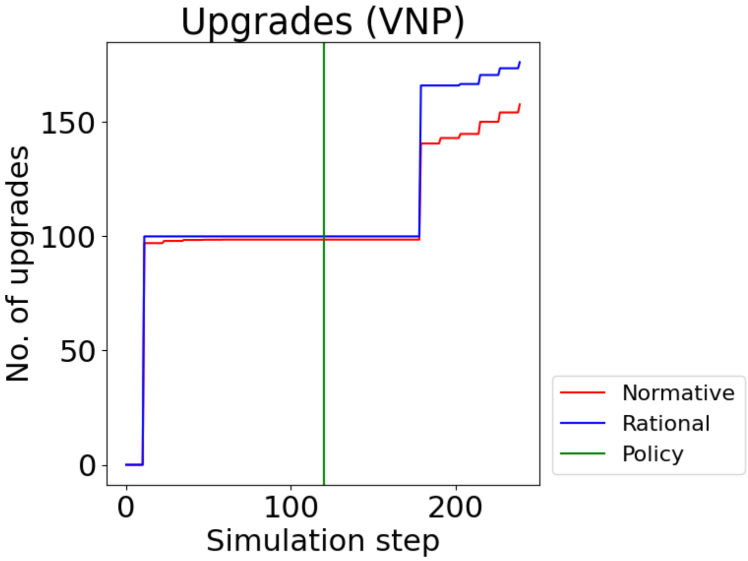


Fig. 6.30: Upgrades for rational and normative agents, under very strict policy conditions. Despite stricter restrictions, the distance between rational and normative upgrades is considerably smaller than in goal-oriented scenarios

6.5.2 Value-driven agents

Agents with norms have the ability to choose which norms to follow, the social ones or the legal ones. When agents no longer have to obey the rules, but are allowed to make calculated decisions about if and when it would be advantageous for them to break them, things become more complicated.

In the case of normative agents, the fishing effort is much more evenly distributed through the years compared to previous scenarios. There is also a much smaller difference in upgrades between rational and normative agents, which is a better match for the real world scenario which inspired these models, as both normative and "rational" fishers upgrade their vessels in real life, but normative fishers do not use their increased effort capacity to fish more, but to fish more comfortably. This is also reflected in the smaller difference in loans between the two agent types.

When it comes to breaking the policy norm, agents who value their gains over conforming to the rules, will break them with impunity. Even agents who value their social norms over policy rules will break the law, even if not quite as much. Both agent types sell some quota and fish illegally to make up the difference. This happens because selling quota promotes **Achievement** by increasing Savings and has no immediate downsides for any other values. It's an artifact of the way the sandbox is built and calibrated with no direct equivalent observed in real life. However, it shows that when agents are given a decision process that allows them to reason about their norms and values, they can exhibit unexpected behavior, which is something that purely normative or goal-driven agents cannot do as readily.

6.5.3 Limitations

Adding punishment would nuance the approach further, leading to more informative results, but it would require us to estimate to what degree people would find the punishment more taxing than the hit to their interests brought about by following the policy. For now, the level of cheating serves as an indicator of whether a given policy will be met with approval, disgruntlement or outright opposition. The results for the scenario with value-driven agents show more nuance in the results, with agents breaking the policy when doing so appears more advantageous to them than upholding it. By comparing the results of scenarios where agents are driven purely by norms and/or goals to those of scenarios where agents are also driven by values we can get an idea of how disruptive the policy is likely to be perceived.

There are, however, two caveats to this approach to building scenarios. First, these scenarios were developed with the benefit of hindsight. The effects of quota policy being introduced to the small scale fisheries of Norway are already known, as are the norms driving part of the fishers in that fleet. Therefore, we could include the norms we already know had the most impact on the behaviour, and make a well educated guess about the values of the different fisher groups. This is a significant advantage when it comes to interpreting the results and assessing the validity of the scenarios. In other, future facing, cases there is no way to know for sure which norms and values will be affected and what alternative behaviors may come to light, not without heavily involving stakeholders in the modeling process. Thus, using this framework and architecture may yield the best results when paired with a participatory approach to building agent based models.

Second, norms and values do not exist in a vacuum. They are part of an interconnected webs of social norms, values, habits, practices, expectations, goals and other motivators which govern social and private life at all levels. Policy impacting parts of this web, such as fishing policy changing the way fishers make a living, will have ripple effects that spread out through the community and, thus, indirectly have an impact far beyond their stated target. In the case of small scale fisheries, these indirect impacts can lead all the way to whole fishing communities becoming ghost towns when fishers are no longer able to make a living and move away in search of different work, followed by fish processing plants closing and their workers moving away and so on. These effects can be delayed, manifesting only once fishers age out of the fishery and their descendants opt to sell the quota and move away. Since the price of quota is usually very high, it acts as a considerable barrier to entry into the fishery and therefore the fishers who exit the fishery by selling the quota are not replaced by young fishers taking up the trade. This eventually leads to the same decline of the fishing community. If we want to model the effects of such policies on the social element of SES, the models must include not only the parts of the system that are directly targeted, but also those that are likely to see cascading effects.

Chapter 7

Conclusion and future work

In this last chapter, we will outline our conclusions and plans for future work. Given the nature of the research, much work remains to be done, especially concerning the design of a participatory modelling protocol in which to put our architecture and sandbox approach to the test.

7.1 Conclusions

We set out to investigate the suitability of ABM for modelling the effects of policy on SES, with the understanding that the human component of such systems is already governed by a set of norms and values which are likely to conflict with the policy being introduced. To this end, we broke the research down into three main questions:

- RQ1: What are the particular challenges SES pose to governance and what role do social norms and values play in them?
- RQ2: How to conceptualize norms, values and policy for use in social simulations?
- RQ3: Does our architecture make agents behave in a more realistic manner leading to more realistic simulation results and more useful insights?

We looked at what SES are, and discussed how their complex and interconnected nature does not lend itself to single clear-cut policy solutions. If such systems are to be governed in a suitable and equitable manner, the people in the system, with their complex motivations and behavior, need to be taken into account as fully fledged individuals, not as self-interested, economically driven caricatures of themselves. In order to understand the people and the multiple facets of such systems, policy design processes would gain immensely by involving stakeholders in order to take advantage of their knowledge and expertise.

Furthermore, whatever solutions are proposed and discussed during the policy design process will likely have unforeseen effects on the SES due to its interconnected nature. This is where models come in handy, chief among them ABMs due to their ability to handle complexity in general and to capture the complex decision processes of people in particular. Since much of the work done on the subject of agent decision architectures comes from MAS, these architectures are not fully suitable for social simulation, nor as agile as we'd like in a participatory modelling setting.

As such, we built our own architecture, which is highly modular and contains explicit representations of the cognitive elements that factor into an agent's decision process. We chose to focus on norms and values, because they are some of the most powerful motivators and the most stable over time. The implementation of the architecture maintains the modularity of the architecture and the explicit representations of the cognitive elements. At the same time, we implemented a "sandbox" environment, complete with visual model, agent and cognitive elements builders, as a first step towards developing a fully functional piece of modeling software to be used in participatory settings.

The following sections give more details regarding the answers to each research question.

7.1.1 Answers to RQ1

> What are the particular challenges SES pose to governance and what role do social norms and values play in them?

7.1.1.1 What are SES

SES are complex system in which human society is deeply intertwined with the natural world. Fisheries, forestry, water and land use, national parks, all of these and more are examples of context in which human and natural elements are fused together to where they form a new, distinctly recognizable kind of system. They are not merely complicated, but irreducible. We cannot separate the human component from the ecological component to study them separately and expect to understand all the dynamics that might be relevant to whatever question we're asking or problem we're attempting to solve. Not only that, but they are also vast, open and their boundaries are fuzzy. The many elements that make up an SES are all connected in a web on interactions, cause and effect, and feedbacks. Where one web ends and another web begins is not really a thing anyone can determine with any certainty.

7.1.1.2 Governance challenges

All this makes governing SES a challenging problem, that is nevertheless of utmost importance. Many of our most difficult contemporary problems arise in SES: overfishing, deforestation, damaging tourism, habitat destruction caused by urban and industrial developments, and, of course, climate change. We sometimes tend to focus on the ecological damage inflicted on these systems, but, in SES, ecological deterioration always entails social deterioration. Damaged ecologies harm the communities that depend on them for continued survival. At the same time, ill-advised policy can bring about social end economic changes that damage the environment, or can harm communities in an attempt to preserve the environment.

The complexity of SES means that evidence-based policy isn't always the best approach, for two main reasons. First, evidence-based policy works best when the problem it's trying to solve has a clear solution, which is rarely the case in SES. Second, scientific inquiry, especially when conducted at the behest of policy-makers, is not without bias. This means that, when it comes to devising policy for SES, it's more likely than not that policy makers will have to choose from multiple solutions proposed and backed by different stakeholders, each with their own agenda and priorities, and none of these solutions will lack support from scientific data and arguments.

Therefore, if evidence alone can't point to a winning solution, policy makers must turn to values, theirs and their stakeholders'. An alternative policy design process would include the participation of stakeholders and inputs from more scientific disciplines than just the ones that favor quantitative methods. Such a process could ensure that different scientific disciplines cover one another's blind spots and correct one another's biases (most notably, overly simplified models of human behavior), while multiple stakeholder perspectives would ensure relevant local knowledge is brought to the table for consideration, and no single party gets free reign to impose their values, perspective and preferred approach on everyone else unchallenged.

7.1.1.3 The relevance of values and social norms

In matters of experimenting with policy alternatives in particular, agents need to have a decision process that can operate with social norms and values, at least. Values describe what kind of world states the agents prefer, and are extremely stable over time. Social norms are collectively agreed upon behaviors and goals that are considered proper and/or expected in certain contexts. Taken together, values and social norms form a particularly stable and consistent framework for the decision making processes of an agent, encompassing both motivations and preferred means of pursuing said motivations. Policy, as another kind of norm, fits in this framework as either supporting/reinforcing (when it promotes the values of an agent or works together with the social norms of an agent) or antagonistic/conflicting (when it goes against the values of an agent or conflicts with and the social norms of an agent).

7.1.2 Answers to RQ2

How to conceptualize norms, values and policy for use in social simulations?

In chapter 3 we looked at the ways these systems are modeled, especially, with regards to stakeholder involvement. The conclusion of this part of the thesis is that while ABM is a uniquely suitable choice for SES modeling, the field still faces some issues in terms of reproducibility, comparability and reusability of models and model components, as well as a rather underdeveloped approach to representing social elements and decision-making processes.

ABM is at its most useful when analytical solutions are either too complex to be computed or lead to an oversimplification of the system being modelled. It can integrate related, connected or interacting elements that would traditionally be treated separately by separate disciplines of science, as well as processes that occur at different scales of time and space. And, especially relevant to the participatory policy design suited to SES governance, it can integrate stakeholder perspectives and integrate *with* stakeholders - to the extent that participatory modelling has become widely accepted in SES modelling.

Despite all these advantages, the field still struggles with some appreciable deficits, such as issues reproducing, comparing and reusing models or parts of models. Moreover, while there is progress being made with regards to the inclusion of nuanced social elements and decision processes, there is certainly room for improvement. The dearth of realistic social and decision models is particularly troublesome if ABM is to be included in the policy design process for governing systems in which the human component is crucial to the functioning and behavior of the system. If the models fail to capture it properly, the results will lack informative power or can be downright misleading.

