Doctrine and Ethics Compliant Autonomy Using An Ontological Framework

Donald P. Brutzman[1], Curtis L. Blais[1], Hsin-Fu Wu[2], Richard Markeloff[*3] Carl Andersen[3]

[1]Naval Postgraduate School
[2]Raytheon Corp.
[3]Raytheon BBN Technologies
brutzman@nps.edu

Abstract

Ensuring ethical robot behavior requires complex representations and methodologies designed to guarantee it. Our approach extends frameworks already used by the U.S. military to ensure human ethical and doctrinal behavior by human beings. These have built in advantages of being able to express complex plans and constraints, yet remaining intelligible to humans, a requirement for ethical responsibility and liability. To extend the framework to machines, mission constructs are expressed using an Autonomous Vehicle Command Language (AVCL) expressing mission actions and outcomes that can readily be translated to runnable source code in several programming languages. Missions written in AVCL can be validated via translation to an RDF/OWL Mission Execution Ontology (MEO) supporting queried proofs of ethical correctness. MEO ensures that missions are both semantically valid and compliant with ethical constraints. These technologies implement a simulation, testing, and certification regime that can serve as a foundation for human authority over and trust in robots capable of lethal force.

Robotic agent capabilities are evolving rapidly, complicating the challenge of controlling them. In particular, the problem of maintaining human control of robots similar to that exercised over human armed forces members is complicated by robot inability to understand and execute natural language directives and constraints. A number of authors express skepticism that ethical robots can even be achieved. (Forbes.com 2019) However, we offer evidence that recent development of action languages intelligible to both humans and robots makes it possible to achieve these control-oriented goals.

Challenges and Criteria for Ethically Compliant Systems

Our work adapts frameworks for human ethical responsibility used successfully in collaborative military operations, even across varying human cultures and platforms. These frameworks are rooted in international humanitarian law (Wikipedia 2021a) and are frequently manifested in military training and operations as Rules of Engagement (ROE) (United States Marine Corps Post 2005; Wikipedia 2021b) that "define the circumstances, conditions, degree, and manner in which the use of force, or actions which might be construed as provocative, may be applied." U.S. armed forces are trained in ROE and associated ethics doctrines that emphasize ultimate human responsibility for the actions of military organizations of human agents. This emphasis on human beings as controlling ethical actors is explicitly preserved in U.S. doctrine about robot autonomy (U.S. Department of Defense 2012). We identify implicit requirements of these frameworks, transferred to a human-robot context.

- **Predictability**. Robot control methods and associated development methodologies must be sufficiently reliable to enable prediction of robot behavior in any situation.
- **Authority**. Robot control must support ultimate (indirect) control by qualified, well-informed humans over robot outcomes.
- **Responsibility**. Because only human beings can adopt moral responsibility, any robotic failures must be traceable back to a specific human entity (e.g. programmers, manufacturers, operators, leadership).
- **Liability**. Liability assignment (whether legal or moral) requires that parties involved in robotic development and employment can reasonably foresee outcomes for which they are responsible.

The question of which autonomy approaches might satisfy the above requirements is a subject of active research. Research and operational experience by the authors over the past two decades has informed our own conceptualization of how the problem must be solved. Some top-level, recurring criteria include:

- **Expressiveness**. A robot control framework must be able to express real world missions at a useful level of detail, including actions and goals, decision criteria, sequencing, and constraints upon behavior. Actions and goals should be decomposable into smaller elements to mirror human cognitive methods of dealing with complexity.
- **Relevance and Intelligibility**. Ideally, the framework should smoothly integrate with existing military and civil frameworks for control and decision-making for human

[*]We dedicate the present paper to the late Dr. Richard Markeloff and thank him for his insightful guidance of this work, especially regarding future lines of investigation.

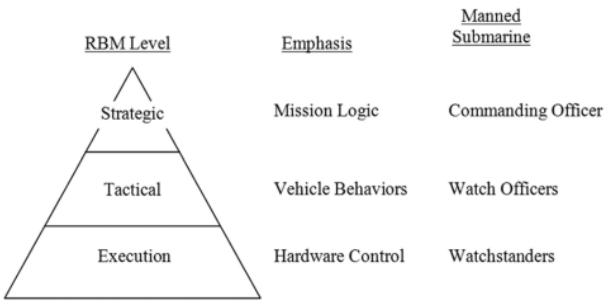

Figure 1: The RBM software architecture is based on the hierarchical control paradigm employed in naval vessels.

beings. This implies that all humans developing or using the robot system must understand the framework and how their own choices contribute to its behavior.

- **Provability & Tractability**. To ensure robot control and predictability, the framework should support *proofs that robots will behave ethically, and particularly proofs of robot compliance with behavioral constraints. This criterion creates a tension with Expressiveness, as powerful, expressive mission representations may not be easily amenable to proof.*
- ***Developmental Usability***. *To ensure wide applicability, the framework must be easily used by developers of real world robot control systems, across platforms and languages.*

The RBM Architecture

Over years of practice, the authors developed a number of formalisms that meet all the above criteria. As a conceptual framework, we developed a three-level control architecture, the Rational Behavior Model (Byrnes et al. 1996), shown in Figure 1, that applies roles and tasks familiar to manned ships and aircraft to a robot context.

- Execution is the lowest control level, analogous to junior human crew members executing atomic commands (e.g. "Left rudder 30 degrees"). For robots, execution involves control and management of hardware systems that directly interact with the vehicle's physical environment.
- Tactical control directs execution level functions to achieve more complex, but largely scripted tasks such as directing a vessel or conducting an area search or mapping.
- Strategic control, in turn, oversees tactical behaviors and corresponds to manned vessel command. Most ethical control issues are usefully represented and addressed at this level.

This division of roles is familiar to most military personnel and also supports widely used control theories in military operations such as OODA & SDA. These roles also help conceptualize how failures might occur and how responsibility for them might be traced back to specific human system developers or users. The authors' work focuses on

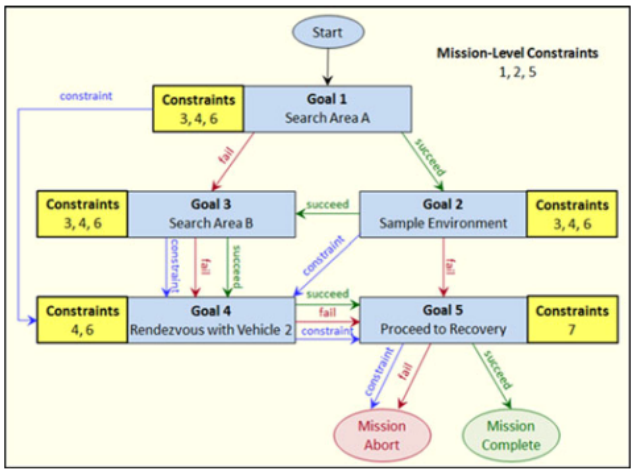

Figure 2: MEA mission-flow graph for a search and sample mission with ternary branching for imminent ethical-constraint violations.

Strategic level decision making as the locus of familiar ethical reasoning patterns, but RBM highlights that Execution and Tactical-level tasks must also be represented and implemented for higher level reasoning to function correctly.

Modeling Using AVCL & MEA

The authors' ideas about mission representations, languages and expressiveness have evolved over many years of experimentation with robot control, including sea trials with the Phoenix (Brutzman 1994) and Aries (Brutzman et al. 2013) Autonomous Underwater Vehicles (AUVs). Their work has coalesced to using Finite State Machines (FSMs) with decomposable States as an expressive and intelligible framework that is still amenable to proof.

Mission Execution Automata (MEA) (McGhee, Brutzman, and Davis 2012) are FSMs with several important properties supporting control criteria. MEAs are process flow graphs in which each Goal state embodies a task or process extended in time; Goals may be decomposed into subgraphs to achieve greater modeling fidelity. Figure 2 shows an example of a mission expressed in MEA. An important MEA innovation is the use of three distinct transition types between Goals: succeed, fail, and constraint. Succeed transitions typically leads to a next Goal in the larger process. Most real world Goal failures (e.g an engine failing to ignite) are modeled without any transition from the current Goal, which is stil underway, The fail transition models only a final, irrecoverable failure, often prompting mission abort or recovery. The constraint transition models recognition of the agent that the current goal is incompatible with current ethical constraints. An insight of MEA is that constraint failures should be represented distinctly from general failures because they often transition to a different Goal.