Steps are being made to overcome the later, with many agent architectures including norms and normative decision processes, as well as values and value-driven decision processes. While not all these architectures are suitable for use in social simulations of SES, they certainly offer significant guidance for what’s possible, relevant and useful, and in what contexts.

7.1.3 Answers to RQ3

Does our architecture make agents behave in a more realistic manner leading to more realistic simulation results and more useful insights?

In an attempt to fill some of the gap left in ABM by either overly simplified decision models or cumbersome architectures designed for MAS, we built our own normative value-driven agent architecture for use in social simulations of policy effects in SES. Its conceptualization and implementation are described in detail in chapters 4 and 5. Our main goal for the architecture was that it have explicit and flexible representations of norms, values and any other cognitive or social component of the agent's decision making processes so that the agents would be able to behave in a more realistic manner when confronted with policy changes, while also remaining lightweight enough to be used in social simulations.

The requirement for explicit representations of norms, values, actions, goals and other cognitive elements led to a strongly modular implementation that allows for a sandbox approach to building models. Any cognitive elements already implemented can be reused, and new agents and simulation scenarios can be created just by mixing and/or recombining them. Thus, many simpler and related scenarios can share most of their cognitive and environmental elements, and thus can be built and run within the same sandbox.

We demonstrate this approach in chapter 6 by building and running a number of different scenarios concerning the introduction of an ITQ fishing policy in fisheries with self-interested goal-driven agents, normative goal-driven agents, self-interested value-driven agents and community-oriented value driven agents, as well as populations containing a combination of these agent types. The simulations are not meant to serve in any real life context, but demonstrate the use and capabilities of the architecture and sandbox approach.

7.2 Future work

This work barely scratches the surface with regards to what can be achieved and what needs to be achieved in order for our approach to building social simulations to become effective when used in a real life study of policy effects in SES. The remainder of this chapter is dedicated to briefly touching on some of the directions this work can take in the future.

7.2.1 Extending the decision architecture

When perusing the literature on the subject of human motivation and decision making, you'll come across other theories and approaches besides the norm-value one we use in this work. Because of the nature of the subject, it's not accurate to say that one of these theories is more "correct" than another, but we can certainly say that a particular theory is likely to be a better fit for the situation we want to model. Thus, in order to ensure the suitability of our architecture for a wider range of application scenarios, we should focus on including other cognitive elements that play a role in the decision making process of humans.

It also bears repeating that the implementation of the architecture focuses on making future developments additive, meaning that any future norms, actions, entities, goals, value systems or reasoners would not have to replace existing elements, nor would existing elements necessarily need to be adapted to fit future developments.

7.2.1.1 Motivators

This work takes a particular view of the human decision process, and the motivators we chose to conceptualize and implement reflect that. We consider goals, norms and values to be some of the most fundamental elements humans use to make decisions, which is why we chose to focus on them first. However, there are other elements at play when it comes to human behavior, and they come into play at different in different decision contexts or at different places in the decision process.

For instance, deciding what to eat on a daily basis, has less to do with norms and values, and more to do with habits when it comes down to the mechanics of the decision process. Norms and values play a role in delimiting the choice space for acceptable foods, but they don't necessarily guide the process of choosing a food from within that space. For instance, one may decide to forgo eating meat for environmental or animal welfare reasons, but this still leaves a huge selection of options. Deciding between a salad or french fries for lunch on any given day is less likely to be actively influenced by any norms, and more likely to be decided based on existing habits. Trying to study the effects of a policy which aims at improving the eating habits of a target community by using agents which can reason about values and norms, but not about habits, may not achieve the best results.

Therefore, we want to expand the architecture to include motivators such as habits, social practices, identity, roles or cultural dimensions. The inclusion of these motivators comes, of course, with a need for reasoners that can deal with them, in combination with any other motivators already part of the sandbox.

Given the modular design of both the conceptual and implemented architecture, these new motivators and their associated reasoners would not require much in the way of integration with existing conceptual structures and code. The most involved aspect of this expansion would be designing the new reasoners so that they'd be able to operate with the particular representation of existing and newly introduced motivators in a way that is consistent with the conceptual or theoretical interactions between them.

7.2.1.2 Social agents

The framework and models dont contain much in the way of explicit social interaction, as vital as it is in normative decision making. The scenarios we used to illustrate the use and functioning of the architecture assume a live and let live norm, combined with conformity people mind their own business, but there is still a desire to not stand out. There is nothing in the architecture that makes defining social interactions and influences more onerous than defining any of the other motivators and behaviors because the conceptual model doesn't forbid functions which take the behavior or mental states of other agents as inputs or outputs. However, we are missing certain essential elements of social interactions, especially where norms and policy are concerned and future development should absolutely include, at the very least, social punishment for norm breaking, and a communication system for the agents.

Another issue of interest is the formation and extinction of norms. This is of particular interest for policy initiatives that bank on morphing into social norms over time, thus forgoing the need for active enforcement efforts in order to guarantee the desired behavior on part of the people.

We have so far considered only norms concerning the behavior of people towards the SES, towards one another and towards their interests, but the success of a policy can live or die on the quality of the relationship between the governing and the governed. A relationship built on trust, combined with norms favourable towards state authority, can ensure even sub-optimal policies will be complied with. Conversely, if the relationship is strained, or the norms favour self-governance to centralized institutions, even a favourable policy may be met with resistance.

7.2.2 Participatory modeling

As we've pointed out in chapters 2 and 3, these models require significant local knowledge about the communities and natural systems being modeled, and one of the more successful approaches in eliciting this knowledge with regards to building ABMs is participatory modelling. The implementation of the architecture and the sandbox approach are already geared towards facilitating the participation of stakeholders in building models, but we do not, as of the writing of this thesis, have a full participatory protocol in place.

For now, the architecture has a couple of properties that we designed in anticipation of using for participatory modelling, but we did not run systematic tests since they are outside the immediate scope of this work. The first of these elements is support for reusability, comparison and reproducibility of models and model elements by virtue of the model and agent architecture. In order to maintain these properties outside the boundaries of one project, the models and their elements are easy to save and disseminate. The second element is the graphic user interface, which is still fairly rudimentary at this point, but makes scenario building easier and more intuitive. We will have to conduct user studies in order to refine the UI and UX to where we can be confident the software is indeed comfortable and engaging to use for stakeholders.

Given the use of the Unity game engine for implementation and the early efforts at a visual interface for building models we presented in this work, we consider companion modelling [50] to be the most likely candidate for developing an effective participatory protocol.

7.2.2.1 Sandbox utilities

In keeping with the participatory approach, the sandbox needs to gain certain utilities to help modelers organize the various elements used in building the models.

- Descriptive scenario builder: it would allow stakeholders a more hands-on approach while taking part in the participatory modelling process. Additionally, it would allow a faster iteration time for each scenario development, up to building scenarios on the spot, if they are just recombinations of already existing sandbox elements. This would encourage stakeholders to build and experiment with more "what-if" scenarios, and thus gain a better understanding of both the SES in question and of the possibilities afforded by ABM, hopefully fueling the next iteration of the participatory modelling process
- Automatic model verification: when building scenarios by combining sandbox elements, it'd be extremely useful to have the system notify the user of any conflicts. This is already a significant part of software development and there are methodologies that help detect issues arising from incompatible system behaviors. Because there are very few limitations imposed on the nature of the cognitive elements in the architecture, this will have to be a partial verification since we can't guarantee that we'll be able to detect all normative conflicts, for example. However, even partial support in this regard takes off much of the cognitive effort involved in designing these scenarios, with the caveat that users must be made very aware that the verification process has limits and what those limits are.

7.3 Concluding remarks

This thesis stubbornly attempts to maintain a high level view of socio-ecological systems, their governance challenges, the agent based models which can support the study of such systems and the policy design process, stakeholders and participatory approaches. It would be tempting to pick a subject and go on a deep research dive in order to gain as much expertise as possible. After all, isn't that what a PhD is for?

In its own way, this thesis is trying to make the point that no, not all PhDs are for specializing until your expertise is as narrow and as sharp as a needle point. When confronted with something as complex and vast as a socio-ecological system, or a problem as wicked and far reaching as climate change, someone needs to occasionally come up with a reminder of the larger picture, even if it means some of the details get lost in the process. None of us can study an SES by themselves, nor solve overfishing, nor stop or reverse deforestation, devise policy that would get everyone to only use renewable sources of energy. These problems require that science, policy makers, stakeholders and the larger public all cooperate to manage these affairs as effectively, equitably and ethically as possible. I hope we, who tangle with SES, get to keep that in mind throughout our research efforts.

References

1. James M Acheson. *Capturing the commons: devising institutions to manage the Maine lobster industry*. Upne, 2003.
2. James M Acheson. Institutional failure in resource management. *Annu. Rev. Anthropol.*, 35:117–134, 2006.
3. James M Acheson and Roy Gardner. Modeling disaster: the failure of the management of the new england groundfish industry. *North American Journal of Fisheries Management*, 31(6):1005–1018, 2011.
4. S Agnarsson, T Matthiasson, and F Giry. Consolidation and distribution of quota holdings in the icelandic fisheries. *Marine Policy*, 72:263–270, 2016.
5. Icek Ajzen. The theory of planned behavior. *Organizational behavior and human decision processes*, 50(2):179–211, 1991.
6. Andrew EF Allison, Mark E Dickson, Karen T Fisher, and Simon F Thrush. Dilemmas of modelling and decision-making in environmental research. *Environmental Modelling & Software*, 99:147–155, 2018.
7. Edward H Allison, Blake D Ratner, Björn Åsgård, Rolf Willmann, Robert Pomeroy, and John Kurien. Rights-based fisheries governance: from fishing rights to human rights. *Fish and Fisheries*, 13(1):14–29, 2012.
8. Li An. Modeling human decisions in coupled human and natural systems: Review of agent-based models. *Ecological Modelling*, 229:25–36, 2012.
9. Li An, Volker Grimm, and Billie L Turner II. Meeting grand challenges in agent-based models. *Journal of Artificial Societies and Social Simulation*, 23(1), 2020.
10. Ragnar Arnason. Icelands itq system creates new wealth. *The electronic journal of sustainable development*, 1(2):35–41, 2008.
11. Kathleen Emery Auth. *Fishing for common ground: broadening the definition of'rights-based'fisheries management in Iceland's Westfjords*. PhD thesis, 2012.
12. Tina Balke and Nigel Gilbert. How do agents make decisions? a survey. *Journal of Artificial Societies and Social Simulation*, 17(4):13, 2014.
13. Gary S Becker. Crime and punishment: An economic approach. In *The economic dimensions of crime*, pages 13–68. Springer, 1968.
14. Tristan M Behrens, Koen V Hindriks, and Jürgen Dix. Towards an environment interface standard for agent platforms. *Annals of Mathematics and Artificial Intelligence*, 61(4):261–295, 2011.
15. Karl Benediktsson and Anna Karlsdóttir. Iceland: crisis and regional development–thanks for all the fish? *European urban and regional studies*, 18(2):228–235, 2011.
16. Eiríkur Bergmann. Sense of sovereignty. how national sentiments have influenced iceland's european policy. *Stjórnmál og stjórnsỳsla*, 5(2):203, 2009.
17. Charles Bethea. Conjuring maines clairvoyant kush, April 2021. [Online; posted 19-April-2021].
18. Reinette Biggs, Alta De Vos, Rika Preiser, Hayley Clements, Kristine Maciejewski, and Maja Schlüter. *The Routledge handbook of research methods for social-ecological systems*. Taylor & Francis, 2021.
19. Mike Bithell, Edmund Chattoe-Brown, and Bruce Edmonds. The large-scale, systematic and iterated comparison of agent-based policy models. In *Advances in Social Simulation*, pages 367–378. Springer, 2022.
20. Jessica Blythe, Jennifer Silver, Louisa Evans, Derek Armitage, Nathan J Bennett, Michele-Lee Moore, Tiffany H Morrison, and Katrina Brown. The dark side of transformation: latent risks in contemporary sustainability discourse. *Antipode*, 50(5):1206–1223, 2018.
21. Rafael H Bordini, Jomi F Hübner, and Renata Vieira. Jason and the golden fleece of agent-oriented programming. In *Multi-agent programming*, pages 3–37. Springer, 2005.
22. Paul J Brantingham, Patricia L Brantingham, et al. *Environmental criminology*. Sage Publications Beverly Hills, CA, 1981.
23. Calum Brown and Mark Rounsevell. How can social–ecological system models simulate the emergence of social–ecological crises? *People and Nature*, 3(1):88–103, 2021.
24. Muffy Calder, Claire Craig, Dave Culley, Richard de Cani, Christl A Donnelly, Rowan Douglas, Bruce Edmonds, Jonathon Gascoigne, Nigel Gilbert, Caroline Hargrove, et al. Computational modelling for decision-making: where, why, what, who and how. *Royal Society open science*, 5(6):172096, 2018.
25. Marco Campennì. On the immergence of norms: a normative agent architecture. In *In Proceedings of AAAI Symposium, Social and Organizational Aspects of Intelligence, Washington DC*. Citeseer, 2007.