A recurring concern for the authors is support for proof of ethical behavior. Earlier work restricted the use of loops in MEAs because of difficulties deriving proofs; more recent work relaxes the restriction provided timeout conditions are

added to ensure mission completion. By intention, lower-level Goal pre- and post- conditions are represented only implicitly in the model and must be described externally using text and diagrams for use by developers nd commanders.

A major advantage of MEAs is that they are amenable to formal proof, software development, and even machine understanding, yet are also easily expressed graphically in charts and graphical user interfaces for human understanding. For formal MEA representation, the authors adapted their own Autonomous Vehicle Command Language (AVCL), a schema-constrained XML data model supporting autonomous vehicle mission definition, execution, and management (Davis, Blais, and Brutzman 2006). The authors developed a simulation and graphical display environment, AUV Workbench (Weekley et al. 2004), for testing of AVCL missions. The authors have also implemented translators from AVCL to leading autonomy development languages (Java, Lisp, Prolog) as well as presentation languages (HTML5,KML, X3D). The former effort supports the Developmental Usability criterion by easing the process of developing actual autonomy applications that comply with the MEA framework.

MEA/AVCL Translation also supports the Provability criterion by creating formal models in languages amenable to validation, or proofs of coherent and ethical behavior. Advantage of MEA as an representation of strategic-level missions whose behavior can be guaranteed given certain assumptions. Prove goals can be fulfilled, missions can be finite, constraints can be adhered to.

Validation Using MEO

The authors work on validation has also evolved, from initial efforts using Prolog (eventually rejected as potentially undecidable) to use of dialects of the OWL language. Using the OWL ontology language(Group et al. 2009), the authors created the Mission Execution Ontology (MEO) (Brutzman et al. 2017, 2018; Brutzman, Blais, and Wu 2020) , which defines classes for all relevant mission concepts. Here, missions are defined as graphs of inter-related OWL class instances, shown in Figure 3. OWL includes class and property restrictions, expressed as logical formulas, that enforce MEA behaviors. Selected MEO restriction formulas are shown in Figure 4.

A crucial MEO feature is its use of a Vehicle class and associated Vehicle capabilities towards Goals and Constraints. Specifically, the Vehicle *canExecute* property expresses that the Vehicle can complete a particular Goal, while the Vehicle *canIdentify* property expresses that a Vehicle can recognize an imminent potential violation of some ethical, doctrinal, or command Constraint. These relationships allow the modeling of complex, multi-Vehicle missions with varying Vehicle capabilities as well as contingency plans for responding to looming constraint violations. MEO's representations naturally support the integration of human oversight into a larger mission: for example, constraints can force consultation with a human commander before engaging in lethal force.

With the above relationships, MEO supports a variety of logical validation proofs about defined Missions. First, MEO

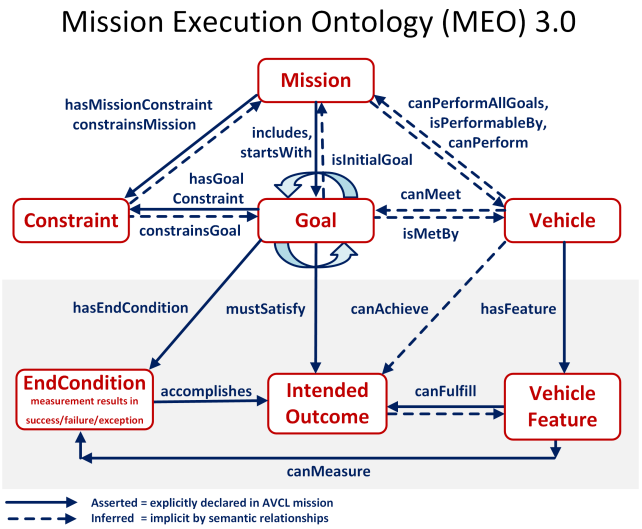

Figure 3: MEO classes and role predicates expressing the MEA mission model structure.

can validate that proposed Vehicles are capable of completing mission Goals. Importantly, MEO can validate that no mission execution exists in which robot Vehicles fail to comply with Constraints, and can even verify that at least some potential executions might achieve the mission Goals.

An important feature of the combined AVCL/MEA/MEO framework is that it abstracts complex aspects of the autonomy problem. Goal pre- and post- conditions and conditions determining Goal success or failure are represented only implicitly in the model. For real world mission executions to mirror an abstracted MEO execution, these details would need to be described externally using text and diagrams for use by developers and commanders. Constraint semantics, including conditions indicating imminent violation, are similarly abstracted away, and could be developed independently of the model. Alternatively, future work could extend MEO with Tactical- and Execution- level primitives that model Goal and Constraint dynamics in more detail.

Two central design decisions in OWL semantics complicate the use of MEO in mission modeling. First, OWL uses an *open-world* semantics that assumes some world facts may not be known. As a result, open-world OWL restrictions intended to constrain or bound the world sometimes instead add to it by inferring previously unknown entities. Second, OWL's lack of a *unique-names* assumption sometimes causes unexpected inference that two model entities are actually the same entity. The authors are currently exploring the use of SHACL shapes (Pareti and Konstantinidis 2021) to achieve proof semantics similar to OWL without its peculiarities.

Related Work

Other authors have identified and confronted the competing challenges identified in the first section; a useful survey of algorithmic solutions is available (Yu et al. 2018). A

Rules	DL Equations	Plain-language description
M = Mission Rules		
M1	Mission ⊑ ∃startsWith.Goal ⊓ = 1.startsWith	A Mission can only start with a Goal and must start with exactly one Goal
M2	Mission ⊑ ∃includes.Goal ⊓ ≥1.includes	A Mission can only include Goals and must include one of more Goals
M3	Mission ⊑ ∃hasConstraint.Constraint ⊓ ≥0.hasConstraint	A Mission can be constrained only by Constraints and can have 0 or more
M4	startsWith ⊑ includes	A Mission must include the Goal that it starts with
M5	Mission ⊑ ∃performableBy.Vehicle ⊓ ≥0.performableBy	A Mission can only be performed by a Vehicle and can be performable by 0 or more Vehicles
M6	performableBy(M,V) ⊑ ∀(hasConstraint(M,C) ∘ canIdentify(V,C))	A Mission cannot be performable by a Vehicle unless that Vehicle has the ability to identify all Constraints associated with that mission
M7	performableBy(M,V)⊑ ∀(includes(M,G) ∘ hasCapability(V,G))	A Mission cannot be performable by a Vehicle unless that Vehicle has the capability to accomplish all Goals included in that Mission
V = Vehicle Rules		
V1	Vehicle ⊑ ∃hasFeature.Vehicle_Feature ⊓ ≥0.hasFeature	The only allowable features of a Vehicle are VehicleFeature. A Vehicle can have 0 or more VehicleFeatures
V2	canPerform ≡ performableBy⁻	performableBy and canPerform are inversely equivalent
V3	meetsRequirement ≡ hasFeature ∘ canFulfill	A Vehicle meets a GoalRequirement if and only if it has a VehicleFeature that can fullfill that GoalRequirement
V4	hasFeature ∘ canTest ⊑ canIdentify	If a Vehicle has a VehicleFeature that can test a Constraint, then that Vehicle can identify that constraint
V5	hasCapability(V,G) ⊑ ∀(requires(G,R) ⊓ meetsRequirement(V,R))	If a Vehicle meets all GoalRequirements for a specific Goal, then that vehicle has the capability for that Goal
F = Feature Rules		
F1	VehicleFeature ⊑ ∃canFulfill.GoalRequirement ⊓ ≥0.canFulfill	A VehicleFeature can only fulfill GoalRequirements and may be able to fulfill 0 or more GoalRequirements
F2	VehicleFeature ⊑ ∃can_test.Constraint ⊓ ≥0.can_test	A VehicleFeature can only test Constraints and may be able to test 0 or more Constraints
C = Constraint Rules		
C1	Constraint ⊑ ∃appliesTo.(Mission ⊔ Goal)	A Constraint can apply to a Mission or a Goal (and nothing else)
C2	Constraint ⊑ ≥1.appliesTo.Goal	A Constraint must apply to at least one Goal
C3	appliesTo ∘ includes ⊑ appliesTo	A Constraint that applies to a Mission must also apply to all of the Goals that Mission includes
EC = End Condition Rules		
EC1	EndCondition ≡ {SUCCEED, FAIL, VIOLATE}	Possible ending conditions are SUCCEED, FAIL, and VIOLATE (i.e., imminent Constraint violation)
G = Goal Rules		
G1	Goal ⊑ ∃requires.GoalRequirement ⊓ ≥0.requires	A Goal can only require a GoalRequirement and may require 0 or more Goal Requirements
G2	Goal ⊑ ∃hasEndCondition.EndCondition ⊓ ≤1.hasEndCondition.End_Condition	A Goal's ending state must be an EndCondition, and a Goal can end with at most one EndCondition
G3	Goal ⊑ ∃isNext.Goal	A Goal can only have other Goals next
G4	hasEndCondition(G,SUCCEED) ⊔ hasEndCondition (G,FAIL) ⊔ hasEndCondition (G,VIOLATE) ⊑ isNext(G,G2)	A Goal can only have an immediate successor based on the existence of an ending state for that Goal
G5	Goal(G) ⊑ ≤1.(is_next(G,G2) ⊓ end_state(G,SUCCEED)) ⊔ ≤1.(is_next(G,G2) ⊓ end_state(G,FAIL)) ⊔ ≤1.(is_next(G,G2) ⊓ end_state(G,VIOLATE))	A Goal can have no more than one immediate successor in the event of a specific ending state
G6	Goal ⊑ ∃follows.Goal	A Goal can only be followed by another Goal
G7	Goal(G) ⊑ ¬follows(G,G)	A Goal cannot follow itself (no loops)
G8	isNext ⊑ follows	A Goal follows another Goal if it is the next Goal
G9	follows ∘ follows ⊑ follows	follows is transitive (if follows(A,B) and follows(B,C), then follows(A,C))
G10	includes ≡ startsWith ∘ follows	All Goals in a Mission must *potentially* follow the starting Goal (satisfiability vice entailment)