26. Cristiano Castelfranchi, Frank Dignum, Catholijn M Jonker, and Jan Treur. Deliberative normative agents: Principles and architecture. In *International workshop on agent theories, architectures, and languages*, pages 364–378. Springer, 1999.
27. Catherine Chambers, Níels Einarsson, and Anna Karlsdóttir. *Small-Scale Fisheries in Iceland: Local Voices and Global Complexities*, pages 329–349. Springer International Publishing, Cham, 2020.
28. Catherine Chambers, Guðrún Helgadóttir, and Courtney Carothers. little kings: community, change and conflict in icelandic fisheries. *Maritime Studies*, 16(1):1–26, 2017.
29. Nazli Choucri. Mapping sustainability. In *Mapping Sustainability*, pages 3–28. Springer, 2007.
30. Cindy Chu. Thirty years later: the global growth of itqs and their influence on stock status in marine fisheries. *Fish and Fisheries*, 10(2):217–230, 2009.
31. Adrienne Chung and Rajiv N Rimal. Social norms: A review. *Review of Communication Research*, 4:1–28, 2016.
32. Matthew J Colloff, Sandra Lavorel, Lorrae E van Kerkhoff, Carina A Wyborn, Ioan Fazey, Russell Gorddard, Georgina M Mace, Wendy B Foden, Michael Dunlop, I Colin Prentice, et al. Transforming conservation science and practice for a postnormal world. *Conservation Biology*, 31(5):1008–1017, 2017.
33. Rosaria Conte and Mario Paolucci. On agent-based modeling and computational social science. *Frontiers in psychology*, 5:668, 2014.
34. Muriel Cote and Andrea J Nightingale. Resilience thinking meets social theory: situating social change in socio-ecological systems (ses) research. *Progress in human geography*, 36(4):475–489, 2012.
35. Mehdi Dastani. 2apl: a practical agent programming language. *Autonomous agents and multi-agent systems*, 16(3):214–248, 2008.
36. Judith IM De Groot and Geertje Schuitema. How to make the unpopular popular? policy characteristics, social norms and the acceptability of environmental policies. *Environmental Science & Policy*, 19:100–107, 2012.
37. Frank Dignum. Autonomous agents with norms. *Artificial intelligence and law*, 7(1):69–79, 1999.
38. Frank Dignum. Foundations of social simulations for crisis situations. In *Social Simulation for a Crisis*, pages 15–37. Springer, 2021.
39. Frank Dignum, Rui Prada, and Gert Jan Hofstede. From autistic to social agents. In *Proceedings of the 2014 international conference on Autonomous Agents and Multi-Agent Systems*, pages 1161–1164, 2014.
40. Virginia Dignum. Social agents: bridging simulation and engineering. *Communications of the ACM*, 60(11):32–34, 2017.
41. Virginia Dignum and Frank Dignum. Societal challenges need social agents. In *2017 AAAI Spring Symposium Series*, 2017.
42. Graham J Edgar, Trevor J Ward, and Rick D Stuart-Smith. Weaknesses in stock assessment modelling and management practices affect fisheries sustainability. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 29(11):2010–2016, 2019.
43. Bruce Edmonds et al. Some pitfalls to beware when applying models to issues of policy relevance. In *Simulating Social Complexity*, pages 801–822. Springer, 2017.
44. Bruce Edmonds et al. Using agent-based modelling to inform policy–what could possibly go wrong? In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 1–16. Springer, 2018.
45. Bruce Edmonds and Scott Moss. From kiss to kids–an anti-simplisticmodelling approach. In *International workshop on multi-agent systems and agent-based simulation*, pages 130–144. Springer, 2004.
46. Bruce Edmonds and Gary Polhill. Open modelling for simulators. In *Societal benefits of freely accessible technologies and knowledge resources*, pages 237–254. IGI Global, 2015.
47. Kristinn Nikulás Edvardsson, Cezara Păstrăv, and Karl Benediktsson. Mapping the geographical consolidation of fishing activities in iceland during the maturation of the itq fisheries management system. *Applied geography*, 97:85–97, 2018.
48. Lukas Egli, Hanna Weise, Viktoriia Radchuk, Ralf Seppelt, and Volker Grimm. Exploring resilience with agent-based models: state of the art, knowledge gaps and recommendations for coping with multidimensionality. *Ecological Complexity*, 40:100718, 2019.
49. Joshua M Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.
50. Michel Étienne. *Companion modelling: a participatory approach to support sustainable development*. Springer, 2013.

51. Einar Eythórsson. Coastal communities and itq management. the case of icelandic fisheries. *Sociologia Ruralis*, 36(2):212–223, 1996.
52. Einar Eythórsson. A decade of itq-management in icelandic fisheries: consolidation without consensus. *Marine Policy*, 24(6):483–492, 2000.
53. Amy Fallon, Bruce Lankford, and Derek Weston. Navigating wicked water governance in the" solutionscape" of science, policy, practice, and participation. *Ecology and Society*, 26(2), 2021.
54. RIC Chris Francis. Data weighting in statistical fisheries stock assessment models. *Canadian Journal of Fisheries and Aquatic Sciences*, 68(6):1124–1138, 2011.
55. Katia Frangoudes and Manuel Bellanger. Fishers opinions on marketization of property rights and the quota system in france. *Marine Policy*, 80:107–112, 2017.
56. Elizabeth A Fulton, Anthony DM Smith, David C Smith, and Ingrid E van Putten. Human behaviour: the key source of uncertainty in fisheries management. *Fish and fisheries*, 12(1):2–17, 2011.
57. Suzanne H Gage and Harry R Sumnall. Rat park: How a rat paradise changed the narrative of addiction. *Addiction*, 114(5):917–922, 2019.
58. Christian Gehring, Stelian Coros, Marco Hutter, Michael Bloesch, Markus A Hoepflinger, and Roland Siegwart. Control of dynamic gaits for a quadrupedal robot. In *2013 IEEE international conference on Robotics and automation*, pages 3287–3292. IEEE, 2013.
59. Nicholas M Gotts, George AK van Voorn, J Gareth Polhill, Eline de Jong, Bruce Edmonds, Gert Jan Hofstede, and Ruth Meyer. Agent-based modelling of socio-ecological systems: Models, projects and ontologies. *Ecological Complexity*, 40:100728, 2019.
60. Sonia Graham, Jon Barnett, Ruth Fincher, Anna Hurlimann, Colette Mortreux, and Elissa Waters. The social values at risk from sea-level rise. *Environmental Impact Assessment Review*, 41:45–52, 2013.
61. Jonathan Gray, Jason Hilton, and Jakub Bijak. Choosing the choice: Reflections on modelling decisions and behaviour in demographic agent-based models. *Population Studies*, 71(sup1):85–97, 2017.
62. Volker Grimm, Steven F Railsback, Christian E Vincenot, Uta Berger, Cara Gallagher, Donald L DeAngelis, Bruce Edmonds, Jiaqi Ge, Jarl Giske, Juergen Groeneveld, et al. The odd protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, 23(2), 2020.
63. Volker Grimm, Eloy Revilla, Uta Berger, Florian Jeltsch, Wolf M Mooij, Steven F Railsback, Hans-Hermann Thulke, Jacob Weiner, Thorsten Wiegand, and Donald L DeAngelis. Pattern-oriented modeling of agent-based complex systems: lessons from ecology. *science*, 310(5750):987–991, 2005.
64. Jürgen Groeneveld, Birgit Müller, Carsten M Buchmann, Gunnar Dressler, Cheng Guo, Niklas Hase, Falk Hoffmann, F John, Christian Klassert, T Lauf, et al. Theoretical foundations of human decision-making in agent-based land use models–a review. *Environmental modelling & software*, 87:39–48, 2017.
65. Madeleine Gustavsson. Examining the cultural sustainabilityof two different ways of governing fishing practices. *Marine Policy*, 97:262–269, 2018.
66. Martyn Hammersley. Is the evidence-based practice movement doing more good than harm? reflections on iain chalmers' case for research-based policy making and practice. *Evidence & policy: a journal of research, debate and practice*, 1(1):85–100, 2005.
67. Aaron Hatcher and Sean Pascoe. Non-compliance and fisheries policy formulation. In *Developments in aquaculture and fisheries science*, volume 36, pages 355–373. Elsevier, 2006.
68. Samaneh Heidari, Maarten Jensen, and Frank Dignum. Simulations with values. *Advances in Social Simulation*, pages 201–215, 2020.
69. Samaneh Heidari, Nanda Wijermans, and Frank Dignum. Agents with dynamic social norms. Springer, 2019.
70. Jason Hickel. The contradiction of the sustainable development goals: Growth versus ecology on a finite planet. *Sustainable Development*, 27(5):873–884, 2019.
71. Ray Hilborn. The state of the art in stock assessment: where we are and where we are going. *Scientia Marina*, 67(S1):15–20, 2003.
72. Ray Hilborn. Managing fisheries is managing people: what has been learned? *Fish and Fisheries*, 8(4):285–296, 2007.
73. Ray Hilborn. Measuring fisheries management performance. *ICES Journal of Marine Science*, 10 2020. fsaa119.
74. Koen V Hindriks. Programming rational agents in goal. In *Multi-agent programming*, pages 119–157. Springer, 2009.
75. Isla D Hodgson, Steve M Redpath, Anke Fischer, and Juliette Young. Who knows best? understanding the use of research-based knowledge in conservation conflicts. *Journal of environmental management*, 231:1065–1075, 2019.

76. Eriko Hoshino, Ingrid van Putten, Sean Pascoe, and Simon Vieira. Individual transferable quotas in achieving multiple objectives of fisheries management. *Marine Policy*, 113:103744, 2020.
77. Suvi Ignatius, Alyne Delaney, and Päivi Haapasaari. Socio-cultural values as a dimension of fisheries governance: The cases of baltic salmon and herring. *Environmental science & policy*, 94:1–8, 2019.
78. Valur Ingimundarson. Fighting the cod wars in the cold war: Iceland's challenge to the western alliance in the 1970s. *The RUSI journal*, 148(3):88–94, 2003.
79. Christopher D Ives and Dave Kendal. The role of social values in the management of ecological systems. *Journal of environmental management*, 144:67–72, 2014.
80. Rebecca A Kelly, Anthony J Jakeman, Olivier Barreteau, Mark E Borsuk, Sondoss ElSawah, Serena H Hamilton, Hans Jørgen Henriksen, Sakari Kuikka, Holger R Maier, Andrea Emilio Rizzoli, et al. Selecting among five common modelling approaches for integrated environmental assessment and management. *Environmental modelling & software*, 47:159–181, 2013.
81. Matthias Kokorsch, Anna Karlsdóttir, and Karl Benediktsson. Improving or overturning the itq system? views of stakeholders in icelandic fisheries. *Maritime Studies*, 14(1):1–22, 2015.
82. Martin J Kollingbaum and Timothy J Norman. Norm adoption in the noa agent architecture. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 1038–1039, 2003.
83. Gunnar Helgi Kristinsson. Iceland: Vulnerability in a fish-based economy. *Cooperation and conflict*, 22(4):245–253, 1987.
84. Rebecca L Lewison, Candan U Soykan, and Janet Franklin. Mapping the bycatch seascape: multispecies and multi-scale spatial patterns of fisheries bycatch. *Ecological Applications*, 19(4):920–930, 2009.
85. Beth Macy. *Dopesick: Dealers, doctors, and the drug company that addicted America*. Little, Brown, 2018.
86. Moamin A Mahmoud, Mohd Sharifuddin Ahmad, Mohd Zaliman Mohd Yusoff, and Aida Mustapha. A review of norms and normative multiagent systems. *The Scientific World Journal*, 2014, 2014.
87. David Marino Jr. More than 90% of maine towns still dont allow recreational marijuana sales, September 2021. [Online; posted 20-September-2021].
88. Mark N Maunder and André E Punt. A review of integrated analysis in fisheries stock assessment. *Fisheries Research*, 142:61–74, 2013.
89. Anita Maurstad. To fish or not to fish: small-scale fishing and changing regulations of the cod fishery in northern norway. *Human organization*, pages 37–47, 2000.
90. Katie McShane. Why environmental ethics shouldnt give up on intrinsic value. *Environmental Ethics*, 29(1):43–61, 2007.
91. Eugenia Merayo, Rasmus Nielsen, Ayoe Hoff, and Max Nielsen. Are individual transferable quotas an adequate solution to overfishing and overcapacity? evidence from danish fisheries. *Marine Policy*, 87:167–176, 2018.
92. Fabiola V Miranda and Wolfgang B Stotz. Why do state interventions in artisanal fisheries often fail in chile? misalignment to the way of life of fishers. *Marine Policy*, 132:104693, 2021.
93. Hannah Muelder and Tatiana Filatova. One theory-many formalizations: Testing different code implementations of the theory of planned behaviour in energy agent-based models. *JASSS*, 2018.
94. Roldan Muradian and Unai Pascual. A typology of elementary forms of human-nature relations: a contribution to the valuation debate. *Current opinion in environmental sustainability*, 35:8–14, 2018.
95. Andrea Joslyn Nightingale, Siri Eriksen, Marcus Taylor, Timothy Forsyth, Mark Pelling, Andrew Newsham, Emily Boyd, Katrina Brown, Blane Harvey, Lindsey Jones, et al. Beyond technical fixes: Climate solutions and the great derangement. *Climate and Development*, 12(4):343–352, 2020.
96. Frode Olsgard, Morten T Schaanning, Stephen Widdicombe, Mike A Kendall, and Melanie C Austen. Effects of bottom trawling on ecosystem functioning. *Journal of Experimental Marine Biology and Ecology*, 366(1-2):123–133, 2008.
97. Julia Olson. Understanding and contextualizing social impacts from the privatization of fisheries: An overview. *Ocean & Coastal Management*, 54(5):353–363, 2011.
98. Elinor Ostrom. A general framework for analyzing sustainability of social-ecological systems. *Science*, 325(5939):419–422, 2009.
99. Daniel Ovando, Christopher M Free, Olaf P Jensen, and Ray Hilborn. A history and evaluation of catch-only stock assessment models. *Fish and Fisheries*, 23(3):616–630, 2022.
100. Rodrigo Oyanedel, Stefan Gelcich, and EJ Milner-Gulland. A synthesis of (non-) compliance theories with applications to small-scale fisheries research and practice. *Fish and Fisheries*, 21(6):1120–1134, 2020.

101. David OSullivan, Tom Evans, Steven Manson, Sara Metcalf, Arika Ligmann-Zielinska, and Chris Bone. Strategic directions for agent-based modeling: avoiding the yaawn syndrome. *Journal of land use science*, 11(2):177–187, 2016.
102. Gísli Pálsson and Agnar Helgason. Figuring fish and measuring men: the individual transferable quota system in the icelandic cod fishery. *Ocean & Coastal Management*, 28(1-3):117–146, 1995.
103. Sean Pascoe, Eriko Hoshino, Ingrid Van Putten, and Simon Vieira. Retrospective assessment of itqs to inform research needs and to improve their future design and performance. *FRDC, Canberra*, 2019.
104. Laura Pereira, Nadia Sitas, Federica Ravera, Amanda Jimenez-Aceituno, and Andrew Merrie. Building capacities for transformative change towards sustainability: imagination in intergovernmental science-policy scenario processes. *Elementa: Science of the Anthropocene*, 7, 2019.
105. Caroline Criado Perez. *Invisible women: Data bias in a world designed for men*. Abrams, 2019.
106. Alexander Pokahr, Lars Braubach, and Winfried Lamersdorf. Jadex: A bdi reasoning engine. In *Multi-agent programming*, pages 149–174. Springer, 2005.
107. Richard B Pollnac. Social and cultural characteristics of fishing peoples. *Marine & Freshwater Behaviour & Phy*, 14(1):23–39, 1988.
108. Rika Preiser, Reinette Biggs, Alta De Vos, and Carl Folke. Social-ecological systems as complex adaptive systems. *Ecology and Society*, 23(4), 2018.
109. Sam Quinones. *Dreamland: The true tale of America's opiate epidemic*. Bloomsbury Publishing USA, 2015.
110. Marc Raibert, Kevin Blankespoor, Gabriel Nelson, and Rob Playter. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes*, 41(2):10822–10825, 2008.
111. Sabaheta Ramcilovic-Suominen and Graham Epstein. Towards an analytical framework for forest law compliance. *International Forestry Review*, 14(3):326–336, 2012.
112. Jerome R Ravetz. Usable knowledge, usable ignorance: incomplete science with policy implications. *Knowledge*, 9(1):87–116, 1987.
113. Steve Rayner. Uncomfortable knowledge: the social construction of ignorance in science and environmental policy discourses. *Economy and Society*, 41(1):107–125, 2012.
114. Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.
115. Horst W Rittel and Melvin M Webber. Wicked problems. *Man-made Futures*, 26(1):272–280, 1974.
116. Mark DA Rounsevell, Derek T Robinson, and Dave Murray-Rust. From actors to agents in socio-ecological systems models. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1586):259–269, 2012.
117. Diego Salgueiro-Otero and Elena Ojea. A better understanding of social-ecological systems is needed for adapting fisheries to climate change. *Marine Policy*, 122:104123, 2020.
118. Andrea Saltelli and Mario Giampietro. What is wrong with evidence based policy, and how can it be improved? *Futures*, 91:62–71, 2017.
119. Daniel Sarewitz. Science and environmental policy: an excess of objectivity. 2000.
120. Paula Satizábal. The unintended consequences of responsible fishingfor small-scale fisheries: lessons from the pacific coast of colombia. *Marine Policy*, 89:50–57, 2018.
121. Jens Schade and Bernhard Schlag. Acceptability of urban transport pricing strategies. *Transportation Research Part F: Traffic Psychology and Behaviour*, 6(1):45–61, 2003.
122. Thomas C Schelling. Dynamic models of segregation. *Journal of mathematical sociology*, 1(2):143–186, 1971.
123. Caroline Schill, John M Anderies, Therese Lindahl, Carl Folke, Stephen Polasky, Juan Camilo Cárdenas, Anne-Sophie Crépin, Marco A Janssen, Jon Norberg, and Maja Schlüter. A more dynamic understanding of human behaviour for the anthropocene. *Nature Sustainability*, 2(12):1075–1082, 2019.
124. Maja Schlüter, Andres Baeza, Gunnar Dressler, Karin Frank, Jürgen Groeneveld, Wander Jager, Marco A Janssen, Ryan RJ McAllister, Birgit Müller, Kirill Orach, et al. A framework for mapping and comparing behavioural theories in models of social-ecological systems. *Ecological economics*, 131:21–35, 2017.
125. Maja Schlüter, L Jamila Haider, Steven J Lade, Emilie Lindkvist, Romina Martin, Kirill Orach, Nanda Wijermans, and Carl Folke. Capturing emergent phenomena in social-ecological systems. *Ecology and Society*, 24(3), 2019.
126. Jule Schulze, Birgit Müller, Jürgen Groeneveld, and Volker Grimm. Agent-based modelling of social-ecological systems: achievements, challenges, and a way forward. *Journal of Artificial Societies and Social Simulation*, 20(2), 2017.