Figure 4: Selected MEO semantic constraints expressed as OWL restrictions.

variety of approaches are represented in the literature, including Answer Set Programming (Berreby, Bourgne, and Ganascia 2015), deontic (Bringsjord, Arkoudas, and Bello 2020) and action (Wooldridge and Van Der Hoek 2005) logics, analogical reasoning (Blass and Forbus 2015), Markov Decision Processes (MDPs) (Nashed, Svegliato, and Zilberstein 2021) and Reinforcement Learning (Wu and Lin 2018), among others. Here, we comment on several promising efforts.

Deontic and action logic approaches often confront philosophical problems and are not scalable or easily deployable. In contrast, Berreby's Answer Set Programming work could be a tractable method of representing and reasoning over rich ethical constraints. Using an Event Calculus framework, this system can represent uncertain, conflicting actions and effects of multiple agents, enabling sophisticated proof reasoning about detailed consequences. This expressiveness also enables reasoning about competing ethical priorities, a focus of many authors, which the present work assumes are resolved by the mission plan. However, the representational detail in Berreby may degrade its intelligibility to human commanders.

Work on Markov Decision Processes and Reinforcement Learning can be said to employ consequentialist (as opposed to deontic) ethics because their activities are based on numerical reward functions. A primary problem for these approaches is achieving the absolute guarantees of action or restraint that are built into the present work. Decision-making based on rewards runs the risk of prioritizing the reward of some mission goal above that of ethical behavior, especially when rewards and policies are machine learned. Proving correct behavior is even more challenging. Nashed et. al's work makes headway in this area, showing that MDPs can be constrained to obey intuitive ethical precepts such as the Golden Rule.

Analogical reasoning approaches face similar criticism of potential unreliability, because it is difficult to guarantee that they will successfully retrieve any guiding ethical principle from their library via similarity search. However, these approaches at least face head on the challenging problem of a robot interpreting its environment. The present work assumes built-in capabilities in this area, e.g. that a robot can interpret its situation in enough detail to recognize when an ethical constraint is in imminent breach.

Conclusion

When compared to other leading approaches to ethical robot autonomy, the present work has several advantages when viewed in reference to the criteria introduced earlier. Our work is often comparable or superior in Expressiveness, striking a good balance between mission fidelity and human Intelligibility. Such fidelity has been achieved without compromising the Provability & Tractability required to engender human trust, even for real world missions. Because our framework is an extension of human command hierarchies, it appears more Relevant to existing military practice. Finally, the present work also enjoys Usability advantages due to its tested operational maturity and the developed infrastructure supporting its use across platforms.

References

Berreby, F.; Bourgne, G.; and Ganascia, J.-G. 2015. Modelling moral reasoning and ethical responsibility with logic programming. In *Logic for programming, artificial intelligence, and reasoning*, 532–548. Springer.