127. Shalom H Schwartz. Normative influences on altruism. In *Advances in experimental social psychology*, volume 10, pages 221–279. Elsevier, 1977.
128. Shalom H Schwartz. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In *Advances in experimental social psychology*, volume 25, pages 1–65. Elsevier, 1992.
129. Shalom H Schwartz. An overview of the schwartz theory of basic values. *Online readings in Psychology and Culture*, 2(1):2307–0919, 2012.
130. James C Scott. *Seeing like a state*. yale university Press, 2008.
131. Olivia Serdeczny, Eleanor Waters, and Sander Chan. *Non-economic loss and damage in the context of climate change: understanding the challenges*. Number 3/2016. discussion paper, 2016.
132. Andrew M Song and Ratana Chuenpagdee. Eliciting values and principles of fishery stakeholders in south korea: a methodological exploration. *Society & natural resources*, 28(10):1075–1091, 2015.
133. Andrew M Song, Ratana Chuenpagdee, and Svein Jentoft. Values, images, and principles: What they represent and how they may improve fisheries governance. *Marine Policy*, 40:167–175, 2013.
134. Cara Steger, Shana Hirsch, Chris Cosgrove, Sarah Inman, Eric Nost, Xoco Shinbrot, Jessica PR Thorn, Daniel G Brown, Adrienne Grêt-Regamey, Birgit Müller, et al. Linking model design and application for transdisciplinary approaches in social-ecological systems. *Global Environmental Change*, 66:102201, 2021.
135. Nicoleta Stoicea, Andrew Costa, Luis Periel, Alberto Uribe, Tristan Weaver, and Sergio D Bergese. Current perspectives on the opioid crisis in the us healthcare system: a comprehensive literature review. *Medicine*, 98(20), 2019.
136. Holger Strassheim and Pekka Kettunen. When does evidence-based policy turn into policy-based evidence? configurations, contexts and mechanisms. *Evidence & Policy: A Journal of Research, Debate and Practice*, 10(2):259–277, 2014.
137. Yi Sun and Liang Cheng. A survey on agent-based modelling and equation-based modelling. *Citeseerx, Department of Computer Science, Georgia State University, Atlanta*, 2005.
138. Dan Thorman, Lorraine Whitmarsh, and Christina Demski. Policy acceptance of low-consumption governance approaches: The effect of social norms and hypocrisy. *Sustainability*, 12(3):1247, 2020.
139. Petra Tschakert, Jon Barnett, Neville Ellis, Carmen Lawrence, Nancy Tuana, Mark New, Carmen Elrick-Barr, Ram Pandit, and David Pannell. Climate change and loss, as if people mattered: values, places, and experiences. *Wiley Interdisciplinary Reviews: Climate Change*, 8(5):e476, 2017.
140. Gudmundur F Ulfarsson and Fred L Mannering. Differences in male and female injury severities in sport-utility vehicle, minivan, pickup and passenger car accidents. *Accident Analysis & Prevention*, 36(2):135–147, 2004.
141. Jeroen CJM Van Den Bergh, Ada Ferrer-i Carbonell, and Giuseppe Munda. Alternative models of individual behaviour and implications for environmental policy. *Ecological Economics*, 32(1):43–61, 2000.
142. Jeroen P Van der Sluijs, Arthur C Petersen, Peter HM Janssen, James S Risbey, and Jerome R Ravetz. Exploring the quality of evidence for complex and contested policy decisions. *Environmental Research Letters*, 3(2):024008, 2008.
143. Thomas L van der Weide. *Arguing to motivate decisions*. PhD thesis, University Utrecht, 2011.
144. Hakique N Virani and Rebecca J Haines-Saah. Drug decriminalization: A matter of justice and equity, not just health. *American journal of preventive medicine*, 58(1):161–164, 2020.
145. Karla D Wagner, L James Bovet, Bruce Haynes, Alfred Joshua, and Peter J Davidson. Training law enforcement to respond to opioid overdose with naloxone: Impact on knowledge, attitudes, and interactions with community members. *Drug and Alcohol Dependence*, 165:22–28, 2016.
146. Michael P Wellman. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems*, 30(6):1175–1189, 2016.
147. Carina Wyborn, Amber Datta, Jasper Montana, Melanie Ryan, Peat Leith, Brian Chaffin, Clark Miller, and Lorrae Van Kerkhoff. Co-producing sustainability: reordering the governance of science, policy, and practice. *Annual Review of Environment and Resources*, 44(1), 2019.

Appendix A

SIKS Dissertatiereeks

==== 2011 ====

2011-01 Botond Cseke (RUN) Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier(UU) Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

2011-03 Jan Martijn van der Werf (TUE) Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU) Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05 Base van der Raadt (VU) Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

2011-06 Yiwen Wang (TUE) Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT) Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU) BDI-based Generation of Robust Task-Oriented Dialogues

2011-09 Tim de Jong (OU) Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT) Cloud Content Contention

2011-11 Dhaval Vyas (UT) Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE) Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT) Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR) Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA) The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM) Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA) Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM) Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU) The Mind ' s Eye on Personal Profiles

2011-20 Qing Gu (VU) Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD) Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA) System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA) Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT) Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)) Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU) Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU) Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein(UVA) Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE) Discrimination-aware Classification

2011-30 Egon van den Broek (UT) Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR) Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR) Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU) Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU) Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU) Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU) Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN) Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

2011-38 Nyree Lemmens (UM) Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU) Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU) Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT) Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU) Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU) Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT) Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT) Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD) Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz(VU) Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT) Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT) Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

==== 2012 ====

2012-01 Terry Kakeeto (UvT) Relationship Marketing for SMEs in Uganda

2012-02 Muhammad Umair(VU) Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

2012-03 Adam Vanya (VU) Supporting Architecture Evolution by Mining Software Repositories

2012-04 Jurriaan Souer (UU) Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU) Maturing Interorganisational Information Systems

2012-06 Wolfgang Reinhardt (OU) Awareness Support for Knowledge Workers in Research Networks

2012-07 Rianne van Lambalgen (VU) When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

2012-08 Gerben de Vries (UVA) Kernel Methods for Vessel Trajectories

2012-09 Ricardo Neisse (UT) Trust and Privacy Management Support for Context-Aware Service Platforms

2012-10 David Smits (TUE) Towards a Generic Distributed Adaptive Hypermedia Environment

2012-11 J.C.B. Rantham Prabhakara (TUE) Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

2012-12 Kees van der Sluijs (TUE) Model Driven Design and Data Integration in Semantic Web Information Systems

2012-13 Suleman Shahid (UvT) Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

2012-14 Evgeny Knutov(TUE) Generic Adaptation Framework for Unifying Adaptive Web-based Systems

2012-15 Natalie van der Wal (VU) Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.

2012-16 Fiemke Both (VU) Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

2012-17 Amal Elgammal (UvT) Towards a Comprehensive Framework for Business Process Compliance

2012-18 Eltjo Poort (VU) Improving Solution Architecting Practices

2012-19 Helen Schonenberg (TUE) What's Next? Operational Support for Business Process Execution

2012-20 Ali Bahramisharif (RUN) Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

2012-21 Roberto Cornacchia (TUD) Querying Sparse Matrices for Information Retrieval

2012-22 Thijs Vis (UvT) Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

2012-23 Christian Muehl (UT) Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction

2012-24 Laurens van der Werff (UT) Evaluation of Noisy Transcripts for Spoken Document Retrieval

2012-25 Silja Eckartz (UT) Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application

2012-26 Emile de Maat (UVA) Making Sense of Legal Text

2012-27 Hayrettin Gurkok (UT) Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games

2012-28 Nancy Pascall (UvT) Engendering Technology Empowering Women

2012-29 Almer Tigelaar (UT) Peer-to-Peer Information Retrieval

2012-30 Alina Pommeranz (TUD) Designing Human-Centered Systems for Reflective Decision Making

2012-31 Emily Bagarukayo (RUN) A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure

2012-32 Wietske Visser (TUD) Qualitative multi-criteria preference representation and reasoning

2012-33 Rory Sie (OUN) Coalitions in Cooperation Networks (COCOON)

2012-34 Pavol Jancura (RUN) Evolutionary analysis in PPI networks and applications

2012-35 Evert Haasdijk (VU) Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics

2012-36 Denis Ssebugwawo (RUN) Analysis and Evaluation of Collaborative Modeling Processes

2012-37 Agnes Nakakawa (RUN) A Collaboration Process for Enterprise Architecture Creation

2012-38 Selmar Smit (VU) Parameter Tuning and Scientific Testing in Evolutionary Algorithms

2012-39 Hassan Fatemi (UT) Risk-aware design of value and coordination networks

2012-40 Agus Gunawan (UvT) Information Access for SMEs in Indonesia

2012-41 Sebastian Kelle (OU) Game Design Patterns for Learning

2012-42 Dominique Verpoorten (OU) Reflection Amplifiers in self-regulated Learning

2012-43 Withdrawn

2012-44 Anna Tordai (VU) On Combining Alignment Techniques

2012-45 Benedikt Kratz (UvT) A Model and Language for Business-aware Transactions

2012-46 Simon Carter (UVA) Exploration and Exploitation of Multilingual Data for Statistical Machine Translation

2012-47 Manos Tsagkias (UVA) Mining Social Media: Tracking Content and Predicting Behavior

2012-48 Jorn Bakker (TUE) Handling Abrupt Changes in Evolving Time-series Data

2012-49 Michael Kaisers (UM) Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions

2012-50 Steven van Kervel (TUD) Ontologogy driven Enterprise Information Systems Engineering

2012-51 Jeroen de Jong (TUD) Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

==== 2013 ====

2013-01 Viorel Milea (EUR) News Analytics for Financial Decision Support

2013-02 Erietta Liarou (CWI) MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing

2013-03 Szymon Klarman (VU) Reasoning with Contexts in Description Logics

2013-04 Chetan Yadati(TUD) Coordinating autonomous planning and scheduling

2013-05 Dulce Pumareja (UT) Groupware Requirements Evolutions Patterns

2013-06 Romulo Goncalves(CWI) The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience

2013-07 Giel van Lankveld (UvT) Quantifying Individual Player Differences

2013-08 Robbert-Jan Merk(VU) Making enemies: cognitive modeling for opponent agents in fighter pilot simulators

2013-09 Fabio Gori (RUN) Metagenomic Data Analysis: Computational Methods and Applications

2013-10 Jeewanie Jayasinghe Arachchige(UvT) A Unified Modeling Framework for Service Design.

2013-11 Evangelos Pournaras(TUD) Multi-level Reconfigurable Self-organization in Overlay Services

2013-12 Marian Razavian(VU) Knowledge-driven Migration to Services

2013-13 Mohammad Safiri(UT) Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly

2013-14 Jafar Tanha (UVA) Ensemble Approaches to Semi-Supervised Learning Learning

2013-15 Daniel Hennes (UM) Multiagent Learning - Dynamic Games and Applications

2013-16 Eric Kok (UU) Exploring the practical benefits of argumentation in multi-agent deliberation

2013-17 Koen Kok (VU) The PowerMatcher: Smart Coordination for the Smart Electricity Grid

2013-18 Jeroen Janssens (UvT) Outlier Selection and One-Class Classification

2013-19 Renze Steenhuizen (TUD) Coordinated Multi-Agent Planning and Scheduling

2013-20 Katja Hofmann (UvA) Fast and Reliable Online Learning to Rank for Information Retrieval

2013-21 Sander Wubben (UvT) Text-to-text generation by monolingual machine translation

2013-22 Tom Claassen (RUN) Causal Discovery and Logic

2013-23 Patricio de Alencar Silva(UvT) Value Activity Monitoring

2013-24 Haitham Bou Ammar (UM) Automated Transfer in Reinforcement Learning

2013-25 Agnieszka Anna Latoszek-Berendsen (UM) Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System

2013-26 Alireza Zarghami (UT) Architectural Support for Dynamic Homecare Service Provisioning

2013-27 Mohammad Huq (UT) Inference-based Framework Managing Data Provenance

2013-28 Frans van der Sluis (UT) When Complexity becomes Interesting: An Inquiry into the Information eXperience

2013-29 Iwan de Kok (UT) Listening Heads

2013-30 Joyce Nakatumba (TUE) Resource-Aware Business Process Management: Analysis and Support

2013-31 Dinh Khoa Nguyen (UvT) Blueprint Model and Language for Engineering Cloud Applications

2013-32 Kamakshi Rajagopal (OUN) Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development

2013-33 Qi Gao (TUD) User Modeling and Personalization in the Microblogging Sphere

2013-34 Kien Tjin-Kam-Jet (UT) Distributed Deep Web Search

2013-35 Abdallah El Ali (UvA) Minimal Mobile Human Computer Interaction Promotor: Prof. dr. L. Hardman (CWI/UVA)

2013-36 Than Lam Hoang (TUe) Pattern Mining in Data Streams

2013-37 Dirk Börner (OUN) Ambient Learning Displays

2013-38 Eelco den Heijer (VU) Autonomous Evolutionary Art

2013-39 Joop de Jong (TUD) A Method for Enterprise Ontology based Design of Enterprise Information Systems

2013-40 Pim Nijssen (UM) Monte-Carlo Tree Search for Multi-Player Games

2013-41 Jochem Liem (UVA) Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning

2013-42 Léon Planken (TUD) Algorithms for Simple Temporal Reasoning

2013-43 Marc Bron (UVA) Exploration and Contextualization through Interaction and Concepts

==== 2014 ====

2014-01 Nicola Barile (UU) Studies in Learning Monotone Models from Data

2014-02 Fiona Tuliyano (RUN) Combining System Dynamics with a Domain Modeling Method

2014-03 Sergio Raul Duarte Torres (UT) Information Retrieval for Children: Search Behavior and Solutions

2014-04 Hanna Jochmann-Mannak (UT) Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation

2014-05 Jurriaan van Reijsen (UU) Knowledge Perspectives on Advancing Dynamic Capability

2014-06 Damian Tamburri (VU) Supporting Networked Software Development

2014-07 Arya Adriansyah (TUE) Aligning Observed and Modeled Behavior

2014-08 Samur Araujo (TUD) Data Integration over Distributed and Heterogeneous Data Endpoints

2014-09 Philip Jackson (UvT) Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language

2014-10 Ivan Salvador Razo Zapata (VU) Service Value Networks

2014-11 Janneke van der Zwaan (TUD) An Empathic Virtual Buddy for Social Support

2014-12 Willem van Willigen (VU) Look Ma, No Hands: Aspects of Autonomous Vehicle Control

2014-13 Arlette van Wissen (VU) Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains

2014-14 Yangyang Shi (TUD) Language Models With Meta-information

2014-15 Natalya Mogles (VU) Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare

2014-16 Krystyna Milian (VU) Supporting trial recruitment and design by automatically interpreting eligibility criteria

2014-17 Kathrin Dentler (VU) Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability

2014-18 Mattijs Ghijsen (UVA) Methods and Models for the Design and Study of Dynamic Agent Organizations

2014-19 Vinicius Ramos (TUE) Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support

2014-20 Mena Habib (UT) Named Entity Extraction and Disambiguation for Informal Text: The Missing Link

2014-21 Kassidy Clark (TUD) Negotiation and Monitoring in Open Environments

2014-22 Marieke Peeters (UU) Personalized Educational Games - Developing agent-supported scenario-based training

2014-23 Eleftherios Sidirourgos (UvA/CWI) Space Efficient Indexes for the Big Data Era

2014-24 Davide Ceolin (VU) Trusting Semi-structured Web Data

2014-25 Martijn Lappenschaar (RUN) New network models for the analysis of disease interaction

2014-26 Tim Baarslag (TUD) What to Bid and When to Stop

2014-27 Rui Jorge Almeida (EUR) Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty

2014-28 Anna Chmielowiec (VU) Decentralized k-Clique Matching

2014-29 Jaap Kabbedijk (UU) Variability in Multi-Tenant Enterprise Software

2014-30 Peter de Cock (UvT) Anticipating Criminal Behaviour

2014-31 Leo van Moergestel (UU) Agent Technology in Agile Multiparallel Manufacturing and Product Support

2014-32 Naser Ayat (UvA) On Entity Resolution in Probabilistic Data

2014-33 Tesfa Tegegne (RUN) Service Discovery in eHealth

2014-34 Christina Manteli(VU) The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.

2014-35 Joost van Ooijen (UU) Cognitive Agents in Virtual Worlds: A Middleware Design Approach

2014-36 Joos Buijs (TUE) Flexible Evolutionary Algorithms for Mining Structured Process Models

2014-37 Maral Dadvar (UT) Experts and Machines United Against Cyberbullying

2014-38 Danny Plass-Oude Bos (UT) Making brain-computer interfaces better: improving usability through post-processing.

2014-39 Jasmina Maric (UvT) Web Communities, Immigration, and Social Capital

2014-40 Walter Omona (RUN) A Framework for Knowledge Management Using ICT in Higher Education

2014-41 Frederic Hogenboom (EUR) Automated Detection of Financial Events in News Text

2014-42 Carsten Eijckhof (CWI/TUD) Contextual Multidimensional Relevance Models

2014-43 Kevin Vlaanderen (UU) Supporting Process Improvement using Method Increments

2014-44 Paulien Meesters (UvT) Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.

2014-45 Birgit Schmitz (OUN) Mobile Games for Learning: A Pattern-Based Approach

2014-46 Ke Tao (TUD) Social Web Data Analytics: Relevance, Redundancy, Diversity

2014-47 Shangsong Liang (UVA) Fusion and Diversification in Information Retrieval

==== 2015 ====

2015-01 Niels Netten (UvA) Machine Learning for Relevance of Information in Crisis Response

2015-02 Faiza Bukhsh (UvT) Smart auditing: Innovative Compliance Checking in Customs Controls

2015-03 Twan van Laarhoven (RUN) Machine learning for network data

2015-04 Howard Spoelstra (OUN) Collaborations in Open Learning Environments

2015-05 Christoph Bösch(UT) Cryptographically Enforced Search Pattern Hiding

2015-06 Farideh Heidari (TUD) Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes

2015-07 Maria-Hendrike Peetz(UvA) Time-Aware Online Reputation Analysis

2015-08 Jie Jiang (TUD) Organizational Compliance: An agent-based model for designing and evaluating organizational interactions

2015-09 Randy Klaassen(UT) HCI Perspectives on Behavior Change Support Systems

2015-10 Henry Hermans (OUN) OpenU: design of an integrated system to support lifelong learning

2015-11 Yongming Luo(TUE) Designing algorithms for big graph datasets: A study of computing bisimulation and joins

2015-12 Julie M. Birkholz (VU) Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks

2015-13 Giuseppe Procaccianti(VU) Energy-Efficient Software

2015-14 Bart van Straalen (UT) A cognitive approach to modeling bad news conversations

2015-15 Klaas Andries de Graaf (VU) Ontology-based Software Architecture Documentation

2015-16 Changyun Wei (UT) Cognitive Coordination for Cooperative Multi-Robot Teamwork

2015-17 André van Cleeff (UT) Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs

2015-18 Holger Pirk (CWI) Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories

2015-19 Bernardo Tabuenca (OUN) Ubiquitous Technology for Lifelong Learners

2015-20 Loïs Vanhée(UU) Using Culture and Values to Support Flexible Coordination

2015-21 Sibren Fetter (OUN) Using Peer-Support to Expand and Stabilize Online Learning

015-22 Zhemin Zhu(UT) Co-occurrence Rate Networks

2015-23 Luit Gazendam (VU) Cataloguer Support in Cultural Heritage

2015-24 Richard Berendsen (UVA) Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation

2015-25 Steven Woudenberg (UU) Bayesian Tools for Early Disease Detection

2015-26 Alexander Hogenboom (EUR) Sentiment Analysis of Text Guided by Semantics and Structure

2015-27 Sándor Héman (CWI) Updating compressed colomn stores

2015-28 Janet Bagorogoza(TiU) KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO

2015-29 Hendrik Baier (UM) Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains

2015-30 Kiavash Bahreini(OU) Real-time Multimodal Emotion Recognition in E-Learning

2015-31 Yakup Koç (TUD) On the robustness of Power Grids

2015-32 Jerome Gard(UL) Corporate Venture Management in SMEs

2015-33 Frederik Schadd (TUD) Ontology Mapping with Auxiliary Resources

2015-34 Victor de Graaf(UT) Gesocial Recommender Systems

2015-35 Jungxao Xu (TUD) Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

==== 2016 ====

2016-01 Syed Saiden Abbas (RUN) Recognition of Shapes by Humans and Machines

2016-02 Michiel Christiaan Meulendijk (UU) Optimizing medication reviews through decision support: prescribing a better pill to swallow

2016-03 Maya Sappelli (RUN) Knowledge Work in Context: User Centered Knowledge Worker Support

2016-04 Laurens Rietveld (VU) Publishing and Consuming Linked Data

2016-05 Evgeny Sherkhonov (UVA) Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers

2016-06 Michel Wilson (TUD) Robust scheduling in an uncertain environment

2016-07 Jeroen de Man (VU) Measuring and modeling negative emotions for virtual training

2016-08 Matje van de Camp (TiU) A Link to the Past: Constructing Historical Social Networks from Unstructured Data

2016-09 Archana Nottamkandath (VU) Trusting Crowdsourced Information on Cultural Artefacts

2016-10 George Karafotias (VUA) Parameter Control for Evolutionary Algorithms

2016-11 Anne Schuth (UVA) Search Engines that Learn from Their Users

2016-12 Max Knobbout (UU) Logics for Modelling and Verifying Normative Multi-Agent Systems

2016-13 Nana Baah Gyan (VU) The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach

2016-14 Ravi Khadka (UU) Revisiting Legacy Software System Modernization

2016-15 Steffen Michels (RUN) Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments

2016-16 Guangliang Li (UVA) Socially Intelligent Autonomous Agents that Learn from Human Reward

2016-17 Berend Weel (VU) Towards Embodied Evolution of Robot Organisms

2016-18 Albert Meroño Peñuela (VU) Refining Statistical Data on the Web

2016-19 Julia Efremova (Tu/e) Mining Social Structures from Genealogical Data

2016-20 Daan Odijk (UVA) Context & Semantics in News & Web Search

2016-21 Alejandro Moreno Célleri (UT) From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground

2016-22 Grace Lewis (VU) Software Architecture Strategies for Cyber-Foraging Systems

2016-23 Fei Cai (UVA) Query Auto Completion in Information Retrieval

2016-24 Brend Wanders (UT) Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach

2016-25 Julia Kiseleva (TU/e) Using Contextual Information to Understand Searching and Browsing Behavior

2016-26 Dilhan Thilakarathne (VU) In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains

2016-27 Wen Li (TUD) Understanding Geo-spatial Information on Social Media

2016-28 Mingxin Zhang (TUD) Large-scale Agent-based Social Simulation - A study on epidemic prediction and control

2016-29 Nicolas Höning (TUD) Peak reduction in decentralised electricity systems -Markets and prices for flexible planning

2016-30 Ruud Mattheij (UvT) The Eyes Have It

2016-31 Mohammad Khelghati (UT) Deep web content monitoring

2016-32 Eelco Vriezekolk (UT) Assessing Telecommunication Service Availability Risks for Crisis Organisations

2016-33 Peter Bloem (UVA) Single Sample Statistics, exercises in learning from just one example