Blass, J. A.; and Forbus, K. D. 2015. Moral decision-making by analogy: Generalizations versus exemplars. In *Twenty-Ninth AAAI conference on artificial intelligence*.

Bringsjord, S.; Arkoudas, K.; and Bello, P. 2020. Toward a General Logicist Methodology for Engineering Ethically Correct Roborts. In *Machine Ethics and Robot Ethics*, 291–297. Routledge.

Brutzman, D.; Blais, C.; McGhee, R.; and Davis, D. 2017. Position Paper: Rational Behavior Model (RBM) and Human-Robot Ethical Constraints Using Mission Execution Ontology (MEO). In *2017 AAAI Fall Symposium Series*.

Brutzman, D.; Blais, C. L.; Davis, D. T.; and McGhee, R. B. 2018. Ethical mission definition and execution for maritime robots under human supervision. *IEEE Journal of Oceanic Engineering*, 43(2): 427–443.

Brutzman, D.; Davis, D.; Lucas, G. R.; and McGhee, R. 2013. Run-time ethics checking for autonomous unmanned vehicles: Developing a practical approach. In *Proceedings of the 18th International Symposium on Unmanned Untethered Submersible Technology (UUST), Portsmouth, New Hampshire*.

Brutzman, D. P. 1994. A virtual world for an autonomous underwater vehicle. Technical report, Naval Postgraduate School, Monterey, CA, U.S.

Brutzman, D. P.; Blais, C. L.; and Wu, H.-F. 2020. Ethical Control of Unmanned Systems: Lifesaving/Lethal Scenarios for Naval Operations. Technical report, Naval Postgraduate School, Monterey, CA, U.S.

Byrnes, R. B.; Healey, A. J.; McGhee, R. B.; Nelson, M. L.; Kwak, S.-H.; and Brutzman, D. P. 1996. The rational behavior software architecture for intelligent ships. *Naval Engineers Journal*, 108(2): 43–55.

Davis, D.; Blais, C.; and Brutzman, D. 2006. Autonomous vehicle command language for simulated and real robotic forces. In *Fall Simulation Interoperability Workshop*.

Forbes.com. 2019. Can We Teach Machines A Code Of Ethics? https://www.forbes.com/sites/insights-intelai/2019/03/27/can-we-teach-machines-a-code-of-ethics/.

Group, W. O. W.; et al. 2009. OWL 2 web ontology language document overview. *http://www. w3. org/TR/owl2-overview/*.

McGhee, R.; Brutzman, D.; and Davis, D. 2012. Recursive Goal Refinement and Iterative Task Abstraction for Top-Level Control of Autonomous Mobile Robots by Mission Execution Automata-A UUV Example. Technical report.

Nashed, S.; Svegliato, J.; and Zilberstein, S. 2021. Ethically compliant planning within moral communities. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 188–198.

Pareti, P.; and Konstantinidis, G. 2021. A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs. *arXiv preprint arXiv:2112.01441*.

United States Marine Corps. Post 2005. Law of War / Introduction to Rules of Engagement. https://www.trngcmd.marines.mil/Portals/207/Docs/TBS/B130936LawofWarandRulesOfEngagement.pdf.

U.S. Department of Defense. 2012. Autonomy in Weapon Systems. https://www.esd.whs.mil/portals/54/documents/dd/issuances/dodd/300009p.pdf.

Weekley, J.; Brutzman, D.; Healey, A.; Davis, D.; and Lee, D. 2004. AUV workbench: integrated 3D for interoperable mission rehearsal, reality and replay. In *Proceedings of the Mine Warfare Association Australian-American Mine Warfare Conference*.

Wikipedia. 2021a. International humanitarian law — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/International_humanitarian_law.

Wikipedia. 2021b. Rules of engagement — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Rules_of_engagement.

Wooldridge, M.; and Van Der Hoek, W. 2005. On obligations and normative ability: Towards a logical analysis of the social contract. *Journal of Applied Logic*, 3(3-4): 396–420.

Wu, Y.-H.; and Lin, S.-D. 2018. A low-cost ethics shaping approach for designing reinforcement learning agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Yu, H.; Shen, Z.; Miao, C.; Leung, C.; Lesser, V. R.; and Yang, Q. 2018. Building ethics into artificial intelligence. *arXiv preprint arXiv:1812.02953*.