2016-34 Dennis Schunselaar (TUE) Configurable Process Trees: Elicitation, Analysis, and Enactment

2016-35 Zhaochun Ren (UVA) Monitoring Social Media: Summarization, Classification and Recommendation

2016-36 Daphne Karreman (UT) Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies

2016-37 Giovanni Sileno (UvA) Aligning Law and Action - a conceptual and computational inquiry

2016-38 Andrea Minuto (UT) MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design

2016-39 Merijn Bruijnes (UT) Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect

2016-40 Christian Detweiler (TUD) Accounting for Values in Design

2016-41 Thomas King (TUD) Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

2016-42 Spyros Martzoukos (UVA) Combinatorial and Compositional Aspects of Bilingual Aligned Corpora

2016-43 Saskia Koldijk (RUN) Context-Aware Support for Stress Self-Management: From Theory to Practice

2016-44 Thibault Sellam (UVA) Automatic Assistants for Database Exploration

2016-45 Bram van de Laar (UT) Experiencing Brain-Computer Interface Control

2016-46 Jorge Gallego Perez (UT) Robots to Make you Happy

2016-47 Christina Weber (UL) Real-time foresight - Preparedness for dynamic innovation networks

2016-48 Tanja Buttler (TUD) Collecting Lessons Learned

2016-49 Gleb Polevoy (TUD) Participation and Interaction in Projects. A Game-Theoretic Analysis

2016-50 Yan Wang (UVT) The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

==== 2017 ====

2017-01 Jan-Jaap Oerlemans (UL) Investigating Cybercrime

2017-02 Sjoerd Timmer (UU) Designing and Understanding Forensic Bayesian Networks using Argumentation

2017-03 Daniël Harold Telgen (UU) Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines

2017-04 Mrunal Gawade (CWI) MULTI-CORE PARALLELISM IN A COLUMN-STORE

2017-05 Mahdieh Shadi (UVA) Collaboration Behavior

2017-06 Damir Vandic (EUR) Intelligent Information Systems for Web Product Search

2017-07 Roel Bertens (UU) Insight in Information: from Abstract to Anomaly

2017-08 Rob Konijn (VU) Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery

2017-09 Dong Nguyen (UT) Text as Social and Cultural Data: A Computational Perspective on Variation in Text

2017-10 Robby van Delden (UT) (Steering) Interactive Play Behavior

2017-11 Florian Kunneman (RUN) Modelling patterns of time and emotion in Twitter #anticipointment

2017-12 Sander Leemans (TUE) Robust Process Mining with Guarantees

2017-13 Gijs Huisman (UT) Social Touch Technology - Extending the reach of social touch through haptic technology

2017-14 Shoshannah Tekofsky (UvT) You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior

2017-15 Peter Berck, Radboud University (RUN) Memory-Based Text Correction

2017-16 Aleksandr Chuklin (UVA) Understanding and Modeling Users of Modern Search Engines

2017-17 Daniel Dimov (UL) Crowdsourced Online Dispute Resolution

2017-18 Ridho Reinanda (UVA) Entity Associations for Search

2017-19 Jeroen Vuurens (TUD) Proximity of Terms, Texts and Semantic Vectors in Information Retrieval

2017-20 Mohammadbashir Sedighi (TUD) Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

2017-21 Jeroen Linssen (UT) Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)

2017-22 Sara Magliacane (VU) Logics for causal inference under uncertainty

2017-23 David Graus (UVA) Entities of Interest— Discovery in Digital Traces

2017-24 Chang Wang (TUD) Use of Affordances for Efficient Robot Learning

2017-25 Veruska Zamborlini (VU) Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search

2017-26 Merel Jung (UT) Socially intelligent robots that understand and respond to human touch

2017-27 Michiel Joosse (UT) Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors

2017-28 John Klein (VU) Architecture Practices for Complex Contexts

2017-29 Adel Alhuraibi (UVT) From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT

2017-30 Wilma Latuny (UVT) The Power of Facial Expressions

2017-31 Ben Ruijl (UL) Advances in computational methods for QFT calculations

2017-32 Thaer Samar (RUN) Access to and Retrievability of Content in Web Archives

2017-33 Brigit van Loggem (OU) Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity

2017-34 Maren Scheffel (OUN) The Evaluation Framework for Learning Analytics

2017-35 Martine de Vos (VU) Interpreting natural science spreadsheets

2017-36 Yuanhao Guo (UL) Shape Analysis for Phenotype Characterisation from High-throughput Imaging

2017-37 Alejandro Montes García (TUE) WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy

2017-38 Alex Kayal (TUD) Normative Social Applications

2017-39 Sara Ahmadi (RUN) Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR

2017-40 Altaf Hussain Abro (VUA) Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems"

2017-41 Adnan Manzoor (VUA) Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle

2017-42 Elena Sokolova (RUN) Causal discovery from mixed and missing data with applications on ADHD datasets

2017-43 Maaike de Boer (RUN) Semantic Mapping in Video Retrieval

2017-44 Garm Lucassen (UU) Understanding User Stories - Computational Linguistics in Agile Requirements Engineering

2017-45 Bas Testerink (UU) Decentralized Runtime Norm Enforcement

2017-46 Jan Schneider (OU) Sensor-based Learning Support

2017-47 Yie Yang (TUD) Crowd Knowledge Creation Acceleration

2017-48 Angel Suarez (OU) Collaborative inquiry-based learning

==== 2018 ====

2018-01 Han van der Aa (VUA) Comparing and Aligning Process Representations

2018-02 Felix Mannhardt (TUE) Multi-perspective Process Mining

2018-03 Steven Bosems (UT) Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction

2018-04 Jordan Janeiro (TUD) Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks

2018-05 Hugo Huurdeman (UVA) Supporting the Complex Dynamics of the Information Seeking Process

2018-06 Dan Ionita (UT) Model-Driven Information Security Risk Assessment of Socio-Technical Systems

2018-07 Jieting Luo (UU) A formal account of opportunism in multi-agent systems

2018-08 Rick Smetsers (RUN) Advances in Model Learning for Software Systems

2018-09 Xu Xie (TUD) Data Assimilation in Discrete Event Simulations

2018-10 Julienka Mollee (VUA) Moving forward: supporting physical activity behavior change through intelligent technology

2018-11 Mahdi Sargolzaei (UVA) Enabling Framework for Service-oriented Collaborative Networks

2018-12 Xixi Lu (TUE) Using behavioral context in process mining

2018-13 Seyed Amin Tabatabaei (VUA) Using behavioral context in process mining: Exploring the added value of computational models for increasing the use of renewable energy in the residential sector

2018-14 Bart Joosten (UVT) Detecting Social Signals with Spatiotemporal Gabor Filters

2018-15 Naser Davarzani (UM) Biomarker discovery in heart failure

2018-16 Jaebok Kim (UT) Automatic recognition of engagement and emotion in a group of children

2018-17 Jianpeng Zhang (TUE) On Graph Sample Clustering

2018-18 Henriette Nakad (UL) De Notaris en Private Rechtspraak

2018-19 Minh Duc Pham (VUA) Emergent relational schemas for RDF

2018-20 Manxia Liu (RUN) Time and Bayesian Networks

2018-21 Aad Slootmaker (OUN) EMERGO: a generic platform for authoring and playing scenario-based serious games

2018-22 Eric Fernandes de Mello Araújo (VUA) Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks

2018-23 Kim Schouten (EUR) Semantics-driven Aspect-Based Sentiment Analysis

2018-24 Jered Vroon (UT) Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots

2018-25 Riste Gligorov (VUA) Serious Games in Audio-Visual Collections

2018-26 Roelof de Vries (UT) Theory-Based And Tailor-Made: Motivational Messages for Behavior Change Technology

2018-27 Maikel Leemans (TUE) Hierarchical Process Mining for Scalable Software Analysis

2018-28 Christian Willemse (UT) Social Touch Technologies: How they feel and how they make you feel

2018-29 Yu Gu (UVT) Emotion Recognition from Mandarin Speech

2018-30 Wouter Beek (VU) The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

==== 2019 ====

2019-01 Rob van Eijk (UL) Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification

2019-02 Emmanuelle Beauxis- Aussalet (CWI, UU) Statistics and Visualizations for Assessing Class Size Uncertainty

2019-03 Eduardo Gonzalez Lopez de Murillas (TUE) Process Mining on Databases: Extracting Event Data from Real Life Data Sources

2019-04 Ridho Rahmadi (RUN) Finding stable causal structures from clinical data

2019-05 Sebastiaan van Zelst (TUE) Process Mining with Streaming Data

2019-06 Chris Dijkshoorn (VU) Nichesourcing for Improving Access to Linked Cultural Heritage Datasets

2019-07 Soude Fazeli (TUD) Recommender Systems in Social Learning Platforms

2019-08 Frits de Nijs (TUD) Resource-constrained Multi-agent Markov Decision Processes

2019-09 Fahimeh Alizadeh Moghaddam (UVA) Self-adaptation for energy efficiency in software systems

2019-10 Qing Chuan Ye (EUR) Multi-objective Optimization Methods for Allocation and Prediction

2019-11 Yue Zhao (TUD) Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs

2019-12 Jacqueline Heinerman (VU) Better Together

2019-13 Guanliang Chen (TUD) MOOC Analytics: Learner Modeling and Content Generation

2019-14 Daniel Davis (TUD) Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses

2019-15 Erwin Walraven (TUD) Planning under Uncertainty in Constrained and Partially Observable Environments

2019-16 Guangming Li (TUE) Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models

2019-17 Ali Hurriyetoglu (RUN) Extracting actionable information from microtexts

2019-18 Gerard Wagenaar (UU) Artefacts in Agile Team Communication

2019-19 Vincent Koeman (TUD) Tools for Developing Cognitive Agents

2019-20 Chide Groenouwe (UU) Fostering technically augmented human collective intelligence

2019-21 Cong Liu (TUE) Software Data Analytics: Architectural Model Discovery and Design Pattern Detection

2019-22 Martin van den Berg (VU) Improving IT Decisions with Enterprise Architecture

2019-23 Qin Liu (TUD) Intelligent Control Systems: Learning, Interpreting, Verification

2019-24 Anca Dumitrache (VU) Truth in Disagreement- Crowdsourcing Labeled Data for Natural Language Processing

2019-25 Emiel van Miltenburg (UVT) Pragmatic factors in (automatic) image description

2019-26 Prince Singh (UT) An Integration Platform for Synchromodal Transport

2019-27 Alessandra Antonaci (OUN) The Gamification Design Process applied to (Massive) Open Online Courses

2019-28 Esther Kuindersma (UL) Cleared for take-off: Game-based learning to prepare airline pilots for critical situations

2019-29 Daniel Formolo (VU) Using virtual agents for simulation and training of social skills in safety-critical circumstances

2019-30 Vahid Yazdanpanah (UT) Multiagent Industrial Symbiosis Systems

2019-31 Milan Jelisavcic (VUA) Alive and Kicking: Baby Steps in Robotics

2019-32 Chiara Sironi (UM) Monte-Carlo Tree Search for Artificial General Intelligence in Games

2019-33 Anil Yaman (TUE) Evolution of Biologically Inspired Learning in Artificial Neural Networks

2019-34 Negar Ahmadi (TUE) EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES

2019-35 Lisa Facey-Shaw (OUN) Gamification with digital badges in learning programming

2019-36 Kevin Ackermans (OUN) Designing Video-Enhanced Rubrics to Master Complex Skills

2019-37 Jian Fang (TUD) Database Acceleration on FPGAs

2019-38 Akos Kadar (OUN) Learning visually grounded and multilingual representations

==== 2020 ====

2020-01 Armon Toubman (UL) Calculated Moves: Generating Air Combat Behaviour

2020-02 Marcos de Paula Bueno (UL) Unraveling Temporal Processes using Probabilistic Graphical Models

2020-03 Mostafa Deghani (UvA) Learning with Imperfect Supervision for Language Understanding

2020-04 Maarten van Gompel (RUN) Context as Linguistic Bridges

2020-05 Yulong Pei (TUE) On local and global structure mining

2020-06 Preethu Rose Anish (UT) Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support

2020-07 Wim van der Vegt (OUN) Towards a software architecture for reusable game components

2020-08 Ali Mirsoleimani (UL) Structured Parallel Programming for Monte Carlo Tree Search

2020-09 Myriam Traub (UU) Measuring Tool Bias & Improving Data Quality for Digital Humanities Research

2020-10 Alifah Syamsiyah (TUE) In-database Preprocessing for Process Mining

2020-11 Sepideh Mesbah (TUD) Semantic-Enhanced Training Data AugmentationMethods for Long-Tail Entity Recognition Models

2020-12 Ward van Breda (VU) Predictive Modeling in E-Mental Health: Exploring Applicability in Personalised Depression Treatment

2020-13 Marco Virgolin (CWI) Design and Application of Gene-pool Optimal Mixing Evolutionary Algorithms for Genetic Programming

2020-14 Mark Raasveldt (CWI/UL) Integrating Analytics with Relational Databases

2020-15 Konstantinos Georgiadis (OU) Smart CAT: Machine Learning for Configurable Assessments in Serious Games

2020-16 Ilona Wilmont (RUN) Cognitive Aspects of Conceptual Modelling

2020-17 Daniele Di Mitri (OU) The Multimodal Tutor: Adaptive Feedback from Multimodal Experiences

2020-18 Georgios Methenitis (TUD) Agent Interactions & Mechanisms in Markets with Uncertainties: Electricity Markets in Renewable Energy Systems

2020-19 Guido van Capelleveen (UT) Industrial Symbiosis Recommender Systems

2020-20 Albert Hankel (VU) Embedding Green ICT Maturity in Organisations

2020-21 Karine da Silva Miras de Araujo (VU) Where is the robot?: Life as it could be

2020-22 Maryam Masoud Khamis (RUN) Understanding complex systems implementation through a modeling approach: the case of e-government in Zanzibar

2020-23 Rianne Conijn (UT) The Keys to Writing: A writing analytics approach to studying writing processes using keystroke logging

2020-24 Lenin da Nobrega Medeiros (VUA/RUN) How are you feeling, human? Towards emotionally supportive chatbots

2020-25 Xin Du (TUE) The Uncertainty in Exceptional Model Mining

2020-26 Krzysztof Leszek Sadowski (UU) GAMBIT: Genetic Algorithm for Model-Based mixed-Integer op-Timization

2020-27 Ekaterina Muravyeva (TUD) Personal data and informed consent in an educational context

2020-28 Bibeg Limbu (TUD) Multimodal interaction for deliberate practice: Training complex skills with augmented reality

2020-29 Ioan Gabriel Bucur (RUN) Being Bayesian about Causal Inference

2020-30 Bob Zadok Blok (UL) Creatief, Creatieve, Creatiefst

2020-31 Gongjin Lan (VU) Learning better – From Baby to Better

2020-32 Jason Rhuggenaath (TUE) Revenue management in online markets: pricing and online advertising

2020-33 Rick Gilsing (TUE) Supporting service-dominant business model evaluation in the context of business model innovation

2020-34 Anna Bon (MU) Intervention or Collaboration? Redesigning Information and Communication Technologies for Development

2020-35 Siamak Farshidi (UU) Multi-Criteria Decision-Making in Software Production

==== 2021 ====

2021-01 Francisco Xavier Dos Santos Fonseca (TUD) Location-based Games for Social Interaction in Public Space

2021-02 Rijk Mercuur (TUD) Simulating Human Routines: Integrating Social Practice Theory in Agent-Based Models

2021-03 Seyyed Hadi Hashemi (UVA) Modeling Users Interacting with Smart Devices

2021-04 Ioana Jivet (OU) The Dashboard That Loved Me: Designing adaptive learning analytics for self-regulated learning

2021-05 Davide Dell'Anna (UU) Data-Driven Supervision of Autonomous Systems

2021-06 Daniel Davison (UT) "Hey robot, what do you think?" How children learn with a social robot

2021-07 Armel Lefebvre (UU) Research data management for open science

2021-08 Nardie Fanchamps (OU) The Influence of Sense-Reason-Act Programming on Computational Thinking

2021-09 Cristina Zaga (UT) The Design of Robothings. Non-Anthropomorphic and Non-Verbal Robots to Promote Childrens Collaboration Through Play

2021-10 Quinten Meertens (UvA) Misclassification Bias in Statistical Learning

2021-11 Anne van Rossum (UL) Nonparametric Bayesian Methods in Robotic Vision

2021-12 Lei Pi (UL) External Knowledge Absorption in Chinese SMEs

2021-13 Bob R. Schadenberg (UT) Robots for Autistic Children: Understanding and Facilitating Predictability for Engagement in Learning

2021-14 Negin Samaeemofrad (UL) Business Incubators: The Impact of Their Support

2021-15 Onat Ege Adali (TU/e) Transformation of Value Propositions into Resource Re-Configurations through the Business Services Paradigm

2021-16 Esam A. H. Ghaleb (MU) BIMODAL EMOTION RECOGNITION FROM AUDIO-VISUAL CUES

2021-17 Dario Dotti (UM) Human Behavior Understanding from motion and bodily cues using deep neural networks

2021-18 Remi Wieten (UU) Bridging the Gap Between Informal Sense-Making Tools and Formal Systems - Facilitating the Construction of Bayesian Networks and Argumentation Frameworks

2021-19 Roberto Verdecchia (VU) Architectural Technical Debt: Identification and Management

2021-20 Masoud Mansoury (TU/e) Understanding and Mitigating Multi-Sided Exposure Bias in Recommender Systems

2021-21 Pedro Thiago Timbó Holanda (CWI) Progressive Indexes

2021-22 Sihang Qiu (TUD) Conversational Crowdsourcing

2021-23 Hugo Manuel Proença (LIACS) Robust rules for prediction and description

2021-24 Kaijie Zhu (TUE) On Efficient Temporal Subgraph Query Processing

2021-25 Eoin Martino Grua (VUA) The Future of E-Health is Mobile: Combining AI and Self-Adaptation to Create Adaptive E-Health Mobile Applications

2021-26 Benno Kruit (CWI & VU) Reading the Grid: Extending Knowledge Bases from Human-readable Tables

2021-27 Jelte van Waterschoot (UT) Personalized and Personal Conversations: Designing Agents Who Want to Connect With You

2021-28 Christoph Selig (UL) Understanding the Heterogeneity of Corporate Entrepreneurship Programs

==== 2022 ====

2022-1 Judith van Stegeren (UT) Flavor text generation for role-playing video games

2022-2 Paulo da Costa (TU/e) Data-driven Prognostics and Logistics Optimisation: A Deep Learning Journey

2022-3 Ali el Hassouni (VUA) A Model A Day Keeps The Doctor Away: Reinforcement Learning For Personalized Healthcare

2022-4 Ünal Aksu (UU) A Cross-Organizational Process Mining Framework

2022-5 Shiwei Liu (TU/e) Sparse Neural Network Training with In-Time Over-Parameterization

2022-6 Reza Refaei Afshar (TU/e) Machine Learning for Ad Publishers in Real Time Bidding

2022-7 Sambit Praharaj (OU) Measuring the Unmeasurable? Towards Automatic Co-located Collaboration Analytics

2022-8 Maikel L. van Eck (TU/e) Process Mining for Smart Product Design

2022-9 Oana Andreea Inel (VUA) Understanding Events: A Diversity-driven Human-Machine Approach

2022-10 Felipe Moraes Gomes (TUD) Examining the Effectiveness of Collaborative Search Engines

2022-11 Mirjam de Haas (UT) Staying engaged in child-robot interaction, a quantitative approach to studying preschoolers engagement with robots and tasks during second-language tutoring

2022-12 Guanyi Chen (UU) Computational Generation of Chinese Noun Phrases

2022-13 Xander Wilcke (VUA) Machine Learning on Multimodal Knowledge Graphs: Opportunities, Challenges, and Methods for Learning on Real-World Heterogeneous and Spatially-Oriented

2022-14 Michiel Overeem (UU) Evolution of Low-Code Platforms

2022-15 Jelmer Jan Koorn (UU) Work in Process: Unearthing Meaning using Process Mining

2022-16 Pieter Gijsbers (TU/e) Systems for AutoML Research

2022-17 Laura van der Lubbe (VUA) Empowering vulnerable people with serious games and gamification

2022-18 Paris Mavromoustakos Blom (TiU) Player Affect Modelling and Video Game Personalisation

2022-19 Bilge Yigit Ozkan (UU) Cybersecurity Maturity Assessment and Standardisation

2022-20 Fakhra Jabeen (VUA) Dark Side of the Digital Media - Computational Analysis of Negative Human Behaviors on Social Media

2022-21 Seethu Mariyam Christopher (UM) Intelligent Toys for Physical and Cognitive Assessments

2022-22 Alexandra Sierra Rativa (TiU) Virtual Character Design and its potential to foster Empathy, Immersion, and Collaboration Skills in Video Games and Virtual Reality Simulations

2022-23 Ilir Kola (TUD) Enabling Social Situation Awareness in Support Agents

2022-24 Samaneh Heidari (UU) Agents with Social Norms and Values - A framework for agent based social simulations with social norms and personal values

2022-25 Anna L.D. Latour (LU) Optimal decision-making under constraints and uncertainty

2022-26 Anne Dirkson (LU) Knowledge Discovery from Patient Forums: Gaining novel medical insights from patient experiences

2022-27 Christos Athanasiadis (UM) Emotion-aware cross-modal domain adaptation in video sequences

2022-28 Onuralp Ulusoy (UU) Privacy in Collaborative Systems

2022-29 Jan Kolkmeier (UT) From Head Transform to Mind Transplant: Social Interactions in Mixed Reality

2022-30 Dean De Leo (CWI) Analysis of Dynamic Graphs on Sparse Arrays

2022-31 Konstantinos Traganos (TU/e) Tackling Complexity in Smart Manufacturing with Advanced Manufacturing Process Management

2022-32 Cezara Pastrav (UU) Social simulation for socio-ecological systems

2022-33 Brinn Hekkelman (TUD/CWI) Fair Mechanisms for Smart Grid Congestion Management

2022-34 Nimat Ullah (VUA) Mind Your Behaviour: Computational Modelling of Emotion & Desire Regulation for Behaviour Change